# Intel® 82801EB I/O Controller Hub 5 (ICH5) / Intel® 82801ER I/O Controller Hub 5 R (ICH5R)

## Datasheet

*April 2003*

Document Number: 252516-001

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. Intel products are not intended for use in medical, life saving, or life sustaining applications.

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 Intel® 82801EB I/O Controller Hub 5 (ICH5) / Intel® 82801ER I/O Controller Hub 5 R (ICH5R) chipset component 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.

Contact your local Intel sales office or your distributor to obtain the latest specifications and before placing your product order.

$I^2C$ 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.

Alert on LAN is a result of the Intel-IBM Advanced Manageability Alliance and a trademark of IBM.

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.

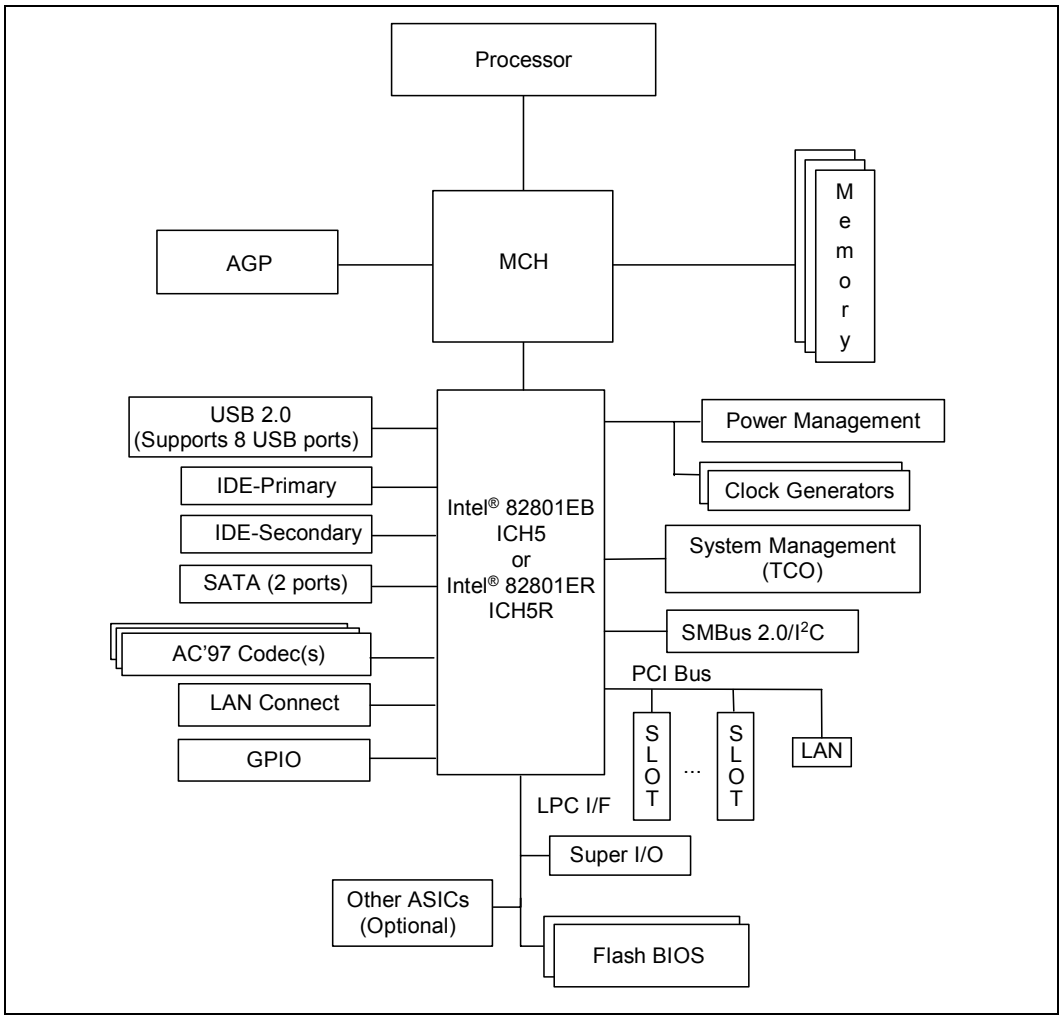
Copyright © 2002–2003, Intel Corporation

# Intel® ICH5/ICH5R Features

- PCI Bus Interface
  - — New: Supports PCI Revision 2.3 Specification at 33 MHz
  - — 6 available PCI REQ/GNT pairs
  - — One PCI REQ/GNT pair can be given higher arbitration priority (intended for external 1394 host controller)
  - — Support for 44-bit addressing on PCI using DAC protocol
- Integrated LAN Controller
  - — New: Integrated ASF Management Controller
  - — WfM 2.0 and IEEE 802.3 Compliant
  - — LAN Connect Interface (LCI)
  - — 10/100 Mbit/sec Ethernet Support
- New: Integrated Serial ATA Host Controllers
  - — Independent DMA operation on two ports.
  - — Data transfer rates up to 1.5 Gb/s (150 MB/s).
  - — RAID Level 0 Support (ICH5R Only)
- Integrated IDE Controller
  - — Supports “Native Mode” Register and Interrupts
  - — Independent timing of up to 4 drives
  - — Ultra ATA/100/66/33, BMIDE and PIO modes
  - — Tri-state modes to enable swap bay
- USB 2.0
  - — New: Includes 4 UHCI Host Controllers, increasing the number of external ports to eight
  - — Includes 1 EHCI Host Controller that supports all eight ports
  - — Includes 1 USB 2.0 high-speed debug port
  - — Supports wake-up from sleeping states S1–S5
  - — Supports legacy Keyboard/Mouse software
- AC-Link for Audio and Telephony Codecs
  - — Support for 3 AC ‘97 2.3 codecs.
  - — Independent bus master logic for 8 channels (PCM In/Out, PCM2 In, Mic 1 Input, Mic 2 Input, Modem In/Out, S/PDIF Out)
  - — Support for up to six channels of PCM audio output (full AC3 decode)
  - — Supports wake-up events
- Interrupt Controller
  - — Supports up to 8 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 Front Side Bus interrupt delivery
- High-Precision Event Timers
  - — Advanced operating system interrupt scheduling
- New: 1.5 V operation with 3.3 V I/O
  - — 5V tolerant buffers on IDE, PCI, USB Over-current and Legacy signals
- Timers Based on 82C54
  - — System timer, Refresh request, Speaker tone output
- New: Integrated 1.5 V Voltage Regulator (INTVR) for the Suspend wells
- Power Management Logic
  - — New: ACPI 2.0 compliant
  - — ACPI-defined power states (C1, S3–S5)
  - — ACPI Power Management Timer
  - — PCI 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 implementations
- External Glue Integration
  - — Integrated Pull-up, Pull-down and Series Termination resistors on IDE, processor I/F
  - — Integrated Pull-down and Series resistors on USB
- Flash BIOS I/F supports BIOS Memory size up to 8 Mbytes
- Low Pin Count (LPC) I/F
  - — Supports two Master/DMA devices.
  - — Support for Security Devices connected to LPC.
- Enhanced DMA Controller
  - — Two cascaded 8237 DMA controllers
  - — PCI DMA: Supports PC/PCI — Includes two PC/PCI REQ#/GNT# pairs
  - — Supports LPC DMA
  - — Supports DMA Collection Buffer to provide Type-F DMA performance for all DMA channels
- 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
- SMBus
  - — New: 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 2-Wire components that are also $I^2C$ compatible
- GPIO
  - — TTL, Open-Drain, Inversion
- Package 31x31 mm 460 mBGA

The Intel® ICH5 / ICH5R may contain design defects or errors known as errata which may cause the products to deviate from published specifications. Current characterized errata are available on request.

## System Block Diagram



# Contents


**1 Introduction** ..... 39

1.1 About This Manual ..... 39
1.2 Overview ..... 42

**2 Signal Description** ..... 49

2.1 Hub Interface to Host Controller ..... 51
2.2 Link to LAN Connect ..... 51
2.3 EEPROM Interface ..... 51
2.4 Flash BIOS Interface ..... 52
2.5 PCI Interface ..... 52
2.6 Serial ATA Interface ..... 54
2.7 IDE Interface ..... 55
2.8 LPC Interface ..... 56
2.9 Interrupt Interface ..... 57
2.10 USB Interface ..... 58
2.11 Power Management Interface ..... 59
2.12 Processor Interface ..... 60
2.13 SMBus Interface ..... 61
2.14 System Management Interface ..... 61
2.15 Real Time Clock Interface ..... 62
2.16 Other Clocks ..... 62
2.17 Miscellaneous Signals ..... 62
2.18 AC-Link ..... 63
2.19 General Purpose I/O ..... 63
2.20 Power and Ground ..... 65
2.21 Pin Straps ..... 66
2.21.1 Functional Straps ..... 66
2.21.2 External RTC Circuitry ..... 67
2.21.3 Power Sequencing Requirements ..... 67
2.21.3.1 V5REF / Vcc3_3 Sequencing Requirements ..... 67
2.21.3.2 3.3 V/1.5 V Standby Power Sequencing Requirements ..... 68
2.21.4 Test Signals ..... 68
2.21.4.1 Test Mode Selection ..... 68

**3 Intel® ICH5 Power Planes and Pin States** ..... 69

3.1 Power Planes ..... 69
3.2 Integrated Pull-Ups and Pull-Downs ..... 70
3.3 IDE Integrated Series Termination Resistors ..... 71
3.4 Output and I/O Signals Planes and States ..... 71
3.5 Power Planes for Input Signals ..... 75

**4 Intel® ICH5 and System Clock Domains** ..... 77

**5 Functional Description** ..... 79

5.1 Hub Interface to PCI Bridge (D30:F0) ..... 79
5.1.1 PCI Bus Interface ..... 79
5.1.2 PCI-to-PCI Bridge Model ..... 80

5.1.3 IDSEL to Device Number Mapping ..... 80
5.1.4 SERR# Functionality ..... 80
5.1.5 Parity Error Detection ..... 82
5.1.6 Standard PCI Bus Configuration Mechanism ..... 83
5.1.6.1 Type 0 to Type 0 Forwarding ..... 83
5.1.6.2 Type 1 to Type 0 Conversion ..... 83
5.1.7 PCI Dual Address Cycle (DAC) Support ..... 84
5.2 LAN Controller (B1:D8:F0) ..... 84
5.2.1 LAN Controller Architectural Overview ..... 85
5.2.1.1 Parallel Subsystem Overview ..... 86
5.2.1.2 FIFO Subsystem Overview ..... 87
5.2.1.3 Serial CSMA/CD Unit Overview ..... 87
5.2.2 LAN Controller PCI Bus Interface ..... 88
5.2.2.1 Bus Slave Operation ..... 88
5.2.2.2 Bus Master Operation ..... 90
5.2.2.3 PCI Power Management ..... 93
5.2.2.4 PCI Reset Signal ..... 94
5.2.2.5 Wake-Up Events ..... 95
5.2.2.6 Wake on LAN* (Preboot Wake-Up) ..... 96
5.2.3 Serial EEPROM Interface ..... 96
5.2.4 CSMA/CD Unit ..... 97
5.2.4.1 Full Duplex ..... 97
5.2.4.2 Flow Control ..... 97
5.2.4.3 Address Filtering Modifications ..... 97
5.2.4.4 VLAN Support ..... 98
5.2.5 Media Management Interface ..... 98
5.2.6 TCO Functionality ..... 98
5.2.6.1 Advanced TCO Mode ..... 98
5.3 Alert Standard Format (ASF) ..... 100
5.3.1 ASF Management Solution Features/Capabilities ..... 101
5.3.2 ASF Hardware Support ..... 102
5.3.2.1 82562EM/EX ..... 102
5.3.2.2 EEPROM (256x16, 1 MHz) ..... 102
5.3.2.3 Legacy Sensor SMBus Devices ..... 102
5.3.2.4 Remote Control SMBus Devices ..... 102
5.3.2.5 ASF Sensor SMBus Devices ..... 102
5.3.3 ASF Software Support ..... 102
5.4 LPC Bridge (w/ System and Management Functions) (D31:F0) ..... 103
5.4.1 LPC Interface ..... 103
5.4.1.1 LPC Cycle Types ..... 104
5.4.1.2 Start Field Definition ..... 104
5.4.1.3 Cycle Type / Direction (CYCTYPE + DIR) ..... 105
5.4.1.4 SIZE ..... 105
5.4.1.5 SYNC ..... 106
5.4.1.6 SYNC Time-Out ..... 106
5.4.1.7 SYNC Error Indication ..... 106
5.4.1.8 LFRAME# Usage ..... 107
5.4.1.9 I/O Cycles ..... 108
5.4.1.10 Bus Master Cycles ..... 108
5.4.1.11 LPC Power Management ..... 108
5.4.1.12 Configuration and Intel® ICH5 Implications ..... 108

5.5 DMA Operation (D31:F0) ..... 109
5.5.1 Channel Priority ..... 110
5.5.1.1 Fixed Priority ..... 110
5.5.1.2 Rotating Priority ..... 110
5.5.2 Address Compatibility Mode ..... 110
5.5.3 Summary of DMA Transfer Sizes ..... 111
5.5.3.1 Address Shifting When Programmed for 16-Bit I/O Count by Words ..... 111
5.5.4 Autoinitialize ..... 111
5.5.5 Software Commands ..... 112
5.5.5.1 Clear Byte Pointer Flip-Flop ..... 112
5.5.5.2 DMA Master Clear ..... 112
5.5.5.3 Clear Mask Register ..... 112
5.6 PCI DMA ..... 113
5.6.1 PCI DMA Expansion Protocol ..... 113
5.6.2 PCI DMA Expansion Cycles ..... 114
5.6.3 DMA Addresses ..... 115
5.6.4 DMA Data Generation ..... 115
5.6.5 DMA Byte Enable Generation ..... 115
5.6.6 DMA Cycle Termination ..... 116
5.6.7 LPC DMA ..... 116
5.6.8 Asserting DMA Requests ..... 116
5.6.9 Abandoning DMA Requests ..... 117
5.6.10 General Flow of DMA Transfers ..... 117
5.6.11 Terminal Count ..... 118
5.6.12 Verify Mode ..... 118
5.6.13 DMA Request Deassertion ..... 118
5.6.14 SYNC Field / LDRQ# Rules ..... 119
5.7 8254 Timers (D31:F0) ..... 120
5.7.1 Timer Programming ..... 120
5.7.2 Reading from the Interval Timer ..... 121
5.7.2.1 Simple Read ..... 122
5.7.2.2 Counter Latch Command ..... 122
5.7.2.3 Read Back Command ..... 122
5.8 8259 Interrupt Controllers (PIC) (D31:F0) ..... 123
5.8.1 Interrupt Handling ..... 124
5.8.1.1 Generating Interrupts ..... 124
5.8.1.2 Acknowledging Interrupts ..... 124
5.8.1.3 Hardware/Software Interrupt Sequence ..... 125
5.8.2 Initialization Command Words (ICWx) ..... 125
5.8.2.1 ICW1 ..... 125
5.8.2.2 ICW2 ..... 126
5.8.2.3 ICW3 ..... 126
5.8.2.4 ICW4 ..... 126
5.8.3 Operation Command Words (OCW) ..... 126
5.8.4 Modes of Operation ..... 126
5.8.4.1 Fully Nested Mode ..... 126
5.8.4.2 Special Fully-Nested Mode ..... 127
5.8.4.3 Automatic Rotation Mode (Equal Priority Devices) ..... 127
5.8.4.4 Specific Rotation Mode (Specific Priority) ..... 127
5.8.4.5 Poll Mode ..... 127

5.8.4.6 Cascade Mode ..... 128
5.8.4.7 Edge and Level Triggered Mode ..... 128
5.8.4.8 End of Interrupt Operations ..... 128
5.8.4.9 Normal End of Interrupt ..... 128
5.8.4.10 Automatic End of Interrupt Mode ..... 128
5.8.5 Masking Interrupts ..... 129
5.8.5.1 Masking on an Individual Interrupt Request ..... 129
5.8.5.2 Special Mask Mode ..... 129
5.8.6 Steering PCI Interrupts ..... 129
5.9 Advanced Interrupt Controller (APIC) (D31:F0) ..... 130
5.9.1 Interrupt Handling ..... 130
5.9.2 Interrupt Mapping ..... 130
5.9.3 PCI Message-Based Interrupts ..... 131
5.9.3.1 Registers and Bits Associated with PCI Interrupt Delivery ..... 132
5.9.4 Front Side Bus Interrupt Delivery ..... 133
5.9.4.1 Edge-Triggered Operation ..... 133
5.9.4.2 Level-Triggered Operation ..... 133
5.9.4.3 Registers Associated with Front Side Bus Interrupt Delivery ..... 133
5.9.4.4 Interrupt Message Format ..... 133
5.10 Serial Interrupt (D31:F0) ..... 135
5.10.1 Start Frame ..... 135
5.10.2 Data Frames ..... 136
5.10.3 Stop Frame ..... 136
5.10.4 Specific Interrupts Not Supported via SERIRQ ..... 136
5.10.5 Data Frame Format ..... 137
5.11 Real Time Clock (D31:F0) ..... 137
5.11.1 Update Cycles ..... 138
5.11.2 Interrupts ..... 138
5.11.3 Lockable RAM Ranges ..... 139
5.11.4 Century Rollover ..... 139
5.11.5 Clearing Battery-Backed RTC RAM ..... 139
5.12 Processor Interface (D31:F0) ..... 141
5.12.1 Processor Interface Signals ..... 141
5.12.1.1 A20M# (Mask A20) ..... 141
5.12.1.2 INIT# (Initialization) ..... 141
5.12.1.3 FERR#/IGNNE# (Numeric Coprocessor Error / Ignore Numeric Error) ..... 142
5.12.1.4 NMI (Non-Maskable Interrupt) ..... 143
5.12.1.5 Stop Clock Request and CPU Sleep (STPCLK# and CPUSLP#) ..... 143
5.12.1.6 CPU Power Good (CPUPWRGOOD) ..... 143
5.12.2 Dual-Processor Issues ..... 143
5.12.2.1 Signal Differences ..... 143
5.12.2.2 Power Management ..... 144
5.12.3 Speed Strapping for Processor ..... 144
5.13 Power Management (D31:F0) ..... 146
5.13.1 Features ..... 146
5.13.2 Intel® ICH5 and System Power States ..... 146
5.13.3 System Power Planes ..... 148
5.13.4 Intel® ICH5 Power Planes ..... 148
5.13.5 SMI#/SCI Generation ..... 148

5.13.6 Dynamic Processor Clock Control ..... 150
5.13.6.1 Throttling Using STPCLK# ..... 151
5.13.6.2 Transition Rules among S0/Cx and Throttling States ..... 151
5.13.7 Sleep States ..... 151
5.13.7.1 Sleep State Overview ..... 151
5.13.7.2 Initiating Sleep State ..... 152
5.13.7.3 Exiting Sleep States ..... 152
5.13.7.4 Sx-G3-Sx, Handling Power Failures ..... 154
5.13.8 Thermal Management ..... 155
5.13.8.1 THRM# Signal ..... 155
5.13.8.2 THRM# Initiated Passive Cooling ..... 155
5.13.8.3 THRM# Override Software Bit ..... 155
5.13.8.4 Processor Initiated Passive Cooling (Via Programmed Duty Cycle on STPCLK#) ..... 156
5.13.8.5 Active Cooling ..... 156
5.13.9 Event Input Signals and Their Usage ..... 156
5.13.9.1 PWRBTN# (Power Button) ..... 156
5.13.9.2 RI# (Ring Indicator) ..... 157
5.13.9.3 PME# (PCI Power Management Event) ..... 158
5.13.9.4 SYS_RESET# Signal ..... 158
5.13.9.5 THRMTRIP# Signal ..... 158
5.13.10 ALT Access Mode ..... 159
5.13.10.1 Write Only Registers with Read Paths in ALT Access Mode ..... 159
5.13.10.2 PIC Reserved Bits ..... 161
5.13.10.3 Read Only Registers with Write Paths in ALT Access Mode ..... 161
5.13.11 System Power Supplies, Planes, and Signals ..... 161
5.13.11.1 Power Plane Control with SLP_S3#, SLP_S4# and SLP_S5# ..... 161
5.13.11.2 SLP_S4# and Suspend-To-RAM Sequencing ..... 162
5.13.11.3 PWROK Signal ..... 162
5.13.11.4 VRMPWRGD Signal ..... 162
5.13.11.5 Controlling Leakage and Power Consumption during Low-Power States ..... 163
5.13.12 Clock Generators ..... 163
5.13.13 Legacy Power Management Theory of Operation ..... 164
5.13.13.1 APM Power Management ..... 164
5.14 System Management (D31:F0) ..... 164
5.14.1 Theory of Operation ..... 165
5.14.1.1 Detecting a System Lockup ..... 165
5.14.1.2 Handling an Intruder ..... 165
5.14.1.3 Detecting Improper Flash BIOS Programming ..... 165
5.14.1.4 Handling an ECC Error or Other Memory Error ..... 166
5.14.2 Heartbeat and Event Reporting via SMBUS ..... 166
5.15 General Purpose I/O ..... 170
5.15.1 GPIO Mapping ..... 170
5.15.2 Power Wells ..... 173
5.15.3 SMI# and SCI Routing ..... 173
5.16 IDE Controller (D31:F1) ..... 174
5.16.1 PIO Transfers ..... 174
5.16.1.1 IDE Port Decode ..... 174

5.16.1.2 IDE Legacy Mode and Native Mode ..... 175
5.16.1.3 PIO IDE Timing Modes ..... 176
5.16.1.4 IORDY Masking ..... 176
5.16.1.5 PIO 32-Bit IDE Data Port Accesses ..... 176
5.16.1.6 PIO IDE Data Port Prefetching and Posting ..... 177
5.16.2 Bus Master Function ..... 177
5.16.2.1 Physical Region Descriptor Format ..... 177
5.16.2.2 Line Buffer ..... 178
5.16.2.3 Bus Master IDE Timings ..... 178
5.16.2.4 Interrupts ..... 178
5.16.2.5 Bus Master IDE Operation ..... 179
5.16.2.6 Error Conditions ..... 180
5.16.2.7 8237-Like Protocol ..... 180
5.16.3 Ultra ATA/33 Protocol ..... 181
5.16.3.1 Signal Descriptions ..... 181
5.16.3.2 Operation ..... 182
5.16.3.3 CRC Calculation ..... 182
5.16.4 Ultra ATA/66 Protocol ..... 183
5.16.5 Ultra ATA/100 Protocol ..... 183
5.16.6 Ultra ATA/33/66/100 Timing ..... 183
5.16.7 IDE Swap Bay ..... 184
5.16.8 SMI Trapping (APM) ..... 184
5.17 SATA Host Controller (D31:F2) ..... 185
5.17.1 Theory of Operation ..... 185
5.17.1.1 Standard ATA Emulation ..... 185
5.17.1.2 48-Bit LBA Operation ..... 185
5.17.2 Hot Swap Operation ..... 186
5.17.3 Intel® RAID Technology Configuration (Intel® 82801ER ICH5R Only) ..... 186
5.17.3.1 Intel® RAID Technology Option ROM ..... 186
5.17.4 Power Management Operation ..... 186
5.17.4.1 Power State Mappings ..... 186
5.17.4.2 Power State Transitions ..... 187
5.17.4.3 SMI Trapping (APM) ..... 188
5.17.5 SATA Interrupts ..... 188
5.18 High-Precision Event Timers ..... 188
5.18.1 Timer Accuracy ..... 189
5.18.2 Interrupt Mapping ..... 189
5.18.3 Periodic vs. Non-Periodic Modes ..... 189
5.18.4 Enabling the Timers ..... 191
5.18.5 Interrupt Levels ..... 191
5.18.6 Handling Interrupts ..... 191
5.18.7 Issues Related to 64-Bit Timers with 32-Bit Processors ..... 191
5.19 USB UHCI Host Controllers (D29:F0, F1, F2, and F3) ..... 192
5.19.1 Data Structures in Main Memory ..... 192
5.19.1.1 Frame List Pointer ..... 192
5.19.1.2 Transfer Descriptor (TD) ..... 193
5.19.1.3 Queue Head (QH) ..... 197
5.19.2 Data Transfers to/from Main Memory ..... 198
5.19.2.1 Executing the Schedule ..... 198
5.19.2.2 Processing Transfer Descriptors ..... 199
5.19.2.3 Command Register, Status Register, and TD Status Bit Interaction ..... 200

5.19.2.4 Transfer Queuing ..... 200
5.19.3 Data Encoding and Bit Stuffing ..... 204
5.19.4 Bus Protocol ..... 204
5.19.4.1 Bit Ordering ..... 204
5.19.4.2 SYNC Field ..... 204
5.19.4.3 Packet Field Formats ..... 205
5.19.4.4 Address Fields ..... 206
5.19.4.5 Frame Number Field ..... 206
5.19.4.6 Data Field ..... 206
5.19.4.7 Cyclic Redundancy Check (CRC) ..... 206
5.19.5 Packet Formats ..... 207
5.19.5.1 Token Packets ..... 207
5.19.5.2 Start of Frame Packets ..... 207
5.19.5.3 Data Packets ..... 208
5.19.5.4 Handshake Packets ..... 208
5.19.5.5 Handshake Responses ..... 209
5.19.6 USB Interrupts ..... 209
5.19.6.1 Transaction Based Interrupts ..... 209
5.19.6.2 Non-Transaction Based Interrupts ..... 211
5.19.7 USB Power Management ..... 212
5.19.8 USB Legacy Keyboard Operation ..... 212
5.20 USB EHCI Host Controller (D29:F7) ..... 215
5.20.1 EHC Initialization ..... 215
5.20.1.1 Power On ..... 215
5.20.1.2 BIOS Initialization ..... 215
5.20.1.3 Driver Initialization ..... 216
5.20.1.4 EHC Resets ..... 216
5.20.2 Data Structures in Main Memory ..... 216
5.20.3 USB 2.0 Enhanced Host Controller DMA ..... 216
5.20.3.1 Periodic List Execution ..... 216
5.20.3.2 Asynchronous List Execution ..... 218
5.20.4 Data Encoding and Bit Stuffing ..... 219
5.20.5 Packet Formats ..... 219
5.20.6 USB 2.0 Interrupts and Error Conditions ..... 220
5.20.6.1 Aborts on USB 2.0-Initiated Memory Reads ..... 220
5.20.7 USB 2.0 Power Management ..... 221
5.20.7.1 Pause Feature ..... 221
5.20.7.2 Suspend Feature ..... 221
5.20.7.3 ACPI Device States ..... 221
5.20.7.4 ACPI System States ..... 222
5.20.8 Interaction with UHCI Host Controllers ..... 222
5.20.8.1 Port-Routing Logic ..... 223
5.20.8.2 Device Connects ..... 224
5.20.8.3 Device Disconnects ..... 225
5.20.8.4 Effect of Resets on Port-Routing Logic ..... 225
5.20.9 USB 2.0 Legacy Keyboard Operation ..... 225
5.20.10 USB 2.0 Based Debug Port ..... 226
5.20.10.1 Theory of Operation ..... 226
5.21 SMBus Controller (D31:F3) ..... 231
5.21.1 Host Controller ..... 231
5.21.1.1 Command Protocols ..... 232
5.21.1.2 $I^2C$ Behavior ..... 242

5.21.2 Bus Arbitration ..... 243
5.21.3 Bus Timing ..... 243
5.21.3.1 Clock Stretching ..... 243
5.21.3.2 Bus Time Out (Intel® ICH5 as SMBus Master) ..... 243
5.21.4 Interrupts / SMI# ..... 244
5.21.5 SMBALERT# ..... 245
5.21.6 SMBus CRC Generation and Checking ..... 245
5.21.7 SMBus Slave Interface ..... 245
5.21.7.1 Format of Slave Write Cycle ..... 246
5.21.7.2 Format of Read Command ..... 248
5.21.7.3 Format of Host Notify Command ..... 250
5.22 AC '97 Controller (Audio D31:F5, Modem D31:F6) ..... 251
5.22.1 PCI Power Management ..... 253
5.22.2 AC-Link Overview ..... 253
5.22.2.1 AC-Link Output Frame (SDOUT) ..... 256
5.22.2.2 Output Slot 0: Tag Phase ..... 256
5.22.2.3 Output Slot 1: Command Address Port ..... 256
5.22.2.4 Output Slot 2: Command Data Port ..... 257
5.22.2.5 Output Slot 3: PCM Playback Left Channel ..... 257
5.22.2.6 Output Slot 4: PCM Playback Right Channel ..... 257
5.22.2.7 Output Slot 5: Modem Codec ..... 257
5.22.2.8 Output Slot 6: PCM Playback Center Front Channel ..... 257
5.22.2.9 Output Slots 7–8: PCM Playback Left and Right Rear Channels ..... 257
5.22.2.10 Output Slot 9: Playback Sub Woofer Channel ..... 258
5.22.2.11 Output Slots 10–11: Reserved ..... 258
5.22.2.12 Output Slot 12: I/O Control ..... 258
5.22.2.13 AC-Link Input Frame (SDIN) ..... 258
5.22.2.14 Input Slot 0: Tag Phase ..... 259
5.22.2.15 Input Slot 1: Status Address Port / Slot Request Bits ..... 259
5.22.2.16 Input Slot 2: Status Data Port ..... 260
5.22.2.17 Input Slot 3: PCM Record Left Channel ..... 260
5.22.2.18 Input Slot 4: PCM Record Right Channel ..... 260
5.22.2.19 Input Slot 5: Modem Line ..... 260
5.22.2.20 Input Slot 6: Optional Dedicated Microphone Record Data ..... 261
5.22.2.21 Input Slots 7–11: Reserved ..... 261
5.22.2.22 Input Slot 12: I/O Status ..... 261
5.22.2.23 Register Access ..... 261
5.22.3 AC-Link Low Power Mode ..... 262
5.22.3.1 External Wake Event ..... 263
5.22.4 AC '97 Cold Reset ..... 264
5.22.5 AC '97 Warm Reset ..... 264
5.22.6 System Reset ..... 264
5.22.7 Hardware Assist to Determine AC_SDIN Used Per Codec ..... 265
5.22.8 Software Mapping of AC_SDIN to DMA Engine ..... 265

**6 Register and Memory Mapping** ..... 267

6.1 PCI Devices and Functions ..... 268
6.2 PCI Configuration Map ..... 269
6.3 I/O Map ..... 269
6.3.1 Fixed I/O Address Ranges ..... 269

6.3.2 Variable I/O Decode Ranges ..... 272
6.4 Memory Map ..... 273
6.4.1 Boot-Block Update Scheme ..... 274

**7 LAN Controller Registers (B1:D8:F0)** ..... 275

7.1 PCI Configuration Registers (LAN Controller—B1:D8:F0) ..... 275
7.1.1 VID—Vendor Identification Register (LAN Controller—B1:D8:F0) ..... 276
7.1.2 DID—Device Identification Register (LAN Controller—B1:D8:F0) ..... 276
7.1.3 PCICMD—PCI Command Register (LAN Controller—B1:D8:F0) ..... 277
7.1.4 PCISTS—PCI Status Register (LAN Controller—B1:D8:F0) ..... 278
7.1.5 RID—Revision Identification Register (LAN Controller—B1:D8:F0) ..... 279
7.1.6 SCC—Sub-Class Code Register (LAN Controller—B1:D8:F0) ..... 279
7.1.7 BCC—Base-Class Code Register (LAN Controller—B1:D8:F0) ..... 279
7.1.8 CLS—Cache Line Size Register (LAN Controller—B1:D8:F0) ..... 280
7.1.9 PMLT—Primary Master Latency Timer Register (LAN Controller—B1:D8:F0) ..... 280
7.1.10 HEADTYP—Header Type Register (LAN Controller—B1:D8:F0) ..... 280
7.1.11 CSR_MEM_BASE — CSR Memory-Mapped Base Address Register (LAN Controller—B1:D8:F0) ..... 281
7.1.12 CSR_IO_BASE — CSR I/O-Mapped Base Address Register (LAN Controller—B1:D8:F0) ..... 281
7.1.13 SVID — Subsystem Vendor Identification Register (LAN Controller—B1:D8:F0) ..... 281
7.1.14 SID — Subsystem Identification Register (LAN Controller—B1:D8:F0) ..... 282
7.1.15 CAP_PTR — Capabilities Pointer Register (LAN Controller—B1:D8:F0) ..... 282
7.1.16 INT_LN — Interrupt Line Register (LAN Controller—B1:D8:F0) ..... 282
7.1.17 INT_PN — Interrupt Pin Register (LAN Controller—B1:D8:F0) ..... 283
7.1.18 MIN_GNT — Minimum Grant Register (LAN Controller—B1:D8:F0) ..... 283
7.1.19 MAX_LAT — Maximum Latency Register (LAN Controller—B1:D8:F0) ..... 283
7.1.20 CAP_ID — Capability Identification Register (LAN Controller—B1:D8:F0) ..... 283
7.1.21 NXT_PTR — Next Item Pointer Register (LAN Controller—B1:D8:F0) ..... 284
7.1.22 PM_CAP — Power Management Capabilities Register (LAN Controller—B1:D8:F0) ..... 284

7.1.23 PMCSR — Power Management Control/Status Register (LAN Controller—B1:D8:F0) ..... 285
7.1.24 PCIDATA — PCI Power Management Data Register (LAN Controller—B1:D8:F0) ..... 286
7.2 LAN Control / Status Registers (CSR) (LAN Controller—B1:D8:F0) ..... 287
7.2.1 SCB_STA—System Control Block Status Word Register (LAN Controller—B1:D8:F0) ..... 288
7.2.2 SCB_CMD—System Control Block Command Word Register (LAN Controller—B1:D8:F0) ..... 289
7.2.3 SCB_GENPNT—System Control Block General Pointer Register (LAN Controller—B1:D8:F0) ..... 291
7.2.4 PORT—PORT Interface Register (LAN Controller—B1:D8:F0) ..... 291
7.2.5 EEPROM_CNTL—EEPROM Control Register (LAN Controller—B1:D8:F0) ..... 292
7.2.6 MDI_CNTL—Management Data Interface (MDI) Control Register (LAN Controller—B1:D8:F0) ..... 293
7.2.7 REC_DMA_BC—Receive DMA Byte Count Register (LAN Controller—B1:D8:F0) ..... 293
7.2.8 EREC_INTR—Early Receive Interrupt Register (LAN Controller—B1:D8:F0) ..... 294
7.2.9 FLOW_CNTL—Flow Control Register (LAN Controller—B1:D8:F0) ..... 295
7.2.10 PMDR—Power Management Driver Register (LAN Controller—B1:D8:F0) ..... 296
7.2.11 GENCNTL—General Control Register (LAN Controller—B1:D8:F0) ..... 297
7.2.12 GENSTA—General Status Register (LAN Controller—B1:D8:F0) ..... 297
7.2.13 SMB_PCI—SMB via PCI Register (LAN Controller—B1:D8:F0) ..... 298
7.2.14 Statistical Counters (LAN Controller—B1:D8:F0) ..... 299

**8 Hub Interface to PCI Bridge Registers (D30:F0)** ..... 301

8.1 PCI Configuration Registers (D30:F0) ..... 301
8.1.1 VID—Vendor Identification Register (HUB-PCI—D30:F0) ..... 302
8.1.2 DID—Device Identification Register (HUB-PCI—D30:F0) ..... 302
8.1.3 PCICMD—PCI Command Register (HUB-PCI—D30:F0) ..... 303
8.1.4 PCISTS—PCI Status Register (HUB-PCI—D30:F0) ..... 304
8.1.5 RID—Revision Identification Register (HUB-PCI—D30:F0) ..... 305
8.1.6 SCC—Sub-Class Code Register (HUB-PCI—D30:F0) ..... 305
8.1.7 BCC—Base-Class Code Register (HUB-PCI—D30:F0) ..... 305

8.1.8 PMLT—Primary Master Latency Timer Register (HUB-PCI—D30:F0) ..... 305
8.1.9 HEADTYP—Header Type Register (HUB-PCI—D30:F0) ..... 306
8.1.10 PBUS_NUM—Primary Bus Number Register (HUB-PCI—D30:F0) ..... 306
8.1.11 SBUS_NUM—Secondary Bus Number Register (HUB-PCI—D30:F0) ..... 306
8.1.12 SUB_BUS_NUM—Subordinate Bus Number Register (HUB-PCI—D30:F0) ..... 306
8.1.13 SMLT—Secondary Master Latency Timer Register (HUB-PCI—D30:F0) ..... 307
8.1.14 IOBASE—I/O Base Register (HUB-PCI—D30:F0) ..... 307
8.1.15 IOLIM—I/O Limit Register (HUB-PCI—D30:F0) ..... 307
8.1.16 SECSTS—Secondary Status Register (HUB-PCI—D30:F0) ..... 308
8.1.17 MEMBASE—Memory Base Register (HUB-PCI—D30:F0) ..... 309
8.1.18 MEMLIM—Memory Limit Register (HUB-PCI—D30:F0) ..... 309
8.1.19 PREF_MEM_BASE—Prefetchable Memory Base Register (HUB-PCI—D30:F0) ..... 309
8.1.20 PREF_MEM_MLT—Prefetchable Memory Limit Register (HUB-PCI—D30:F0) ..... 310
8.1.21 IOBASE_HI—I/O Base Upper 16 Bits Register (HUB-PCI—D30:F0) ..... 310
8.1.22 IOLIM_HI—I/O Limit Upper 16 Bits Register (HUB-PCI—D30:F0) ..... 310
8.1.23 INT_LN—Interrupt Line Register (HUB-PCI—D30:F0) ..... 310
8.1.24 BRIDGE_CNT—Bridge Control Register (HUB-PCI—D30:F0) ..... 311
8.1.25 HI1_CMD—Hub Interface 1 Command Control Register (HUB-PCI—D30:F0) ..... 312
8.1.26 DEVICE_HIDE—Secondary PCI Device Hiding Register (HUB-PCI—D30:F0) ..... 313
8.1.27 CNF—Policy Configuration Register (HUB-PCI—D30:F0) ..... 314
8.1.28 MTT—Multi-Transaction Timer Register (HUB-PCI—D30:F0) ..... 315
8.1.29 PCI_MAST_STS—PCI Master Status Register (HUB-PCI—D30:F0) ..... 315
8.1.30 ERR_CMD—Error Command Register (HUB-PCI—D30:F0) ..... 316
8.1.31 ERR_STS—Error Status Register (HUB-PCI—D30:F0) ..... 316

**9 LPC Interface Bridge Registers (D31:F0)** ..... 317

9.1 PCI Configuration Registers (LPC I/F—D31:F0) ..... 317

9.1.1 VID—Vendor Identification Register (LPC I/F—D31:F0) ..... 318

9.1.2 DID—Device Identification Register (LPC I/F—D31:F0) ..... 318

9.1.3 PCICMD—PCI COMMAND Register (LPC I/F—D31:F0) ..... 319

9.1.4 PCISTS—PCI Status Register (LPC I/F—D31:F0) ..... 320

9.1.5 RID—Revision Identification Register (LPC I/F—D31:F0) ..... 321

9.1.6 PI—Programming Interface Register (LPC I/F—D31:F0) ..... 321

9.1.7 SCC—Sub Class Code Register (LPC I/F—D31:F0) ..... 321

9.1.8 BCC—Base Class Code Register (LPC I/F—D31:F0) ..... 321

9.1.9 HEADTYP—Header Type Register (LPC I/F—D31:F0) ..... 322

9.1.10 PMBASE—ACPI Base Address Register (LPC I/F—D31:F0) ..... 322

9.1.11 ACPI_CNTL—ACPI Control Register (LPC I/F — D31:F0) ..... 323

9.1.12 BIOS_CNTL—BIOS Control Register (LPC I/F—D31:F0) ..... 324

9.1.13 TCO_CNTL — TCO Control Register (LPC I/F — D31:F0) ..... 324

9.1.14 GPIO_BASE—GPIO Base Address Register (LPC I/F—D31:F0) ..... 325

9.1.15 GPIO_CNTL—GPIO Control Register (LPC I/F—D31:F0) ..... 325

9.1.16 PIRQ[n]_ROUT—PIRQ[A,B,C,D] Routing Control Register (LPC I/F—D31:F0) ..... 326

9.1.17 SIRQ_CNTL—Serial IRQ Control Register (LPC I/F—D31:F0) ..... 327

9.1.18 PIRQ[n]_ROUT—PIRQ[E,F,G,H] Routing Control Register (LPC I/F—D31:F0) ..... 328

9.1.19 D31_ERR_CFG—Device 31 Error Configuration Register (LPC I/F—D31:F0) ..... 328

9.1.20 D31_ERR_STS—Device 31 Error Status Register (LPC I/F—D31:F0) ..... 329

9.1.21 PCI_DMA_CFG—PCI DMA Configuration Register (LPC I/F—D31:F0) ..... 329

9.1.22 GEN_CNTL — General Control Register (LPC I/F — D31:F0) ..... 330

9.1.23 GEN_STA—General Status Register (LPC I/F—D31:F0) ..... 332

9.1.24 BACK_CNTL—Backed Up Control Register (LPC I/F—D31:F0) ..... 333

9.1.25 RTC_CONF—Real Time Clock Configuration Register (LPC I/F—D31:F0) ..... 334

9.1.26 COM_DEC—LPC I/F Communication Port Decode Ranges Register (LPC I/F—D31:F0)....335
9.1.27 LPCFDD_DEC—LPC I/F FDD and LPT Decode Ranges Register (LPC I/F—D31:F0)....335
9.1.28 FB_DEC_EN1—Flash BIOS Decode Enable 1 Register (LPC I/F—D31:F0)....336
9.1.29 GEN1_DEC—LPC I/F Generic Decode Range 1 Register (LPC I/F—D31:F0)....337
9.1.30 LPC_EN—LPC I/F Enables Register (LPC I/F—D31:F0)....337
9.1.31 FB_SEL1—Flash BIOS Select 1 Register (LPC I/F—D31:F0)....339
9.1.32 GEN2_DEC—LPC I/F Generic Decode Range 2 Register (LPC I/F—D31:F0)....340
9.1.33 FB_SEL2—Flash BIOS Select 2 Register (LPC I/F—D31:F0)....340
9.1.34 FB_DEC_EN2—Flash BIOS Decode Enable 2 Register (LPC I/F—D31:F0)....341
9.1.35 FUNC_DIS—Function Disable Register (LPC I/F—D31:F0)....342
9.2 DMA I/O Registers (LPC I/F—D31:F0)....344
9.2.1 DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0)....345
9.2.2 DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)....346
9.2.3 DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0)....346
9.2.4 DMACMD—DMA Command Register (LPC I/F—D31:F0)....347
9.2.5 DMASTA—DMA Status Register (LPC I/F—D31:F0)....347
9.2.6 DMA_WRSMSK—DMA Write Single Mask Register (LPC I/F—D31:F0)....348
9.2.7 DMACH_MODE—DMA Channel Mode Register (LPC I/F—D31:F0)....349
9.2.8 DMA Clear Byte Pointer Register (LPC I/F—D31:F0)....349
9.2.9 DMA Master Clear Register (LPC I/F—D31:F0)....350
9.2.10 DMA_CLMSK—DMA Clear Mask Register (LPC I/F—D31:F0)....350
9.2.11 DMA_WRMSK—DMA Write All Mask Register (LPC I/F—D31:F0)....350
9.3 Timer I/O Registers (LPC I/F—D31:F0)....351
9.3.1 TCW—Timer Control Word Register (LPC I/F—D31:F0)....352
9.3.1.1 RDBK_CMD—Read Back Command (LPC I/F—D31:F0)....353
9.3.1.2 LTCH_CMD—Counter Latch Command (LPC I/F—D31:F0)....353
9.3.2 SBYTE_FMT—Interval Timer Status Byte Format Register (LPC I/F—D31:F0)....354

9.3.3 Counter Access Ports Register (LPC I/F—D31:F0) ..... 355
9.4 8259 Interrupt Controller (PIC) Registers (LPC I/F—D31:F0) ..... 355
9.4.1 Interrupt Controller I/O MAP (LPC I/F—D31:F0) ..... 355
9.4.2 ICW1—Initialization Command Word 1 Register (LPC I/F—D31:F0) ..... 356
9.4.3 ICW2—Initialization Command Word 2 Register (LPC I/F—D31:F0) ..... 357
9.4.4 ICW3—Master Controller Initialization Command Word 3 Register (LPC I/F—D31:F0) ..... 357
9.4.5 ICW3—Slave Controller Initialization Command Word 3 Register (LPC I/F—D31:F0) ..... 358
9.4.6 ICW4—Initialization Command Word 4 Register (LPC I/F—D31:F0) ..... 358
9.4.7 OCW1—Operational Control Word 1 (Interrupt Mask) Register (LPC I/F—D31:F0) ..... 359
9.4.8 OCW2—Operational Control Word 2 Register (LPC I/F—D31:F0) ..... 359
9.4.9 OCW3—Operational Control Word 3 Register (LPC I/F—D31:F0) ..... 360
9.4.10 ELCR1—Master Controller Edge/Level Triggered Register (LPC I/F—D31:F0) ..... 361
9.4.11 ELCR2—Slave Controller Edge/Level Triggered Register (LPC I/F—D31:F0) ..... 362
9.5 Advanced Interrupt Controller (APIC)(D31:F0) ..... 363
9.5.1 APIC Register Map (LPC I/F—D31:F0) ..... 363
9.5.2 IND—Index Register (LPC I/F—D31:F0) ..... 363
9.5.3 DAT—Data Register (LPC I/F—D31:F0) ..... 364
9.5.4 IRQPA—IRQ Pin Assertion Register (LPC I/F—D31:F0) ..... 364
9.5.5 EOIR—EOI Register (LPC I/F—D31:F0) ..... 365
9.5.6 ID—Identification Register (LPC I/F—D31:F0) ..... 365
9.5.7 VER—Version Register (LPC I/F—D31:F0) ..... 366
9.5.8 REDIR_TBL—Redirection Table (LPC I/F—D31:F0) ..... 367
9.6 Real Time Clock Registers (LPC I/F—D31:F0) ..... 369
9.6.1 I/O Register Address Map (LPC I/F—D31:F0) ..... 369
9.6.2 RTC_REGA—Register A (LPC I/F—D31:F0) ..... 370
9.6.3 RTC_REGB—Register B (General Configuration) (LPC I/F—D31:F0) ..... 371
9.6.4 RTC_REGC—Register C (Flag Register) (LPC I/F—D31:F0) ..... 372
9.6.5 RTC_REGD—Register D (Flag Register) (LPC I/F—D31:F0) ..... 372

9.7 Processor Interface Registers (LPC I/F—D31:F0) ..... 373
9.7.1 NMI_SC—NMI Status and Control Register (LPC I/F—D31:F0) ..... 373
9.7.2 NMI_EN—NMI Enable (and Real Time Clock Index) Register (LPC I/F—D31:F0) ..... 374
9.7.3 PORT92—Fast A20 and Init Register (LPC I/F—D31:F0) ..... 374
9.7.4 COPROC_ERR—Coprocessor Error Register (LPC I/F—D31:F0) ..... 375
9.7.5 RST_CNT—Reset Control Register (LPC I/F—D31:F0) ..... 375
9.8 Power Management PCI Configuration Registers (PM—D31:F0) ..... 376
9.8.1 GEN_PMCON_1—General PM Configuration 1 Register (PM—D31:F0) ..... 377
9.8.2 GEN_PMCON_2—General PM Configuration 2 Register (PM—D31:F0) ..... 378
9.8.3 GEN_PMCON_3—General PM Configuration 3 Register (PM—D31:F0) ..... 379
9.8.4 STPCLK_DEL—Stop Clock Delay Register (PM—D31:F0) ..... 380
9.8.5 USB_TDD—USB Transient Disconnect Detect (PM—D31:F0) ..... 380
9.8.6 SATA_RD_CFG—SATA RAID Configuration (PM—D31:F0) - (Intel® 82801ER ICH5R Only) ..... 380
9.8.7 GPI_ROUT—GPI Routing Control Register (PM—D31:F0) ..... 381
9.8.8 TRP_FWD_EN—IO Monitor Trap Forwarding Enable Register (PM—D31:F0) ..... 382
9.8.9 MON[n]_TRP_RNG—I/O Monitor [4:7] Trap Range Register for Devices 4–7 (PM—D31:F0) ..... 383
9.8.10 MON_TRP_MSK—I/O Monitor Trap Range Mask Register for Devices 4–7 (PM—D31:F0) ..... 383
9.9 APM I/O Decode ..... 384
9.9.1 APM_CNT—Advanced Power Management Control Port Register ..... 384
9.9.2 APM_STS—Advanced Power Management Status Port Register ..... 384
9.10 Power Management I/O Registers ..... 385
9.10.1 PM1_STS—Power Management 1 Status Register ..... 386
9.10.2 PM1_EN—Power Management 1 Enable Register ..... 388
9.10.3 PM1_CNT—Power Management 1 Control ..... 389
9.10.4 PM1_TMR—Power Management 1 Timer Register ..... 390
9.10.5 PROC_CNT—Processor Control Register ..... 390
9.10.6 GPE0_STS—General Purpose Event 0 Status Register ..... 392
9.10.7 GPE0_EN—General Purpose Event 0 Enables Register ..... 394
9.10.8 SMI_EN—SMI Control and Enable Register ..... 396
9.10.9 SMI_STS—SMI Status Register ..... 398
9.10.10 ALT_GP_SMI_EN—Alternate GPI SMI Enable Register ..... 400
9.10.11 ALT_GP_SMI_STS—Alternate GPI SMI Status Register ..... 400
9.10.12 MON_SMI—Device Monitor SMI Status and Enable Register ..... 400
9.10.13 DEVACT_STS — Device Activity Status Register ..... 401

9.10.14 DEVTRAP_EN—Device Trap Enable Register ..... 402
9.11 System Management TCO Registers (D31:F0) ..... 403
9.11.1 TCO_RLD—TCO Timer Reload and Current Value Register ..... 403
9.11.2 TCO_TMR—TCO Timer Initial Value Register ..... 404
9.11.3 TCO_DAT_IN—TCO Data In Register ..... 404
9.11.4 TCO_DAT_OUT—TCO Data Out Register ..... 404
9.11.5 TCO1_STS—TCO1 Status Register ..... 405
9.11.6 TCO2_STS—TCO2 Status Register ..... 406
9.11.7 TCO1_CNT—TCO1 Control Register ..... 407
9.11.8 TCO2_CNT—TCO2 Control Register ..... 408
9.11.9 TCO_MESSAGE1 and TCO_MESSAGE2 Registers ..... 408
9.11.10 TCO_WDSTS—TCO Watchdog Status Register ..... 408
9.11.11 SW_IRQ_GEN—Software IRQ Generation Register ..... 409
9.12 General Purpose I/O Registers (D31:F0) ..... 409
9.12.1 GPIO_USE_SEL—GPIO Use Select Register ..... 410
9.12.2 GP_IO_SEL—GPIO Input/Output Select Register ..... 410
9.12.3 GP_LVL—GPIO Level for Input or Output Register ..... 411
9.12.4 GPO_BLINK—GPO Blink Enable Register ..... 411
9.12.5 GPI_INV—GPIO Signal Invert Register ..... 412
9.12.6 GPIO_USE_SEL2—GPIO Use Select 2 Register ..... 412
9.12.7 GP_IO_SEL2—GPIO Input/Output Select 2 Register ..... 413
9.12.8 GP_LVL2—GPIO Level for Input or Output 2 Register ..... 414

**10 IDE Controller Registers (D31:F1)** ..... 415

10.1 PCI Configuration Registers (IDE—D31:F1) ..... 415
10.1.1 VID—Vendor Identification Register (IDE—D31:F0) ..... 416
10.1.2 DID—Device Identification Register (IDE—D31:F0) ..... 416
10.1.3 PCICMD—PCI Command Register (IDE—D31:F1) ..... 417
10.1.4 PCISTS — PCI Status Register (IDE—D31:F1) ..... 418
10.1.5 RID—Revision Identification Register (IDE—D31:F1) ..... 419
10.1.6 PI—Programming Interface Register (IDE—D31:F1) ..... 419
10.1.7 SCC—Sub Class Code Register (IDE—D31:F1) ..... 419
10.1.8 BCC—Base Class Code Register (IDE—D31:F1) ..... 420
10.1.9 PMLT—Primary Master Latency Timer Register (IDE—D31:F1) ..... 420
10.1.10 PCMD_BAR—Primary Command Block Base Address Register (IDE—D31:F1) ..... 420
10.1.11 PCNL_BAR—Primary Control Block Base Address Register (IDE—D31:F1) ..... 421
10.1.12 SCMD_BAR—Secondary Command Block Base Address Register (IDE D31:F1) ..... 421
10.1.13 SCNL_BAR—Secondary Control Block Base Address Register (IDE D31:F1) ..... 421

10.1.14 BM_BASE — Bus Master Base Address Register (IDE—D31:F1) ..... 422
10.1.15 IDE_SVID — Subsystem Vendor Identification (IDE—D31:F1) ..... 422
10.1.16 IDE_SID — Subsystem Identification Register (IDE—D31:F1) ..... 422
10.1.17 INTR_LN—Interrupt Line Register (IDE—D31:F1) ..... 423
10.1.18 INTR_PN—Interrupt Pin Register (IDE—D31:F1) ..... 423
10.1.19 IDE_TIM — IDE Timing Register (IDE—D31:F1) ..... 424
10.1.20 SLV_IDETIM—Slave (Drive 1) IDE Timing Register (IDE—D31:F1) ..... 425
10.1.21 SDMA_CNT—Synchronous DMA Control Register (IDE—D31:F1) ..... 427
10.1.22 SDMA_TIM—Synchronous DMA Timing Register (IDE—D31:F1) ..... 428
10.1.23 IDE_CONFIG—IDE I/O Configuration Register (IDE—D31:F1) ..... 429
10.2 Bus Master IDE I/O Registers (IDE—D31:F1) ..... 430
10.2.1 BMIC[P,S]—Bus Master IDE Command Register (IDE—D31:F1) ..... 431
10.2.2 BMIS[P,S]—Bus Master IDE Status Register (IDE—D31:F1) ..... 432
10.2.3 BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register (IDE—D31:F1) ..... 432

**11 SATA Controller Registers (D31:F2)** ..... 433

11.1 PCI Configuration Registers (SATA–D31:F2) ..... 433
11.1.1 VID—Vendor Identification Register (SATA—D31:F2) ..... 434
11.1.2 DID—Device Identification Register (SATA—D31:F2) ..... 434
11.1.3 PCICMD—PCI Command Register (SATA–D31:F2) ..... 435
11.1.4 PCISTS — PCI Status Register (SATA–D31:F2) ..... 436
11.1.5 RID—Revision Identification Register (SATA—D31:F2) ..... 437
11.1.6 PI—Programming Interface Register (SATA–D31:F2) ..... 437
11.1.7 SCC—Sub Class Code Register (SATA–D31:F2) ..... 437
11.1.8 BCC—Base Class Code Register (SATA–D31:F2) ..... 438
11.1.9 PMLT—Primary Master Latency Timer Register (SATA–D31:F2) ..... 438
11.1.10 PCMD_BAR—Primary Command Block Base Address Register (SATA–D31:F2) ..... 438
11.1.11 PCNL_BAR—Primary Control Block Base Address Register (SATA–D31:F2) ..... 439

11.1.12 SCMD_BAR—Secondary Command Block Base Address Register (IDE D31:F1) ..... 439
11.1.13 SCNL_BAR—Secondary Control Block Base Address Register (IDE D31:F1) ..... 439
11.1.14 BAR — Legacy Bus Master Base Address Register (SATA–D31:F2) ..... 440
11.1.15 SVID—Subsystem Vendor Identification Register (SATA–D31:F2) ..... 440
11.1.16 SID—Subsystem Identification Register (SATA–D31:F2) ..... 440
11.1.17 CAP—Capabilities Pointer Register (SATA–D31:F2) ..... 441
11.1.18 INT_LN—Interrupt Line Register (SATA–D31:F2) ..... 441
11.1.19 INT_PN—Interrupt Pin Register (SATA–D31:F2) ..... 441
11.1.20 IDE_TIM — IDE Timing Register (SATA–D31:F2) ..... 442
11.1.21 SIDETIM—Slave IDE Timing Register (SATA–D31:F2) ..... 444
11.1.22 SDMA_CNT—Synchronous DMA Control Register (SATA–D31:F2) ..... 445
11.1.23 SDMA_TIM—Synchronous DMA Timing Register (SATA–D31:F2) ..... 446
11.1.24 IDE_CONFIG—IDE I/O Configuration Register (SATA–D31:F2) ..... 447
11.1.25 PID—PCI Power Management Capability Identification Register (SATA–D31:F2) ..... 448
11.1.26 PC—PCI Power Management Capabilities Register (SATA–D31:F2) ..... 448
11.1.27 PMCS—PCI Power Management Control and Status Register (SATA–D31:F2) ..... 449
11.1.28 MID—Message Signaled Interrupt Identifiers Register (SATA–D31:F2) ..... 449
11.1.29 MC—Message Signaled Interrupt Message Control Register (SATA–D31:F2) ..... 450
11.1.30 MA—Message Signaled Interrupt Message Address Register (SATA–D31:F2) ..... 450
11.1.31 MD—Message Signaled Interrupt Message Data Register (SATA–D31:F2) ..... 450
11.1.32 MAP—Address Map Register (SATA–D31:F2) ..... 451
11.1.33 PCS—Port Control and Status Register (SATA–D31:F2) ..... 451
11.1.34 SRI—SATA Registers Index (SATA–D31:F2) ..... 452
11.1.35 SRD—SATA Registers Data (SATA–D31:F2) ..... 452
11.1.36 SIRA—SATA Initialization Register A (SATA–D31:F2) ..... 452
11.1.37 SIRB — SATA Initialization Register B (SATA–D31:F2) ..... 453
11.1.38 PMR0 — Power Management Register Port 0 (SATA–D31:F2) ..... 453

11.1.39 PMR1 — Power Management Register Port 1 (SATA–D31:F2) ..... 453
11.1.40 BFCS—BIST FIS Control/Status Register (SATA–D31:F2) ..... 454
11.1.41 BFTD1—BIST FIS Transmit Data1 Register (SATA–D31:F2) ..... 455
11.1.42 BFTD2—BIST FIS Transmit Data2 Register (SATA–D31:F2) ..... 455
11.2 Bus Master IDE I/O Registers (D31:F2) ..... 456
11.2.1 BMIC[P,S]—Bus Master IDE Command Register (D31:F2) ..... 457
11.2.2 BMIS[P,S]—Bus Master IDE Status Register (D31:F2) ..... 458
11.2.3 BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register (D31:F2) ..... 459

**12 UHCI Controllers Registers** ..... 461

12.1 PCI Configuration Registers (USB—D29:F0/F1/F2/F3) ..... 461
12.1.1 VID—Vendor Identification Register (USB—D29:F0/F1/F2/F3) ..... 462
12.1.2 DID—Device Identification Register (USB—D29:F0/F1/F2/F3) ..... 462
12.1.3 PCICMD—PCI Command Register (USB—D29:F0/F1/F2/F3) ..... 462
12.1.4 PCISTS—PCI Status Register (USB—D29:F0/F1/F2/F3) ..... 463
12.1.5 RID—Revision Identification Register (USB—D29:F0/F1/F2/F3) ..... 463
12.1.6 PI—Programming Interface Register (USB—D29:F0/F1/F2/F3) ..... 464
12.1.7 SCC—Sub Class Code Register (USB—D29:F0/F1/F2/F3) ..... 464
12.1.8 BCC—Base Class Code Register (USB—D29:F0/F1/F2/F3) ..... 464
12.1.9 HEADTYP—Header Type Register (USB—D29:F0/F1/F2/F3) ..... 465
12.1.10 BASE—Base Address Register (USB—D29:F0/F1/F2/F3) ..... 465
12.1.11 SVID — Subsystem Vendor Identification Register (USB—D29:F0/F1/F2/F3) ..... 466
12.1.12 SID — Subsystem Identification Register (USB—D29:F0/F1/F2/F3) ..... 466
12.1.13 INT_LN—Interrupt Line Register (USB—D29:F0/F1/F2/F3) ..... 466
12.1.14 INT_PN—Interrupt Pin Register (USB—D29:F0/F1/F2/F3) ..... 467
12.1.15 USB_RELNUM—Serial Bus Release Number Register (USB—D29:F0/F1/F2/F3) ..... 467
12.1.16 USB_LEGKEY—USB Legacy Keyboard/Mouse Control Register (USB—D29:F0/F1/F2/F3) ..... 468

12.1.17 USB_RES—USB Resume Enable Register (USB—D29:F0/F1/F2/F3) ..... 470
12.2 USB I/O Registers ..... 470
12.2.1 USBCMD—USB Command Register ..... 471
12.2.2 USBSTS—USB Status Register ..... 474
12.2.3 USBINTR—USB Interrupt Enable Register ..... 475
12.2.4 FRNUM—Frame Number Register ..... 475
12.2.5 FRBASEADD—Frame List Base Address Register ..... 476
12.2.6 SOFMOD—Start of Frame Modify Register ..... 476
12.2.7 PORTSC[0,1]—Port Status and Control Register ..... 477

**13 EHCI Controller Registers (D29:F7) ..... 479**

13.1 USB EHCI Configuration Registers (USB EHCI—D29:F7) ..... 479
13.1.1 VID—Vendor Identification Register (USB EHCI—D29:F7) ..... 480
13.1.2 DID—Device Identification Register (USB EHCI—D29:F7) ..... 480
13.1.3 PCICMD—PCI Command Register (USB EHCI—D29:F7) ..... 481
13.1.4 PCISTS—PCI Status Register (USB EHCI—D29:F7) ..... 482
13.1.5 RID—Revision Identification Register (USB EHCI—D29:F7) ..... 483
13.1.6 PI—Programming Interface Register (USB EHCI—D29:F7) ..... 483
13.1.7 SCC—Sub Class Code Register (USB EHCI—D29:F7) ..... 483
13.1.8 BCC—Base Class Code Register (USB EHCI—D29:F7) ..... 483
13.1.9 PMLT—Primary Master Latency Timer Register (USB EHCI—D29:F7) ..... 484
13.1.10 MEM_BASE—Memory Base Address Register (USB EHCI—D29:F7) ..... 484
13.1.11 SVID—USB EHCI Subsystem Vendor ID Register (USB EHCI—D29:F7) ..... 484
13.1.12 SID—USB EHCI Subsystem ID Register (USB EHCI—D29:F7) ..... 485
13.1.13 CAP_PTR—Capabilities Pointer Register (USB EHCI—D29:F7) ..... 485
13.1.14 INT_LN—Interrupt Line Register (USB EHCI—D29:F7) ..... 485
13.1.15 INT_PN—Interrupt Pin Register (USB EHCI—D29:F7) ..... 485
13.1.16 PWR_CAPID—PCI Power Management Capability ID Register (USB EHCI—D29:F7) ..... 486
13.1.17 NXT_PTR1—Next Item Pointer #1 Register (USB EHCI—D29:F7) ..... 486
13.1.18 PWR_CAP—Power Management Capabilities Register (USB EHCI—D29:F7) ..... 487
13.1.19 PWR_CNTL_STS—Power Management Control/Status Register (USB EHCI—D29:F7) ..... 488

13.1.20 DEBUG_CAPID—Debug Port Capability ID Register (USB EHCI—D29:F7) ..... 488
13.1.21 NXT_PTR2—Next Item Pointer #2 Register (USB EHCI—D29:F7) ..... 489
13.1.22 DEBUG_BASE—Debug Port Base Offset Register (USB EHCI—D29:F7) ..... 489
13.1.23 USB_RELNUM—USB Release Number Register (USB EHCI—D29:F7) ..... 489
13.1.24 FL_ADJ—Frame Length Adjustment Register (USB EHCI—D29:F7) ..... 490
13.1.25 PWAKE_CAP—Port Wake Capability Register (USB EHCI—D29:F7) ..... 491
13.1.26 LEG_EXT_CAP—USB EHCI Legacy Support Extended Capability Register (USB EHCI—D29:F7) ..... 491
13.1.27 LEG_EXT_CS—USB EHCI Legacy Support Extended Control / Status Register (USB EHCI—D29:F7) ..... 492
13.1.28 SPECIAL_SMI—Intel Specific USB 2.0 SMI Register (USB EHCI—D29:F7) ..... 493
13.1.29 ACCESS_CNTL—Access Control Register (USB EHCI—D29:F7) ..... 495
13.2 Memory-Mapped I/O Registers ..... 496
13.2.1 CAPLENGTH—Capability Registers Length Register ..... 496
13.2.2 HCIVERSION—Host Controller Interface Version Number Register ..... 497
13.2.3 HCSPARAMS—Host Controller Structural Parameters ..... 497
13.2.4 HCCPARAMS—Host Controller Capability Parameters Register ..... 498
13.2.5 USB2.0_CMD—USB 2.0 Command Register ..... 500
13.2.6 USB2.0_STS—USB 2.0 Status Register ..... 502
13.2.7 USB2.0_INTR—USB 2.0 Interrupt Enable Register ..... 504
13.2.8 FRINDEX—Frame Index Register ..... 505
13.2.9 CTRLDSSEGMENT—Control Data Structure Segment Register ..... 506
13.2.10 PERIODICLISTBASE—Periodic Frame List Base Address Register ..... 506
13.2.11 ASYNCLISTADDR—Current Asynchronous List Address Register ..... 507
13.2.12 CONFIGFLAG—Configure Flag Register ..... 507
13.2.13 PORTSC—Port N Status and Control Register ..... 508
13.2.14 CNTL_STS—Control/Status Register ..... 512
13.2.15 USBPID—USB PIDs Register ..... 514
13.2.16 DATABUF[7:0]—Data Buffer Bytes[7:0] Register ..... 514
13.2.17 CONFIG—Configuration Register ..... 514

**14 SMBus Controller Registers (D31:F3)** ..... 515

14.1 PCI Configuration Registers (SMBUS—D31:F3) ..... 515
14.1.1 VID—Vendor Identification Register (SMBUS—D31:F3) ..... 515
14.1.2 DID—Device Identification Register (SMBUS—D31:F3) ..... 516
14.1.3 PCICMD—PCI Command Register (SMBUS—D31:F3) ..... 516

14.1.4 PCISTS—PCI Status Register (SMBUS—D31:F3) ..... 517
14.1.5 RID—Revision Identification Register (SMBUS—D31:F3) ..... 517
14.1.6 SCC—Sub Class Code Register (SMBUS—D31:F3) ..... 518
14.1.7 BCC—Base Class Code Register (SMBUS—D31:F3) ..... 518
14.1.8 SMB_BASE—SMBUS Base Address Register (SMBUS—D31:F3) ..... 518
14.1.9 SVID — Subsystem Vendor Identification Register (SMBUS—D31:F2/F4) ..... 518
14.1.10 SID — Subsystem Identification Register (SMBUS—D31:F2/F4) ..... 519
14.1.11 INT_LN—Interrupt Line Register (SMBUS—D31:F3) ..... 519
14.1.12 INT_PN—Interrupt Pin Register (SMBUS—D31:F3) ..... 519
14.1.13 HOSTC—Host Configuration Register (SMBUS—D31:F3) ..... 520
14.2 SMBus I/O Registers ..... 521
14.2.1 HST_STS—Host Status Register (SMBUS—D31:F3) ..... 522
14.2.2 HST_CNT—Host Control Register (SMBUS—D31:F3) ..... 523
14.2.3 HST_CMD—Host Command Register (SMBUS—D31:F3) ..... 525
14.2.4 XMIT_SLVA—Transmit Slave Address Register (SMBUS—D31:F3) ..... 525
14.2.5 HST_D0—Host Data 0 Register (SMBUS—D31:F3) ..... 525
14.2.6 HST_D1—Host Data 1 Register (SMBUS—D31:F3) ..... 525
14.2.7 Host_BLOCK_DB—Host Block Data Byte Register (SMBUS—D31:F3) ..... 526
14.2.8 PEC—Packet Error Check (PEC) Register (SMBUS—D31:F3) ..... 526
14.2.9 RCV_SLVA—Receive Slave Address Register (SMBUS—D31:F3) ..... 527
14.2.10 SLV_DATA—Receive Slave Data Register (SMBUS—D31:F3) ..... 527
14.2.11 AUX_STS—Auxiliary Status Register (SMBUS—D31:F3) ..... 527
14.2.12 AUX_CTL—Auxiliary Control Register (SMBUS—D31:F3) ..... 528
14.2.13 SMLINK_PIN_CTL—SMLink Pin Control Register (SMBUS—D31:F3) ..... 528
14.2.14 SMBUS_PIN_CTL—SMBUS Pin Control Register (SMBUS—D31:F3) ..... 529
14.2.15 SLV_STS—Slave Status Register (SMBUS—D31:F3) ..... 529

14.2.16 SLV_CMD—Slave Command Register (SMBUS—D31:F3) ..... 530
14.2.17 NOTIFY_DADDR—Notify Device Address Register (SMBUS—D31:F3) ..... 530
14.2.18 NOTIFY_DLOW—Notify Data Low Byte Register (SMBUS—D31:F3) ..... 531
14.2.19 NOTIFY_DHIGH—Notify Data High Byte Register (SMBUS—D31:F3) ..... 531

**15 AC '97 Audio Controller Registers (D31:F5)** ..... 533

15.1 AC '97 Audio PCI Configuration Space (Audio— D31:F5) ..... 533
15.1.1 VID—Vendor Identification Register (Audio—D31:F5) ..... 534
15.1.2 DID—Device Identification Register (Audio—D31:F5) ..... 534
15.1.3 PCICMD—PCI Command Register (Audio—D31:F5) ..... 535
15.1.4 PCISTS—PCI Status Register (Audio—D31:F5) ..... 536
15.1.5 RID—Revision Identification Register (Audio—D31:F5) ..... 537
15.1.6 PI—Programming Interface Register (Audio—D31:F5) ..... 537
15.1.7 SCC—Sub Class Code Register (Audio—D31:F5) ..... 537
15.1.8 BCC—Base Class Code Register (Audio—D31:F5) ..... 537
15.1.9 HEADTYP—Header Type Register (Audio—D31:F5) ..... 538
15.1.10 NAMBAR—Native Audio Mixer Base Address Register (Audio—D31:F5) ..... 538
15.1.11 NABMBAR—Native Audio Bus Mastering Base Address Register (Audio—D31:F5) ..... 539
15.1.12 MMBAR—Mixer Base Address Register (Audio—D31:F5) ..... 539
15.1.13 MBBAR—Bus Master Base Address Register (Audio—D31:F5) ..... 540
15.1.14 SVID—Subsystem Vendor Identification Register (Audio—D31:F5) ..... 540
15.1.15 SID—Subsystem Identification Register (Audio—D31:F5) ..... 541
15.1.16 CAP_PTR—Capabilities Pointer Register (Audio—D31:F5) ..... 541
15.1.17 INT_LN—Interrupt Line Register (Audio—D31:F5) ..... 541
15.1.18 INT_PN—Interrupt Pin Register (Audio—D31:F5) ..... 542
15.1.19 PCID—Programmable Codec Identification Register (Audio—D31:F5) ..... 542
15.1.20 CFG—Configuration Register (Audio—D31:F5) ..... 543

15.1.21 PID—PCI Power Management Capability Identification Register (Audio—D31:F5) ..... 543
15.1.22 PC—Power Management Capabilities Register (Audio—D31:F5) ..... 544
15.1.23 PCS—Power Management Control and Status Register (Audio—D31:F5) ..... 545
15.2 AC ’97 Audio I/O Space (D31:F5) ..... 546
15.2.1 x_BDBAR—Buffer Descriptor Base Address Register (Audio—D31:F5) ..... 549
15.2.2 x_CIV—Current Index Value Register (Audio—D31:F5) ..... 550
15.2.3 x_LVI—Last Valid Index Register (Audio—D31:F5) ..... 550
15.2.4 x_SR—Status Register (Audio—D31:F5) ..... 551
15.2.5 x_PICB—Position In Current Buffer Register (Audio—D31:F5) ..... 552
15.2.6 x_PIV—Prefetched Index Value Register (Audio—D31:F5) ..... 552
15.2.7 x_CR—Control Register (Audio—D31:F5) ..... 553
15.2.8 GLOB_CNT—Global Control Register (Audio—D31:F5) ..... 554
15.2.9 GLOB_STA—Global Status Register (Audio—D31:F5) ..... 556
15.2.10 CAS—Codec Access Semaphore Register (Audio—D31:F5) ..... 558
15.2.11 SDM—SDATA_IN Map Register (Audio—D31:F5) ..... 558

**16 AC ’97 Modem Controller Registers (D31:F6)** ..... 559

16.1 AC ’97 Modem PCI Configuration Space (D31:F6) ..... 559
16.1.1 VID—Vendor Identification Register (Modem—D31:F6) ..... 560
16.1.2 DID—Device Identification Register (Modem—D31:F6) ..... 560
16.1.3 PCICMD—PCI Command Register (Modem—D31:F6) ..... 561
16.1.4 PCISTS—PCI Status Register (Modem—D31:F6) ..... 562
16.1.5 RID—Revision Identification Register (Modem—D31:F6) ..... 562
16.1.6 PI—Programming Interface Register (Modem—D31:F6) ..... 563
16.1.7 SCC—Sub Class Code Register (Modem—D31:F6) ..... 563
16.1.8 BCC—Base Class Code Register (Modem—D31:F6) ..... 563
16.1.9 HEADTYP—Header Type Register (Modem—D31:F6) ..... 563
16.1.10 MMBAR—Modem Mixer Base Address Register (Modem—D31:F6) ..... 564

16.1.11 MBAR—Modem Base Address Register (Modem—D31:F6) ..... 564
16.1.12 SVID—Subsystem Vendor Identification Register (Modem—D31:F6) ..... 565
16.1.13 SID—Subsystem Identification Register (Modem—D31:F6) ..... 565
16.1.14 CAP_PTR—Capabilities Pointer Register (Modem—D31:F6) ..... 565
16.1.15 INT_LN—Interrupt Line Register (Modem—D31:F6) ..... 566
16.1.16 INT_PIN—Interrupt Pin Register (Modem—D31:F6) ..... 566
16.1.17 PID—PCI Power Management Capability Identification Register (Modem—D31:F6) ..... 566
16.1.18 PC—Power Management Capabilities Register (Modem—D31:F6) ..... 567
16.1.19 PCS—Power Management Control and Status Register (Modem—D31:F6) ..... 567
16.2 AC ’97 Modem I/O Space (D31:F6) ..... 568
16.2.1 x_BDBAR—Buffer Descriptor List Base Address Register (Modem—D31:F6) ..... 570
16.2.2 x_CIV—Current Index Value Register (Modem—D31:F6) ..... 570
16.2.3 x_LVI—Last Valid Index Register (Modem—D31:F6) ..... 571
16.2.4 x_SR—Status Register (Modem—D31:F6) ..... 572
16.2.5 x_PICB—Position in Current Buffer Register (Modem—D31:F6) ..... 573
16.2.6 x_PIV—Prefetch Index Value Register (Modem—D31:F6) ..... 573
16.2.7 x_CR—Control Register (Modem—D31:F6) ..... 574
16.2.8 GLOB_CNT—Global Control Register (Modem—D31:F6) ..... 575
16.2.9 GLOB_STA—Global Status Register (Modem—D31:F6) ..... 576
16.2.10 CAS—Codec Access Semaphore Register (Modem—D31:F6) ..... 578

**17 High-Precision Event Timer Registers** ..... 579

17.1 GCAP_ID—General Capabilities and Identification Register ..... 581
17.2 GEN_CONF—General Configuration Register ..... 581
17.3 GINTR_STA—General Interrupt Status Register ..... 582
17.4 MAIN_CNT—Main Counter Value Register ..... 582
17.5 TIMn_CONF—Timer n Configuration and Capabilities Register ..... 583
17.6 TIMn_COMP—Timer n Comparator Value Register ..... 585

**18 Ballout Definition** ..... 587

**19 Electrical Characteristics** ..... 599

19.1 Thermal Specifications ..... 599
19.2 DC Characteristics ..... 599
19.3 AC Characteristics ..... 606
19.4 Timing Diagrams ..... 619

**20 Package Information** ..... 629

**21 Testability** ..... 631

21.1 Test Mode Description ..... 631
21.2 Tri-State Mode ..... 632
21.3 XOR Chain Mode ..... 632
21.3.1 XOR Chain Testability Algorithm Example ..... 632

**A Register Index** ..... 639

**B Register Bit Index** ..... 661

# Figures


n System Block Diagram ..... 4
1 Intel® ICH5 Interface Signals Block Diagram ..... 50
2 Example External RTC Circuit ..... 67
3 Example V5REF Sequencing Circuit ..... 67
4 Conceptual System Clock Diagram ..... 78
5 Primary Device Status Register Error Reporting Logic ..... 81
6 Secondary Status Register Error Reporting Logic ..... 81
7 NMI# Generation Logic ..... 82
8 Integrated LAN Controller Block Diagram ..... 85
9 64-Word EEPROM Read Instruction Waveform ..... 96
10 LPC Interface Diagram ..... 103
11 Typical Timing for LFRAME# ..... 107
12 Abort Mechanism ..... 107
13 Intel® ICH5 DMA Controller ..... 109
14 DMA Serial Channel Passing Protocol ..... 113
15 DMA Request Assertion through LDRQ# ..... 116
16 Coprocessor Error Timing Diagram ..... 142
17 Signal Strapping ..... 145
18 Physical Region Descriptor Table Entry ..... 178
19 SATA Power States ..... 187
20 Transfer Descriptor ..... 193
21 Example Queue Conditions ..... 201
22 USB Data Encoding ..... 204
23 USB Legacy Keyboard Flow Diagram ..... 213
24 Intel® ICH5-USB Port Connections ..... 223
25 Intel® ICH5-Based Audio Codec ’97 Specification, Version 2.3 ..... 252
26 AC ’97 2.3 Controller-Codec Connection ..... 254
27 AC-Link Protocol ..... 255
28 AC-Link Powerdown Timing ..... 262
29 SDIN Wake Signaling ..... 263
1 Intel® ICH5 Ballout (Topview–Left Side) ..... 588
2 Intel® ICH5 Ballout (Topview–Right Side) ..... 589
3 Clock Timing ..... 619
4 Valid Delay from Rising Clock Edge ..... 619
5 Setup and Hold Times ..... 619
6 Float Delay ..... 620
7 Pulse Width ..... 620
8 Output Enable Delay ..... 620
9 IDE PIO Mode ..... 621
10 IDE Multiword DMA ..... 621
11 Ultra ATA Mode (Drive Initiating a Burst Read) ..... 622
12 Ultra ATA Mode (Sustained Burst) ..... 622
13 Ultra ATA Mode (Pausing a DMA Burst) ..... 623
14 Ultra ATA Mode (Terminating a DMA Burst) ..... 623
15 USB Rise and Fall Times ..... 623
16 USB Jitter ..... 624
17 USB EOP Width ..... 624
18 SMBus Transaction ..... 624
19 SMBus Timeout ..... 625

20 Power Sequencing and Reset Signal Timings ..... 625
21 G3 (Mechanical Off) to S0 Timings ..... 626
22 S0 to S1 to S0 Timing ..... 626
23 S0 to S5 to S0 Timings ..... 627
24 AC'97 Data Input and Output Timings ..... 627
25 Intel® ICH5 Package (Top and Side Views) ..... 629
26 Intel® ICH5 Package (Bottom View) ..... 630
27 Test Mode Entry (XOR Chain Example) ..... 631
28 Example XOR Chain Circuitry ..... 632

# Tables


1 Industry Specifications ..... 39
2 PCI Devices and Functions ..... 42
3 Hub Interface Signals ..... 51
4 LAN Connect Interface Signals ..... 51
5 EEPROM Interface Signals ..... 51
6 Flash BIOS Interface Signals ..... 52
7 PCI Interface Signals ..... 52
8 Serial ATA Interface Signals ..... 54
9 IDE Interface Signals ..... 55
10 LPC Interface Signals ..... 56
11 Interrupt Signals ..... 57
12 USB Interface Signals ..... 58
13 Power Management Interface Signals ..... 59
14 Processor Interface Signals ..... 60
15 SM Bus Interface Signals ..... 61
16 System Management Interface Signals ..... 61
17 Real Time Clock Interface ..... 62
18 Other Clocks ..... 62
19 Miscellaneous Signals ..... 62
20 AC-Link Signals ..... 63
21 General Purpose I/O Signals ..... 63
22 Power and Ground Signals ..... 65
23 Functional Strap Definitions ..... 66
24 Test Mode Selection ..... 68
25 Intel® ICH5 Power Planes ..... 69
26 Integrated Pull-Up and Pull-Down Resistors ..... 70
27 IDE Series Termination Resistors ..... 71
28 Power Plane and States for Output and I/O Signal ..... 72
29 Power Plane for Input Signals ..... 75
30 Intel® ICH5 and System Clock Domains ..... 77
31 Type 0 Configuration Cycle Device Number Translation ..... 83
32 Advanced TCO Functionality ..... 98
33 LPC Cycle Types Supported ..... 104
34 Start Field Bit Definitions ..... 104
35 Cycle Type Bit Definitions ..... 105
36 Transfer Size Bit Definition ..... 105
37 SYNC Bit Definition ..... 106
38 Intel® ICH5 Response to Sync Failures ..... 106
39 DMA Transfer Size ..... 111
40 Address Shifting in 16-Bit I/O DMA Transfers ..... 111
41 DMA Cycle vs. I/O Address ..... 115
42 PCI Data Bus vs. DMA I/O Port Size ..... 115
43 DMA I/O Cycle Width vs. BE[3:0]# ..... 115
44 Counter Operating Modes ..... 121
45 Interrupt Controller Core Connections ..... 123
46 Interrupt Status Registers ..... 124
47 Content of Interrupt Vector Byte ..... 124
48 APIC Interrupt Mapping ..... 130
49 Interrupt Message Address Format ..... 134

50 Interrupt Message Data Format ..... 134
51 Stop Frame Explanation ..... 136
52 Data Frame Format ..... 137
53 Configuration Bits Reset by RTCRST# Assertion ..... 140
54 INIT# Going Active ..... 142
55 NMI Sources ..... 143
56 DP Signal Differences ..... 143
57 Frequency Strap Behavior Based on Exit State ..... 144
58 Frequency Strap Bit Mapping ..... 145
59 General Power States for Systems Using Intel® ICH5 ..... 146
60 State Transition Rules for Intel® ICH5 ..... 147
61 System Power Plane ..... 148
62 Causes of SMI# and SCI ..... 149
63 Sleep Types ..... 152
64 Causes of Wake Events ..... 153
65 GPI Wake Events ..... 153
66 Transitions Due to Power Failure ..... 154
67 Transitions Due to Power Button ..... 156
68 Transitions Due to RI# Signal ..... 157
69 Write Only Registers with Read Paths in ALT Access Mode ..... 159
70 PIC Reserved Bits Return Values ..... 161
71 Register Write Accesses in ALT Access Mode ..... 161
72 Intel® ICH5 Clock Inputs ..... 163
73 Heartbeat Message Data ..... 169
74 GPIO Implementation ..... 170
75 IDE Legacy I/O Ports: Command Block Registers (CS1x# Chip Select) ..... 175
76 IDE Transaction Timings (PCI Clocks) ..... 176
77 Interrupt/Active Bit Interaction Definition ..... 180
78 UltraATA/33 Control Signal Redefinitions ..... 181
79 SATA MSI vs. PCI IRQ Actions ..... 188
80 Legacy Routing ..... 189
81 Frame List Pointer Bit Description ..... 192
82 TD Link Pointer ..... 193
83 TD Control and Status ..... 194
84 TD Token ..... 196
85 TD Buffer Pointer ..... 196
86 Queue Head Block ..... 197
87 Queue Head Link Pointer ..... 197
88 Queue Element Link Pointer ..... 197
89 Command Register, Status Register and TD Status Bit Interaction ..... 200
90 Queue Advance Criteria ..... 202
91 USB Schedule List Traversal Decision Table ..... 203
92 PID Format ..... 205
93 PID Types ..... 205
94 Address Field ..... 206
95 Endpoint Field ..... 206
96 Token Format ..... 207
97 SOF Packet ..... 207
98 Data Packet Format ..... 208
99 Bits Maintained in Low Power States ..... 212

100 USB Legacy Keyboard State Transitions ..... 214
101 UHCI vs. EHCI ..... 215
102 Debug Port Behavior ..... 227
103 Quick Protocol ..... 232
104 Send / Receive Byte Protocol without PEC ..... 232
105 Send/Receive Byte Protocol with PEC ..... 233
106 Write Byte/Word Protocol without PEC ..... 233
107 Write Byte/Word Protocol with PEC ..... 234
108 Read Byte/Word Protocol without PEC ..... 235
109 Read Byte/Word Protocol with PEC ..... 235
110 Process Call Protocol without PEC ..... 236
111 Process Call Protocol with PEC ..... 237
112 Block Read/Write Protocol without PEC ..... 238
113 Block Read/Write Protocol with PEC ..... 239
114 $I^2C$ Block Read ..... 240
115 Block Write–Block Read Process Call Protocol with/without PEC ..... 241
116 Enable for SMBALERT# ..... 244
117 Enables for SMBus Slave Write and SMBus Host Events ..... 244
118 Enables for the Host Notify Command ..... 244
119 Slave Write Cycle Format ..... 246
120 Slave Write Registers ..... 246
121 Command Types ..... 247
122 Read Cycle Format ..... 248
123 Data Values for Slave Read Registers ..... 249
124 Host Notify Format ..... 250
125 Features Supported by Intel® ICH5 ..... 251
126 AC '97 Signals ..... 254
127 Input Slot 1 Bit Definitions ..... 260
128 Output Tag Slot 0 ..... 262
129 AC-link State during PCIRST# ..... 264
130 PCI Devices and Functions ..... 268
131 Fixed I/O Ranges Decoded by Intel® ICH5 ..... 270
132 Variable I/O Decode Ranges ..... 272
133 Memory Decode Ranges from Processor Perspective ..... 273
134 LAN Controller PCI Register Address Map (LAN Controller—B1:D8:F0) ..... 275
135 Configuration of Subsystem ID and Subsystem Vendor ID via EEPROM ..... 282
136 Data Register Structure ..... 286
137 Intel® ICH5 Integrated LAN Controller CSR Space Register Address Map ..... 287
138 Self-Test Results Format ..... 292
139 Statistical Counters ..... 299
140 Hub Interface PCI Register Address Map (HUB-PCI—D30:F0) ..... 301
141 LPC Interface PCI Register Address Map (LPC I/F—D31:F0) ..... 317
142 DMA Registers ..... 344
143 PIC Registers (LPC I/F—D31:F0) ..... 355
144 APIC Direct Registers (LPC I/F—D31:F0) ..... 363
145 APIC Indirect Registers (LPC I/F—D31:F0) ..... 363
146 RTC I/O Registers (LPC I/F—D31:F0) ..... 369
147 RTC (Standard) RAM Bank (LPC I/F—D31:F0) ..... 369
148 Processor Interface PCI Register Address Map (LPC I/F—D31:F0) ..... 373
149 Power Management PCI Register Address Map (PM—D31:F0) ..... 376

150 APM Register Map ..... 384
151 ACPI and Legacy I/O Register Map ..... 385
152 TCO I/O Register Address Map ..... 403
153 Registers to Control GPIO Address Map ..... 409
154 IDE Controller PCI Register Address Map (IDE-D31:F1) ..... 415
155 Bus Master IDE I/O Registers ..... 430
156 SATA Controller PCI Register Address Map (SATA–D31:F2) ..... 433
157 SATA Indexed Registers ..... 452
158 Bus Master IDE I/O Register Address Map ..... 456
159 UHCI Controller PCI Register Address Map (USB—D29:F0/F1/F2/F3) ..... 461
160 USB I/O Registers ..... 470
161 Run/Stop, Debug Bit Interaction SWDBG (Bit 5), Run/Stop (Bit 0) Operation ..... 473
162 USB EHCI PCI Register Address Map (USB EHCI—D29:F7) ..... 479
163 Enhanced Host Controller Capability Registers ..... 496
164 Enhanced Host Controller Operational Register Address Map ..... 499
165 Debug Port Register Address Map ..... 512
166 SMBus Controller PCI Register Address Map (SMBUS—D31:F3) ..... 515
167 SMBus I/O Register Address Map ..... 521
168 AC ‘97 Audio PCI Register Address Map (Audio—D31:F5) ..... 533
169 Intel® ICH5 Audio Mixer Register Configuration ..... 546
170 Native Audio Bus Master Control Registers ..... 548
171 AC ‘97 Modem PCI Register Address Map (Modem—D31:F6) ..... 559
172 Intel® ICH5 Modem Mixer Register Configuration ..... 568
173 Modem Registers ..... 569
174 Memory-Mapped Registers ..... 579
175 Intel® ICH5 Ballout by Signal Name ..... 590
176 Intel® ICH5 Ballout by Ball Number ..... 594
177 DC Current Characteristics ..... 599
178 DC Characteristic Input Signal Association ..... 600
179 DC Input Characteristics ..... 601
180 DC Characteristic Output Signal Association ..... 603
181 DC Output Characteristics ..... 604
182 Other DC Characteristics ..... 605
183 Clock Timings ..... 606
184 PCI Interface Timing ..... 608
185 IDE PIO and Multiword DMA ModeTiming ..... 609
186 Ultra ATA Timing (Mode 0, Mode 1, Mode 2) ..... 610
187 Ultra ATA Timing (Mode 3, Mode 4, Mode 5) ..... 612
188 Universal Serial Bus Timing ..... 614
189 SATA Interface Timings ..... 615
190 SMBus Timing ..... 615
192 LPC Timing ..... 616
193 Miscellaneous Timings ..... 616
191 AC’97 Timing ..... 616
194 Power Sequencing and Reset Signal Timings ..... 617
195 Power Management Timings ..... 618
196 Test Mode Selection ..... 631
197 XOR Test Pattern Example ..... 632
198 XOR Chain #1 (RTCRST# Asserted for 4 PCI Clocks While PWROK Active) ..... 633
199 XOR Chain #2 (RTCRST# Asserted for 5 PCI Clocks While PWROK Active) ..... 634

200 XOR Chain #3 (RTCRST# Asserted for 6 PCI Clocks While PWROK Active) ..... 635
201 XOR Chain #4-1 (RTCRST# Asserted for 7 PCI Clocks While PWROK Active) ..... 636
203 XOR Chain #6 (RTCRST# Asserted for 52 PCI Clocks While PWROK Active) ..... 637
202 XOR Chain #4-2 (RTCRST# Asserted for 7 PCI Clocks While PWROK Active) ..... 637
204 Intel® ICH5 PCI Configuration Registers ..... 639
205 Intel® ICH5 Fixed I/O Registers ..... 650
206 Intel® ICH5 Variable I/O Registers ..... 655

# Revision History

| Revision | Description | Date |
|---|---|---|
| -001 | Initial release | April 2003 |

# Introduction 1

## 1.1 About This Manual

This manual is intended for Original Equipment Manufacturers and BIOS vendors creating Intel® 82801EB ICH5 and Intel® 82801ER ICH5 R (ICH5R)-based products. Throughout this manual, all references to ICH5 refer to both ICH5 and ICH5R components, unless specifically noted otherwise. This manual assumes a working knowledge of the vocabulary and principles of USB, IDE, SATA, AC '97, 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 for the complete details.

**Table 1. Industry Specifications**

| Specification | Location |
|---|---|
| *Low Pin Count Interface Specification, Revision 1.1* (LPC) | http://developer.intel.com/design/chipsets/industry/lpc.htm |
| *Audio Codec '97 Component Specification, Version* 2.3 (also known as *AC '97 v2.3 Specification*) | http://www.intel.com/labs/media/audio/index.htm#97spec23 |
| *Wired for Management Baseline Version 2.0* (WfM) | http://www.intel.com/labs/manage/wfm/wfmspecs.htm |
| *System Management Bus (SMBus) Specification, Version 2.0* | http://www.smbus.org/specs/ |
| *PCI Local Bus Specification, Revision 2.3* (PCI) | http://www.pcisig.com/specifications/conventional |
| *PCI-to-PCI Bridge Architecture Specification, Revision 1.1* | http://www.pcisig.com/specifications/conventional |
| *PCI Power Management Specification, Revision 1.1* | http://www.pcisig.com/specifications/conventional |
| *Universal Serial Bus Revision 2.0 Specification* (USB) | http://www.usb.org/developers/docs/ |
| *Advanced Configuration and Power Interface, Version 2.0b* (ACPI) | http://www.acpi.info/spec.htm |
| *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* (EHCI) | http://developer.intel.com/technology/usb/ehcispec.htm |
| *SATA 1.0a Specification* (Serial ATA) | http://www.serialata.org/collateral/index.shtml |
| *Alert Standard Format (ASF) Specification, Version 1.03* | http://www.dmtf.org/standards/standard_alert.php |
| *IEEE 802.3* | http://standards.ieee.org |
| *AT Attachment - 6 with Packet Interface (ATA/ATAPI - 6)* | http://T13.org (T13 1410D) |
| *Power Management Network Device Class Reference Specification, Revision 1.0* | |

**Chapter 1. Introduction**
Chapter 1 introduces the ICH5 and provides information on manual organization.

**Chapter 2. Signal Description**
Chapter 2 provides a detailed description of each ICH5 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® ICH5 Power Planes and Pin States**
Chapter 3 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® ICH5 and System Clock Domains**
Chapter 4 provides a list of each clock domain associated with the ICH5 in an ICH5-based system.

**Chapter 5. Functional Description**
Chapter 5 provides a detailed description of the functions in the ICH5. 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, D29, D30 and D31 and functions as F0, F1, F2, F3, F4, F5, F6 and F7. For example Device 31 Function 5 is abbreviated as D31:F5, 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 the ICH5's external PCI bus is typically Bus 1, but may be assigned a different number depending upon system configuration.

**Chapter 6. Register and Memory Mappings**
Chapter 6 provides an overview of the registers, fixed I/O ranges, variable I/O ranges and memory ranges decoded by the ICH5.

**Chapter 7. LAN Controller Registers**
Chapter 7 provides a detailed description of all registers that reside in the ICH5's integrated LAN controller. The integrated LAN controller resides on the ICH5's external PCI bus (typically Bus 1) at Device 8, Function 0 (B1:D8:F0).

**Chapter 8. Hub Interface to PCI Bridge Registers**
Chapter 8 provides a detailed description of all registers that reside in the Hub Interface to PCI bridge. This bridge resides at Device 30, Function 0 (D30:F0).

**Chapter 9. LPC Bridge Registers**
Chapter 9 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 the ICH5 including DMA, Timers, Interrupts, Processor Interface, GPIO, Power Management, System Management and RTC.

**Chapter 10. IDE Controller Registers**
Chapter 10 provides a detailed description of all registers that reside in the IDE controller. This controller resides at Device 31, Function 1 (D31:F1).

**Chapter 11. SATA Controller Registers**
Chapter 11 provides a detailed description of all registers that reside in the SATA controller. This controller resides at Device 31, Function 2 (D31:F2).

**Chapter 12. UHCI Controller Registers**
Chapter 12 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 13. EHCI Controller Registers**
Chapter 13 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 14. SMBus Controller Registers**
Chapter 14 provides a detailed description of all registers that reside in the SMBus controller. This controller resides at Device 31, Function 3 (D31:F3).

**Chapter 15. AC '97 Audio Controller Registers**
Chapter 15 provides a detailed description of all registers that reside in the audio controller. This controller resides at Device 31, Function 5 (D31:F5). Note that this chapter of the datasheet does not include the native audio mixer registers. Accesses to the mixer registers are forwarded over the AC-link to the codec where the registers reside.

**Chapter 16. AC '97 Modem Controller Registers**
Chapter 16 provides a detailed description of all registers that reside in the modem controller. This controller resides at Device 31, Function 6 (D31:F6). Note that this chapter of the datasheet does not include the modem mixer registers. Accesses to the mixer registers are forwarded over the AC-link to the codec where the registers reside.

**Chapter 17. High-Precision Event Timers Registers**
Chapter 17 provides a detailed description of all registers that reside in the multimedia timer memory mapped register space.

**Chapter 18. Ballout Definition**
Chapter 18 provides a table of each signal and its ball assignment in the 460 mBGA package.

**Chapter 19. Electrical Characteristics**
Chapter 19 provides all AC and DC characteristics including detailed timing diagrams.

**Chapter 20. Package Information**
Chapter 20 provides drawings of the physical dimensions and characteristics of the 460-mBGA package.

**Chapter 21. Testability**
Chapter 21 provides detail about the implementation of test modes provided in the ICH5.

**Index**
This manual ends with indexes of registers and register bits.

## 1.2 Overview

The ICH5 provides extensive I/O support. Functions and capabilities include:

- *PCI Local Bus Specification, Revision 2.3* with support for 33 MHz PCI operations.
- PCI slots (supports up to 6 Req/Gnt pairs)
- ACPI power management logic support
- Enhanced DMA controller, interrupt controller, and timer functions
- Integrated IDE controller supports Ultra ATA100/66/33
- Integrated SATA controller
- 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
- Integrated ASF controller
- *System Management Bus (SMBus) Specification, Version 2.0* with additional support for $I^2C$ devices
- Supports *Audio Codec '97 Component Specification, Version 2.3* (also known as *AC '97 v2.3 Specification*) link for audio and telephony codecs (up to seven channels)
- Low Pin Count (LPC) interface
- Firmware Hub (FWH) interface support

The ICH5 incorporates a variety of PCI functions that are divided into four logical devices (B0:D30, B0:D31, B0:D29 and B1:D8). D30 is the hub interface-to-PCI bridge, D31 contains the PCI-to-LPC Bridge, IDE controller, SATA controller, SMBus controller and the AC '97 Audio and Modem controller functions and D29 contains the four USB UHCI controllers and one USB EHCI controller. B1:D8 is the integrated LAN controller.

**Table 2. PCI Devices and Functions**

| Bus:Device:Function | Function Description |
|---|---|
| Bus 0:Device 30:Function 0 | Hub Interface to PCI Bridge |
| Bus 0:Device 31:Function 0 | PCI to LPC Bridge[1] |
| Bus 0:Device 31:Function 1 | IDE Controller |
| Bus 0:Device 31:Function 2 | New: SATA Controller |
| Bus 0:Device 31:Function 3 | SMBus Controller |
| Bus 0:Device 31:Function 5 | AC'97 Audio Controller |
| Bus 0:Device 31:Function 6 | AC'97 Modem 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 | New: USB UHCI Controller #4 |
| Bus 0:Device 29:Function 7 | USB 2.0 EHCI 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, DMA.

The following sub-sections provide an overview of the ICH5 capabilities.

### Hub Architecture

The chipset's hub interface architecture ensures that the I/O subsystem; both PCI and the integrated I/O features (SATA, IDE, AC '97, USB, etc.), receive the bandwidth necessary for peak performance.

### PCI Interface

The ICH5 PCI interface provides a 33 MHz, Revision 2.3 compliant implementation. All PCI signals are 5-V tolerant, except PME#. The ICH5 integrates a PCI arbiter that supports up to six external PCI bus masters in addition to the internal ICH5 requests.

### IDE Interface (Bus Master Capability and Synchronous DMA Mode)

The fast IDE interface supports up to four IDE devices providing an interface for IDE hard disks and ATAPI devices. Each IDE device can have independent timings. The IDE interface supports PIO IDE transfers up to 16 Mbytes/sec and Ultra ATA transfers up 100 Mbytes/sec. It does not consume any ISA DMA resources. The IDE interface integrates 16x32-bit buffers for optimal transfers.

The ICH5's IDE system contains two independent IDE signal channels. They can be electrically isolated independently. They can be configured to the standard primary and secondary channels (four devices). There are integrated series resistors on the data and control lines (see Section 5.16 for details).

### SATA Controller

The SATA controller supports two SATA devices providing an interface for SATA hard disks and ATAPI devices. The SATA interface supports PIO IDE transfers up to 16 Mb/s and Serial ATA transfers up to 1.5 Gb/s (150 MB/s).

The ICH5's SATA system contains two independent SATA signal ports. They can be electrically isolated independently. Each SATA device can have independent timings. They can be configured to the standard primary and secondary channels.

### Low Pin Count (LPC) Interface

The ICH5 implements an LPC Interface as described in the *Low Pin Count Interface Specification, Revision 1.1*. The Low Pin Count (LPC) Bridge function of the ICH5 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.

### 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.

The ICH5 supports two types of DMA (LPC and PC/PCI). DMA via LPC is similar to ISA DMA. LPC DMA and PC/PCI DMA use the ICH5's DMA controller. The PC/PCI protocol allows PCI-based peripherals to initiate DMA cycles by encoding requests and grants via two PC/PCI REQ#/GNT# pairs.

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.

The ICH5 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, the ICH5 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 PIC described in the previous section, the ICH5 incorporates the Advanced Programmable Interrupt Controller (APIC).

### Universal Serial Bus (USB) Controller

The ICH5 contains an *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* -compliant 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. The ICH5 also contains four Universal Host Controller Interface (UHCI) controllers that support USB full-speed and low-speed signaling.

The ICH5 supports eight USB 2.0 ports. All eight ports are high-speed, full-speed, and low-speed capable. ICH5's port-routing logic determines whether a USB port is controlled by one of the UHCI controllers or by the EHCI controller. See Section 5.19 and Section 5.20 for details.

### LAN Controller

The ICH5'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 5.2 for details.

### RTC

The ICH5 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 separate 3 V lithium 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 ICH5 configuration.

### Enhanced Power Management

The ICH5's power management functions include enhanced clock control, local and global monitoring support for 14 individual devices, and various low-power (suspend) states (e.g., Suspend-to-DRAM and Suspend-to-Disk). A hardware-based thermal management circuit permits software-independent entrance to low-power states. The ICH5 contains full support for the *Advanced Configuration and Power Interface (ACPI) Specification, Revision 2.0b*.

### System Management Bus (SMBus 2.0)

The ICH5 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.

The ICH5's SMBus host controller provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). Also, the ICH5 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.

## Manageability

The ICH5 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.** The ICH5'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.** The ICH5 looks for the processor to fetch the first instruction after reset. If the processor does not fetch the first instruction, the ICH5 can reboot the system.
- **ECC Error Reporting.** When detecting an ECC error, the host controller has the ability to send one of several messages to the ICH5. The host controller can instruct the ICH5 to generate either an SMI#, NMI, SERR#, or TCO interrupt.
- **Function Disable.** The ICH5 provides the ability to disable the following functions: AC '97 Modem, AC '97 Audio, IDE, SATA, LAN, USB, UHCI, EHCI, 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.** The ICH5 provides an input signal (INTRUDER#) that can be attached to a switch that is activated by the system case being opened. The ICH5 can be programmed to generate an SMI# or TCO interrupt due to an active INTRUDER# signal.
- **SMBus 2.0.** The ICH5 integrates an SMBus controller that provides an interface to manage peripherals (e.g., serial presence detection (SPD) and thermal sensors) with host notify capabilities.

### AC ’97 2.3 Controller

The *AC ’97 v2.3 Specification* defines a digital interface that can be used to attach an *audio codec* (AC), a *modem codec* (MC), an *audio/modem codec* (AMC) or a combination of ACs and MC. The *AC ’97 v2.3 Specification* defines the interface between the system logic and the audio or modem codec, known as the *AC-link*.

By using an audio codec, the AC-link allows for cost-effective, high-quality, integrated audio on Intel’s chipset-based platform. In addition, an AC ’97 soft modem can be implemented with the use of a modem codec. Several system options exist when implementing AC ’97. The ICH5-integrated digital link allows several external codecs to be connected to the ICH5. The system designer can provide audio with an audio codec, a modem with a modem codec, or an integrated audio/modem codec. The digital link is expanded to support three audio codecs or two audio codecs and one modem codec.

The modem implementations for different countries must be taken into consideration, because telephone systems may vary. By using a split design, the audio codecs can be on-board and the modem codec can be placed on a riser.

The digital link in the ICH5 supports the *AC ’97 v2.3 Specification*, so it supports three codecs with independent PCI functions for audio and modem. Microphone input and left and right audio channels are supported for a high quality, two-speaker audio solution. Wake on Ring from Suspend also is supported with the appropriate modem codec.

The ICH5 expands the audio capability with support for up to six channels of PCM audio output (full AC3 decode). Six-channel audio consists of Front Left, Front Right, Back Left, Back Right, Center, and Subwoofer, for a complete surround-sound effect. ICH5 has expanded support for three audio codecs on the AC-link.

### Alert Standard Format (ASF) Controller

The *Alert Standard Format Specification, Version 1.03* supported by the ICH5, defines ASF alerting capabilities including system health information such as BIOS messages, POST alerts, OS failure notifications, and heartbeat signals to indicate the system is up and running on the network. Also included are environmental notifications such as thermal, voltage and fan alerts, which send proactive warnings that something is wrong with the hardware. In addition, asset security is provided by messages such as “cover tamper” and “CPU missing” that notify an IT manager of potential system break-ins and processor or memory theft.

This page is intentionally left blank.

# Signal Description 2

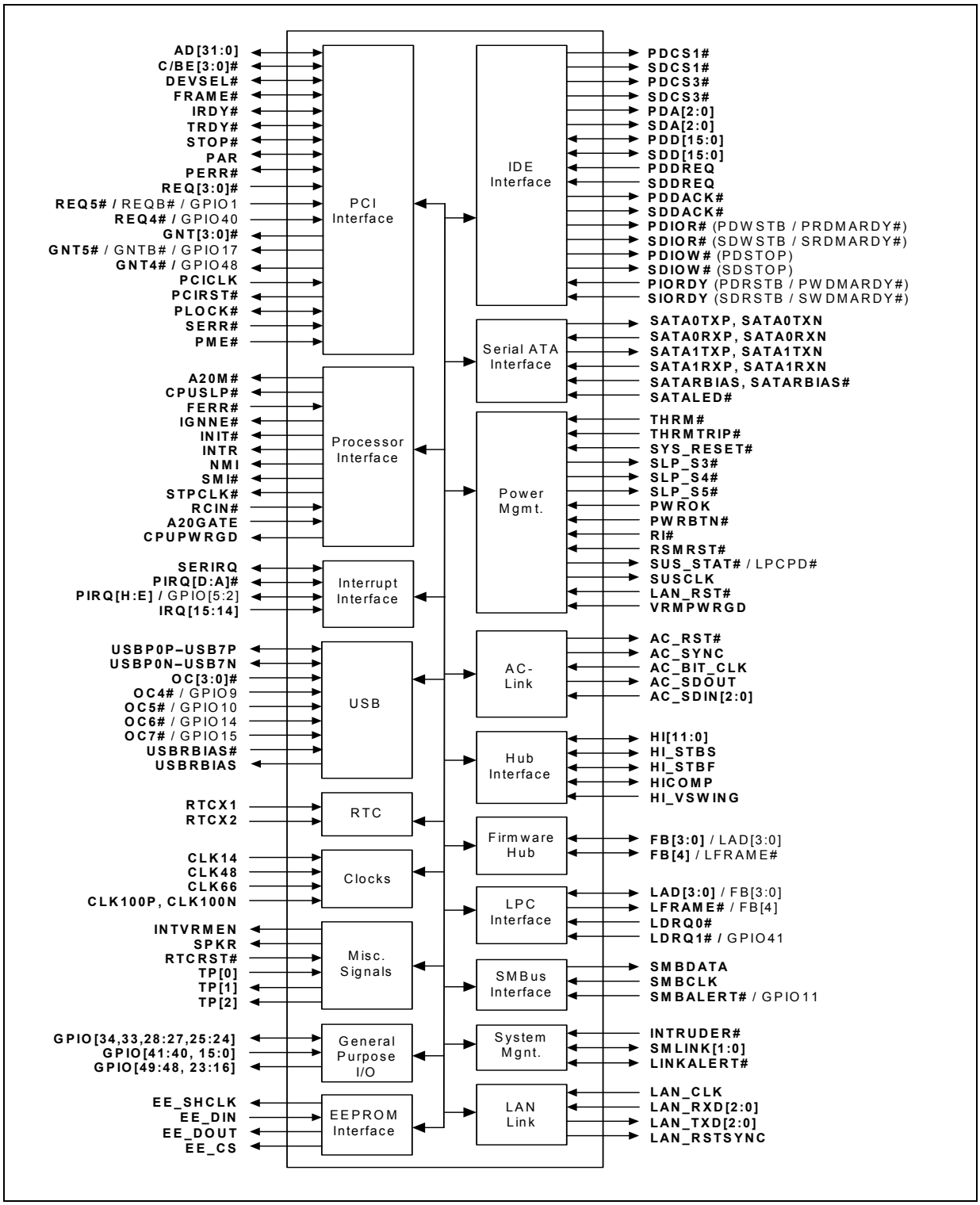
This chapter provides a detailed description of each signal. The signals are arranged in functional groups according to their associated interface.

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 following notations are used to describe the signal type:

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

## Figure 1. Intel® ICH5 Interface Signals Block Diagram



## 2.1 Hub Interface to Host Controller

### Table 3. Hub Interface Signals

| Name | Type | Description |
|---|---|---|
| **HI[11:0]** | I/O | **Hub Interface Signals** |
| **HI_STBF** | I/O | **Hub Interface Strobe First:** The first of two differential strobe signals used to transmit and receive data through the hub interface. |
| **HI_STBS** | I/O | **Hub Interface Strobe Second:** The second of two strobe signals used to transmit and receive data through the hub interface. |
| **HIRCOMP** | I/O | **Hub Interface Impedance Compensation:** This signal is used for hub interface buffer compensation. |
| **HI_VSWING** | I | **Hub Interface Voltage Swing:** This is an analog input used to control the voltage swing and impedance strength of hub interface pins. The expected voltage is 800 mV. |

## 2.2 Link to LAN Connect

### Table 4. LAN Connect Interface Signals

| Name | Type | Description |
|---|---|---|
| **LAN_CLK** | I | **LAN I/F Clock:** This signal is driven by the 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 LAN Connect component. |
| **LAN_RSTSYNC** | O | **LAN Reset/Sync:** The platform LAN Connect component's Reset and Sync signals are multiplexed onto this pin. |

## 2.3 EEPROM Interface

### Table 5. 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 the Intel® ICH5. This signal has an integrated pull-up resistor. |
| **EE_DOUT** | O | **EEPROM Data Out:** This signal transfers data from the ICH5 to the EEPROM. |
| **EE_CS** | O | **EEPROM Chip Select:** This signal is the chip select to the EEPROM. |

## 2.4 Flash BIOS Interface

### Table 6. Flash BIOS Interface Signals

| Name | Type | Description |
|---|---|---|
| **FB[3:0]** / LAD[3:0] | I/O | **Flash BIOS Signals:** These signals are multiplexed with the LPC address signals. |
| **FB4** / LFRAME# | I/O | **Flash BIOS Signals:** This signal is multiplexed with the LPC LFRAME# signal. |

## 2.5 PCI Interface

### Table 7. PCI Interface Signals (Sheet 1 of 3)

| Name | Type | Description |
|---|---|---|
| **AD[31:0]** | I/O | **PCI Address/Data:** AD[31:0] are the signals of the 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. The Intel® ICH5 will drive all 0s on AD[31:0] during the address phase of all PCI Special Cycles. |
| **C/BE[3:0]#** | I/O | **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]# define the bus command. During the data phase C/BE[3:0]# define the Byte Enables.<br>**C/BE[3:0]# — Command Type**<br>0 0 0 0 — Interrupt Acknowledge<br>0 0 0 1 — Special Cycle<br>0 0 1 0 — I/O Read<br>0 0 1 1 — I/O Write<br>0 1 1 0 — Memory Read<br>0 1 1 1 — Memory Write<br>1 0 1 0 — Configuration Read<br>1 0 1 1 — Configuration Write<br>1 1 0 0 — Memory Read Multiple<br>1 1 1 0 — Memory Read Line<br>1 1 1 1 — Memory Write and Invalidate<br>All command encodings not shown are reserved. The ICH5 does not decode reserved values, and therefore will not respond if a PCI master generates a cycle using 1 of the reserved values. |
| **DEVSEL#** | I/O | **Device Select:** The ICH5 asserts DEVSEL# to claim a PCI transaction. As an output, the ICH5 asserts DEVSEL# when a PCI master peripheral attempts an access to an internal ICH5 address or an address destined for the hub interface (main memory or AGP). As an input, DEVSEL# indicates the response to an ICH5-initiated transaction on the PCI bus. DEVSEL# is tri-stated from the leading edge of PCIRST#. DEVSEL# remains tri-stated by the ICH5 until driven by a Target device. |
| **FRAME#** | I/O | **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 the ICH5 when the ICH5 is the target, and FRAME# is an output from the ICH5 when the ICH5 is the Initiator. FRAME# remains tri-stated by the ICH5 until driven by an Initiator. |

## Table 7. PCI Interface Signals (Sheet 2 of 3)

| Name | Type | Description |
|---|---|---|
| **IRDY#** | I/O | **Initiator Ready:** IRDY# indicates the ICH5'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 the ICH5 has valid data present on AD[31:0]. During a read, it indicates the ICH5 is prepared to latch data. IRDY# is an input to the ICH5 when the ICH5 is the Target and an output from the ICH5 when the ICH5 is an Initiator. IRDY# remains tri-stated by the ICH5 until driven by an Initiator. |
| **TRDY#** | I/O | **Target Ready:** TRDY# indicates the ICH5'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, TRDY# indicates that the ICH5, as a Target, has placed valid data on AD[31:0]. During a write, TRDY# indicates the ICH5, as a Target is prepared to latch data. TRDY# is an input to the ICH5 when the ICH5 is the Initiator and an output from the ICH5 when the ICH5 is a Target. TRDY# is tri-stated from the leading edge of PCIRST#. TRDY# remains tri-stated by the ICH5 until driven by a target. |
| **STOP#** | I/O | **Stop:** STOP# indicates that the ICH5, as a Target, is requesting the Initiator to stop the current transaction. STOP# causes the ICH5, as an Initiator, to stop the current transaction. STOP# is an output when the ICH5 is a Target and an input when the ICH5 is an Initiator. STOP# is tri-stated from the leading edge of PCIRST#. STOP# remains tri-stated until driven by the ICH5. |
| **PAR** | I/O | **Calculated/Checked Parity:** PAR uses “even” parity calculated on 36 bits, AD[31:0] plus C/BE[3:0]#. “Even” parity means that the ICH5 counts the number of 1’ within the 36 bits plus PAR and the sum is always even. The ICH5 always calculates PAR on 36 bits regardless of the valid byte enables. The ICH5 generates PAR for address and data phases and only guarantees PAR to be valid one PCI clock after the corresponding address or data phase. The ICH5 drives and tri-states PAR identically to the AD[31:0] lines except that the ICH5 delays PAR by exactly one PCI clock. PAR is an output during the address phase (delayed one clock) for all ICH5 initiated transactions. PAR is an output during the data phase (delayed one clock) when the ICH5 is the Initiator of a PCI write transaction, and when it is the Target of a read transaction. ICH5 checks parity when it is the Target of a PCI write transaction. If a parity error is detected, the ICH5 will set the appropriate internal status bits, and has the option to generate an NMI# or SMI#. |
| **PERR#** | I/O | **Parity Error**: An external PCI device drives PERR# when it receives data that has a parity error. The ICH5 drives PERR# when it detects a parity error. The ICH5 can either generate an NMI# or SMI# upon detecting a parity error (either detected internally or reported via the PERR# signal when serving as the initiator). |
| **REQ[0:3]#**<br>**REQ4# /** GPIO40<br>**REQ5# /** REQB# / GPIO1 | I | **PCI Requests**: The ICH5 supports up to six masters on the PCI bus. The REQ4# pin can instead be used as a GPI. REQ5# is muxed with PC/PCI REQB# (must choose one or the other, but not both). If not used for PCI or PC/PCI, REQ5#/ REQB# can instead be used as GPIO1.<br><br>**NOTE:** R EQ0# is programmable to have improved arbitration latency for supporting PCI-based 1394 controllers. |
| **GNT[0:3]#**<br>**GNT4# /** GPIO48<br>**GNT5# /** GNTB# / GPIO17# | O | **PCI Grants:** The ICH5 supports up to 6 masters on the PCI bus. The GNT4# pin can instead be used as a GPO. GNT5# is multiplexed with PC/PCI GNTB# (must choose one or the other, but not both). If not needed for PCI or PC/PCI, GNT5# can instead be used as a GPIO.<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. GNTB#/GNT5#/GPIO17 has an internal pull-up. |
| **PCICLK** | I | **PCI Clock:** This is a 33 MHz clock. PCICLK provides timing for all transactions on the PCI Bus. |

### Table 7. PCI Interface Signals (Sheet 3 of 3)

| Name | Type | Description |
|---|---|---|
| **PCIRST#** | O | **PCI Reset:** ICH5 asserts PCIRST# to reset devices that reside on the PCI bus. The ICH5 asserts PCIRST# during power-up and when S/W initiates a hard reset sequence through the RC (CF9h) register. The ICH5 drives PCIRST# inactive a minimum of 1 ms after PWROK is driven active. The ICH5 drives PCIRST# active a minimum of 1 ms when initiated through the RC register. |
| **PLOCK#** | I/O | **PCI Lock:** This signal indicates an exclusive bus operation and may require multiple transactions to complete. ICH5 asserts PLOCK# when it performs non-exclusive transactions on the PCI bus. PLOCK# is ignored when PCI masters are granted the bus. |
| **SERR#** | I/OD | **System Error:** SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, the ICH5 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 the ICH5 may drive PME# active due to an internal wake event. The ICH5 will not drive PME# high, but it will be pulled up to VccSus3_3 by an internal pull-up resistor. |
| **REQA# /** GPIO0 **REQB# /** REQ5# / GPIO1 | I | **PC/PCI DMA Request [A:B]:** This request serializes ISA-like DMA requests for the purpose of running ISA-compatible DMA cycles over the PCI bus. This is used by devices such as PCI based Super I/O or audio codecs that need to perform legacy 8237 DMA but have no ISA bus.<br>When not used for PC/PCI requests, these signals can be used as general purpose Inputs. REQB# can instead be used as the 6th PCI bus request. |
| **GNTA# / GPIO16** **GNTB#** / GNT5# / GPIO17 | O | **PC/PCI DMA Acknowledges [A: B]:** This grant serializes an ISA-like DACK# for the purpose of running DMA/ISA Master cycles over the PCI bus. This is used by devices such as PCI based Super/IO or audio codecs which need to perform legacy 8237 DMA but have no ISA bus.<br>When not used for PC/PCI, these signals can be used as General Purpose Outputs. GNTB# can also be used as the 6th PCI bus master grant output. These signal have internal pull-up resistors. |

## 2.6 Serial ATA Interface

### Table 8. Serial ATA Interface Signals

| Name | Type | Description |
|---|---|---|
| **SATA0TXP** **SATA0TXN** | O | **Serial ATA 0 Differential Transmit Pair:** These are outbound high-speed differential signals to Port 0. |
| **SATA0RXP** **SATA0RXN** | I | **Serial ATA 0 Differential Receive Pair:** These are inbound high-speed differential signals from Port 0. |
| **SATA1TXP** **SATA1TXN** | O | **Serial ATA 1 Differential Transmit Pair:** These are outbound high-speed differential signals to Port 1. |
| **SATA1RXP** **SATA1RXN** | I | **Serial ATA 1 Differential Receive Pair:** These are inbound high-speed differential signals from Port 1. |
| **SATARBIAS** **SATARBIAS#** | I | **Serial ATA Resistor Bias:** These are analog connection points for an external resistor to ground. |
| **SATALED#** | OD | **SATA Drive Activity Indicator:** This signal indicates SATA drive activity when driven low. |

## 2.7 IDE Interface

**Table 9. IDE Interface Signals (Sheet 1 of 2)**

| Name | Type | Description |
|---|---|---|
| **PDCS1#, SDCS1#** | O | **Primary and Secondary IDE Device Chip Selects for 100 Range:** For ATA command register block. This output signal is connected to the corresponding signal on the primary or secondary IDE connector. |
| **PDCS3#, SDCS3#** | O | **Primary and Secondary IDE Device Chip Select for 300 Range:** For ATA control register block. This output signal is connected to the corresponding signal on the primary or secondary IDE connector. |
| **PDA[2:0], SDA[2:0]** | O | **Primary and Secondary IDE Device Address:** These output signals are connected to the corresponding signals on the primary or secondary IDE connectors. They are used to indicate which byte in either the ATA command block or control block is being addressed. |
| **PDD[15:0], SDD[15:0]** | I/O | **Primary and Secondary IDE Device Data:** These signals directly drive the corresponding signals on the primary or secondary IDE connector. There is a weak internal pull-down resistor on PDD7 and SDD7. |
| **PDDREQ, SDDREQ** | I | **Primary and Secondary IDE Device DMA Request:** These input signals are directly driven from the DRQ signals on the primary or secondary IDE connector. It is asserted by the IDE device to request a data transfer, and used in conjunction with the PCI bus master IDE function and are not associated with any AT compatible DMA channel. There is a weak internal pull-down resistor on these signals. |
| **PDDACK#, SDDACK#** | O | **Primary and Secondary IDE Device DMA Acknowledge:** These signals directly drive the DAK# signals on the primary and secondary IDE connectors. Each is asserted by the Intel® ICH5 to indicate to IDE DMA slave devices that a given data transfer cycle (assertion of DIOR# or DIOW#) is a DMA data transfer cycle. This signal is used in conjunction with the PCI bus master IDE function and are not associated with any AT-compatible DMA channel. |

### Table 9. IDE Interface Signals (Sheet 2 of 2)

| Name | Type | Description |
|---|---|---|
| **PDIOR# /** (PDWSTB / PRDMARDY#) **SDIOR# /** (SDWSTB / SRDMARDY#) | O | **Primary and Secondary Disk I/O Read (PIO and Non-Ultra DMA):** This is the command to the IDE device that it may drive data onto the PDD or SDD lines. Data is latched by the ICH5 on the deassertion edge of PDIOR# or SDIOR#. The IDE device is selected either by the ATA register file chip selects (PDCS1# or SDCS1#, PDCS3# or SDCS3#) and the PDA or SDA lines, or the IDE DMA acknowledge (PDDAK# or SDDAK#).<br>Primary and Secondary Disk Write Strobe (Ultra DMA Writes to Disk): This is the data write strobe for writes to disk. When writing to disk, ICH5 drives valid data on rising and falling edges of PDWSTB or SDWSTB.<br>Primary and Secondary Disk DMA Ready (Ultra DMA Reads from Disk): This is the DMA ready for reads from disk. When reading from disk, ICH5 deasserts PRDMARDY# or SRDMARDY# to pause burst data transfers. |
| **PDIOW# /** (PDSTOP) **SDIOW# /** (SDSTOP) | O | **Primary and Secondary Disk I/O Write (PIO and Non-Ultra DMA):** This is the command to the IDE device that it may latch data from the PDD or SDD lines. Data is latched by the IDE device on the deassertion edge of PDIOW# or SDIOW#. The IDE device is selected either by the ATA register file chip selects (PDCS1# or SDCS1#, PDCS3# or SDCS3#) and the PDA or SDA lines, or the IDE DMA acknowledge (PDDAK# or SDDAK#).<br>Primary and Secondary Disk Stop (Ultra DMA): ICH5 asserts this signal to terminate a burst. |
| **PIORDY /** (PDRSTB / PWDMARDY#) **SIORDY /** (SDRSTB / SWDMARDY#) | I | **Primary and Secondary I/O Channel Ready (PIO):** This signal will keep the strobe active (PDIOR# or SDIOR# on reads, PDIOW# or SDIOW# on writes) longer than the minimum width. It adds wait-states to PIO transfers.<br>Primary and Secondary Disk Read Strobe (Ultra DMA Reads from Disk): When reading from disk, ICH5 latches data on rising and falling edges of this signal from the disk.<br>Primary and Secondary Disk DMA Ready (Ultra DMA Writes to Disk): When writing to disk, this is de-asserted by the disk to pause burst data transfers. |

## 2.8 LPC Interface

### Table 10. LPC Interface Signals

| Name | Type | Description |
|---|---|---|
| **LAD[3:0]** / FB[3:0] | I/O | **LPC Multiplexed Command, Address, Data:** For LAD[3:0], internal pull-ups are provided. |
| **LFRAME#** / FB4 | O | **LPC Frame:** LFRAME# indicates the start of an LPC cycle, or an abort. |
| **LDRQ0#** **LDRQ1# /** GPIO41 | 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 GPI. |

# 2.9 Interrupt Interface

### Table 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 Section 5.8.6. 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 Section 5.8.6. 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. |
| **IRQ[14:15]** | I | **Interrupt Request 14–15:** These interrupt inputs are connected to the IDE drives. IRQ14 is used by the drives connected to the Primary controller and IRQ15 is used by the drives connected to the Secondary controller. |

## 2.10 USB Interface

### Table 12. USB Interface Signals

| Name | Type | Description |
|---|---|---|
| **USBP0P, USBP0N, USBP1P, 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><br>**NOTE:** No external resistors are required on these signals. The Intel® ICH5 integrates 15 kΩ pull-downs and provides an output driver impedance of 45 Ω which requires no external series resistor |
| **USBP2P, USBP2N, USBP3P, 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><br>**NOTE:** No external resistors are required on these signals. The ICH5 integrates 15 kΩ pull-downs and provides an output driver impedance of 45 Ω which requires no external series resistor |
| **USBP4P, USBP4N, USBP5P, 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><br>**NOTE:** No external resistors are required on these signals. The ICH5 integrates 15 kΩ pull-downs and provides an output driver impedance of 45 Ω which requires no external series resistor |
| **USBP6P, USBP6N, USBP7P, USBP7N** | I/O | **Universal Serial Bus Port 7:6 Differential:** These differential pairs are used to transmit Data/Address/Command signals for ports 6and 7. These ports can be routed to UHCI controller #4 or the EHCI controller.<br><br>**NOTE:** No external resistors are required on these signals. The ICH5 integrates 15 kΩ pull-downs and provides an output driver impedance of 45 Ω which requires no external series resistor |
| **OC[3:0]#**<br>**OC4# /** GPIO9<br>**OC5# /** GPIO10<br>**OC6# /** GPIO14<br>**OC7# /** GPIO15 | I | **Overcurrent Indicators:** These signals set corresponding bits in the USB controllers to indicate that an overcurrent condition has occurred.<br>OC[7:4]# may optionally be used as GPIs. |
| **USBRBIAS,**<br>**USBRBIAS#** | O | **USB Resistor Bias:** These are analog connection point for an external resistor to ground. |

# 2.11 Power Management Interface

### Table 13. Power Management Interface Signals

| Name | Type | Description |
|---|---|---|
| **THRM#** | I | **Thermal Alarm:** This is an active low signal generated by external hardware to start the Hardware clock throttling mode. Can also generate an SMI# or an SCI. |
| **THRMTRIP#** | I | **Thermal Trip**: When low, this signal indicates that a thermal trip from the processor occurred, and the Intel® ICH5 will immediately transition to a S5 state. The ICH5 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><br>**NOTE:** This pin must be used to control the DRAM power in order to use the ICH5's DRAM power-cycling feature. Refer to Section 5.13.11.2 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 ICH5 that core power and PCICLK have been stable for at least 99 ms. PWROK can be driven asynchronously. When PWROK is negated, the ICH5 asserts PCIRST#.<br><br>**NOTE:** PWROK must deassert for a minimum of three RTC clock periods in order for the ICH5 to fully reset the power and properly generate the PCIRST# 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. |
| **RI#** | I | **Ring Indicate:** This signal is an input from the modem interface. It can be enabled as a wake event, and this is preserved across power failures. |
| **SYS_RESET#** | I | **System Reset**: This pin forces an internal reset after being debounced. The ICH5 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:** This signal must be asserted at least 10 ms after the resume well power (VccSus3_3) is valid. When deasserted, this signal is an indication that the resume well power is stable. |
| **SUS_STAT#** / LPCPD# | O | **Suspend Status:** This signal is asserted by the ICH5 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 I/F. |
| **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. |

## 2.12 Processor Interface

**Table 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.<br>**Speed Strap:** During the reset sequence, the Intel® ICH5 drives A20M# high if the corresponding bit is set in the FREQ_STRP register. |
| **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 ICH5 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 ICH5 coprocessor error reporting function is enabled in the General Control Register (Device 31:Function 0, Offset D0, bit 13). If FERR# is asserted, the ICH5 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><br>**NOTE:** FERR# can be used in some states for notification by the processor of pending interrupt events. This functionality is independent of the General Control 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 ICH5 coprocessor error reporting function is enabled in the General Control Register (Device 31:Function 0, Offset D0, bit 13). If FERR# is active, indicating a coprocessor error, a write to the Coprocessor Error 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.<br>**Speed Strap:** During the reset sequence, ICH5 drives IGNNE# high if the corresponding bit is set in the FREQ_STRP register. |
| **INIT#** | O | **Initialization:** INIT# is asserted by the ICH5 for 16 PCI clocks to reset the processor. ICH5 can be configured to support processor BIST. In that case, INIT# will be active when PCIRST# is active. |
| **INTR** | O | **CPU Interrupt:** INTR is asserted by the ICH5 to signal the processor that an interrupt request is pending and needs to be serviced. It is an asynchronous output and normally driven low.<br>**Speed Strap:** During the reset sequence, ICH5 drives INTR high if the corresponding bit is set in the FREQ_STRP register. |
| **NMI** | O | **Non-Maskable Interrupt:** NMI is used to force a non-Maskable interrupt to the processor. The ICH5 can generate an NMI when either SERR# or IOCHK# is asserted. 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.<br>**Speed Strap:** During the reset sequence, ICH5 drives NMI high if the corresponding bit is set in the FREQ_STRP register. |
| **SMI#** | O | **System Management Interrupt:** SMI# is an active low output synchronous to PCICLK. It is asserted by the ICH5 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 ICH5 in response to one of many hardware or software events. When the processor samples STPCLK# asserted, it responds by stopping its internal clock. |

### Table 14. Processor Interface Signals (Sheet 2 of 2)

| Name | Type | Description |
|---|---|---|
| **RCIN#** | I | **Keyboard Controller Reset CPU:** The keyboard controller can generate INIT# to the processor. This saves the external OR gate with the ICH5's other sources of INIT#. When the ICH5 detects the assertion of this signal, INIT# is generated for 16 PCI clocks.<br><br>**NOTE:** The ICH5 will ignore RCIN# assertion during transitions to the 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 PCIsets. |
| **CPUPWRGD** / GPIO49 | OD | **CPU Power Good:** This signal should be connected to the processor's PWRGOOD input. This is an open-drain output signal (external pull-up resistor required) that represents a logical AND of the ICH5's PWROK and VRMPWRGD signals.<br>This signal may optionally be configured as a GPO. |

## 2.13 SMBus Interface

### Table 15. SM Bus Interface Signals

| Name | Type | Description |
|---|---|---|
| **SMBDATA** | I/OD | **SMBus Data:** External pull-up is required. |
| **SMBCLK** | I/OD | **SMBus Clock:** External pull-up 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 GPI. |

## 2.14 System Management Interface

### Table 16. System Management Interface Signals

| Name | Type | Description |
|---|---|---|
| **INTRUDER#** | I | **Intruder Detect:** This signal can be set to disable system if box detected open. This signal's status is readable, so it can be used like a GPI if the Intruder Detection is not needed. |
| **SMLINK[1:0]** | I/OD | **System Management Link:** 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 from the Intel® ICH5 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 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, then 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, then RTCX2 should be left floating. |

## 2.16 Other Clocks

### Table 18. Other Clocks

| Name | Type | Description |
|---|---|---|
| **CLK14** | I | **Oscillator Clock:** Used for 8254 timers. Runs at 14.31818 MHz. This clock is permitted to stop during S3 (or lower) states. |
| **CLK48** | I | **48 MHz Clock:** Used to run the USB controller. Runs at 48 MHz. This clock is permitted to stop during S3 (or lower) states. |
| **CLK66** | I | **66 MHz Clock:** Used to run the hub interface. Runs at 66 MHz. This clock is permitted to stop during S3 (or lower) states. |
| **CLK100P**<br>**CLK100N** | I | **100 MHz Differential Clock:** These signals are used to run the SATA controller. The clock runs at 100 MHz. This clock is permitted to stop during S3 (or lower) states. |

## 2.17 Miscellaneous Signals

### Table 19. Miscellaneous Signals

| Name | Type | Description |
|---|---|---|
| **INTVRMEN** | I | **Internal Voltage Regulator Enable:** This signal enables the internal 1.5 V Suspend regulator. It connects to VccRTC. |
| **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 PCIRST#, its output state is 0.<br>**NOTE:** SPKR is sampled at the rising edge of PWROK as a functional strap. See Section 2.21.1 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>**NOTES:**<br>1. Unless entering the XOR Chain Test Mode, the RTCRST# input must always be high when all other RTC power planes are on<br>2. In the case where the RTC battery is not functional or missing on the platform, the RTCRST# pin must rise before the RSMRST# pin. |
| **TP0** | I | **Test Point 0:** This signal must have an external pull-up to VccSus3_3. |
| **TP1** | O | **Test Point 1:** This signal is not implemented and should be routed to a test point. |
| **TP2** | O | **Test Point 2:** This signal is not implemented and should be routed to a test point. |

## 2.18 AC-Link

### Table 20. AC-Link Signals

| Name | Type | Description |
|---|---|---|
| **AC_RST#** | O | **AC '97 Reset:** Master hardware reset to external codec(s). |
| **AC_SYNC** | O | **AC '97 Sync:** 48 kHz fixed rate sample sync to the codec(s). |
| **AC_BIT_CLK** | I | **AC '97 Bit Clock:** 12.288 MHz serial data clock generated by the external Codec(s). This signal has an integrated pull-down resistor (see Note below). |
| **AC_SDOUT** | O | **AC '97 Serial Data Out:** Serial TDM data output to the codec(s).<br><br>**NOTE:** AC_SDOUT is sampled at the rising edge of PWROK as a functional strap. See Section 2.21.1 for more details. |
| **AC_SDIN[2:0]** | I | **AC '97 Serial Data In 2:0**: Serial TDM data inputs from the three codecs. |

**NOTE:** An integrated pull-down resistor on AC_BIT_CLK is enabled when either:
- The ACLINK Shutoff bit in the AC'97 Global Control Register (See Section 15.2.8) is set to 1, or
- Both Function 5 and Function 6 of Device 31 are disabled.

Otherwise, the integrated pull-down resistor is disabled.

## 2.19 General Purpose I/O

### Table 21. General Purpose I/O Signals (Sheet 1 of 2)

| Name | Type | Description |
|---|---|---|
| **GPIO49** | OD | Fixed as Output only. Processor I/F power well. Can instead be used as CPUPWRGD. |
| **GPIO48** | O | Fixed as Output only. Main power well. Can instead be used as GNT4#. |
| **GPIO[47:42]** | N/A | Not implemented. |
| **GPIO41** | I | Fixed as Input only. Main power well. Can be used instead as LDRQ1#. |
| **GPIO40** | I | Fixed as Input only. Main power well. Can be used instead as REQ4#. |
| **GPIO[39:35]** | N/A | Not implemented. |
| **GPIO34** | I/O | Can be input or output. Main power well. Not multiplexed. |
| **GPIO33** | N/A | Not implemented. |
| **GPIO32** | I/O | Can be input or output. Main power well. Not multiplexed. |
| **GPIO[31:29]** | N/A | Not implemented. |
| **GPIO[28:27]** | I/O | Can be input or output. Resume power well. Not multiplexed. |
| **GPIO26** | I/O | Not implemented. |
| **GPIO25** | I/O | Can be input or output. Resume power well. Not multiplexed. |
| **GPIO24** | I/O | Can be input or output. Resume power well. |
| **GPIO23** | O | Fixed as output only. Main power well. |
| **GPIO22** | OD | Fixed as output only. Main power well. |
| **GPIO21** | O | Fixed as output only. Main power well. |

### Table 21. General Purpose I/O Signals (Sheet 2 of 2)

| Name | Type | Description |
|---|---|---|
| **GPIO20** | O | Fixed as output only. Main power well. |
| **GPIO19** | O | Fixed as output only. Main power well. |
| **GPIO18** | O | Fixed as output only. Main power well. |
| **GPIO[17:16]** | O | Fixed as Output only. Main power well. Can be used instead as PC/PCI GNT[A:B]#. GPIO17 can also alternatively be used for PCI GNT5#. Integrated pull-up resistor. |
| **GPIO[15:14]** | I | Fixed as Input only. Resume power well. Can be used instead as OC[7:6]#. |
| **GPIO[13:12]** | I | Fixed as Input only. Resume power well. Not multiplexed. |
| **GPIO11** | I | Fixed as Input only. Resume power well. Can be used instead as SMBALERT#. |
| **GPIO[10:9]** | I | Fixed as Input only. Resume power well. Can be used instead as OC[5:4]#. |
| **GPIO8** | I | Fixed as Input only. Resume power well. Not multiplexed. This GPIO is recommended for use as the Communications Streaming Architecture (CSA) PME# signal to provide Wake-On-LAN capabilities. The GPI_INV bit corresponding to GPIO8 must be set in order to achieve the correct polarity in the General Purpose Event 0 Status Register. |
| **GPIO7** | I | Fixed as Input only. Main power well. Not multiplexed. |
| **GPIO6** | I | Fixed as Input only. Main power well. |
| **GPIO[5:2]** | I | Fixed as Input only. Main power well. Can be used instead as PIRQ[E:H]#. |
| **GPIO[1:0]** | I | Fixed as Input only. Main power well. Can be used instead as PC/PCI REQ[A:B]#. GPIO1 can also alternatively be used for PCI REQ5#. |

**NOTE:** GPIO[0:7] are 5 V tolerant. GPIO[8:49] **not** 5 V tolerant.

## 2.20 Power and Ground

### Table 22. Power and Ground Signals

| Name | Description |
|---|---|
| **Vcc3_3** | 3.3 V supply for core well I/O buffers (18 pins). This power may be shut off in S3, S4, S5 or G3 states. |
| **Vcc1_5** | 1.5 V supply for core well logic and hub interface logic (25 pins). This power may be shut off in S3, S4, S5, or G3 states. |
| **V5REF** | Reference for 5 V tolerance on core well inputs (2 pins). This power may be shut off in S3, S4, S5, or G3 states. |
| **HIREF** | 350 mV analog input for hub interface (1 pin).<br>This power is shut off in S3, S4, S5, and G3 states. |
| **VccSus3_3** | 3.3 V supply for resume well I/O buffers (10 pins). This power is not expected to be shut off unless the system is unplugged. |
| **VccSus1_5_A** | 1.5 V supply for resume well logic (1 pin). This power is not expected to be shut off unless AC power is not available.<br>**NOTE:**<br>1. This voltage plane is generated internally<br>2. Do not connect the three sets of VccSus1_5_x signal groups on the Intel® ICH5 together. Each group needs to be independently connected to its corresponding decoupling capacitor for optimum noise isolation. |
| **VccSus1_5_B** | 1.5 V supply for resume well logic (3 pins). This power is not expected to be shut off unless AC power is not available.<br>**NOTE:**<br>1. This voltage plane is generated internally<br>2. Do not connect the three sets of VccSus1_5_x signal groups on the ICH5 together. Each group needs to be independently connected to its corresponding decoupling capacitor for optimum noise isolation. |
| **VccSus1_5_C** | 1.5 V supply for resume well logic (2 pins). This power is not expected to be shut off unless AC power is not available.<br>**NOTE:**<br>1. This voltage plane is generated internally<br>2. Do not connect the three sets of VccSus1_5_x signal groups on the ICH5 together. Each group needs to be independently connected to its corresponding decoupling capacitor for optimum noise isolation. |
| **V5REF_Sus** | 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. |
| **VccRTC** | 3.3 V (can drop to 1.0 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>**NOTE:** Implementations should not attempt to clear CMOS by using a jumper to pull VccRTC low. Clearing CMOS in an ICH5-based platform can be done by using a jumper on RTCRST# or GPI, or using SAFEMODE strap. |
| **VccUSBPLL** | 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. |
| **VccSATAPLL** | 1.5 V supply for core well logic (2 pins). This signal is used for the SATA PLL. This power may be shut off in S3, S4, S5, or G3 states. |
| **V_CPU_IO** | Powered by the same supply as the processor I/O voltage (3 pins). This supply is used to drive the processor interface signals listed in Table 14. |
| **Vss** | Grounds (119 pins). |

# 2.21 Pin Straps

## 2.21.1 Functional Straps

The following signals are used for static configuration. They are sampled at the rising edge of PWROK to select configurations, 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.

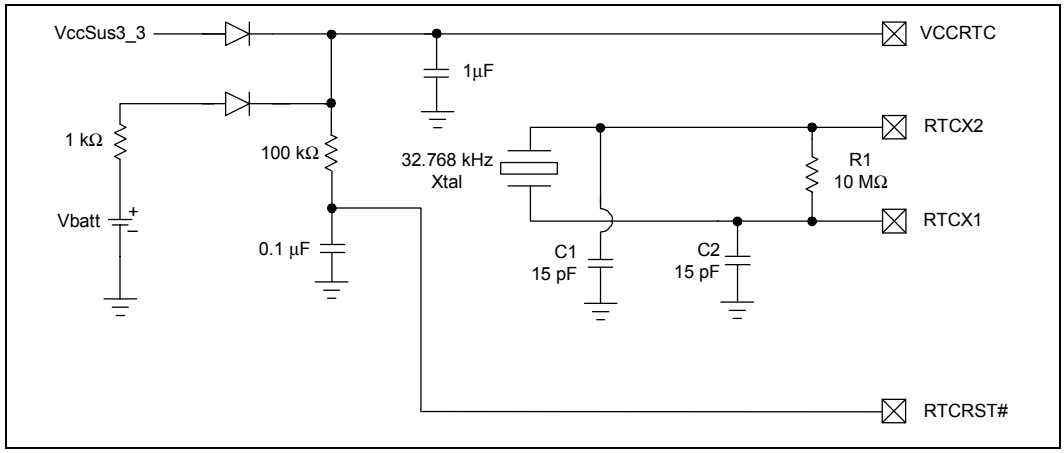
**Table 23. Functional Strap Definitions**

| Signal | Usage | When Sampled | Comment |
|---|---|---|---|
| AC_SDOUT | Safe Mode | Rising Edge of PWROK | The signal has a weak internal pull-down. If the signal is sampled high, the Intel® ICH5 will set the processor speed strap pins for safe mode. Refer to the processor specification for speed strapping definition. The status of this strap is readable via the SAFE_MODE bit (bit 2, D31: F0, Offset D4h). |
| GNTA# | 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 (ICH5 inverts A16 for all cycles targeting FWH BIOS space). The status of this strap is readable via the Top_Swap bit (bit 13, D31: F0, Offset D4h). Note that software will not be able to clear the Top-Swap bit until the system is rebooted without GNTA# being pulled down. |
| GNT5# / GNTB / GPIO17 | Reserved | | This signal has a weak internal pull-up.<br><br>**NOTE:** This signal should not be pulled low. |
| TP1 | Reserved | | This signal has a weak internal pull-down.<br><br>**NOTE:** This signal should not be pulled high. |
| AC_SYNC | Reserved | | This signal has a weak internal pull-down. |
| LINKALERT# | Reserved | | This signal requires an external pullup resistor. |
| 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 (ICH5 will disable the TCO timer system reboot feature). The status of this strap is readable via the NO_REBOOT bit (bit 1, D31: F0, Offset D4h). |

## 2.21.2 External RTC Circuitry

To reduce RTC well power consumption, the ICH5 implements an internal oscillator circuit that is sensitive to step voltage changes in VccRTC. Figure 2 shows a schematic diagram of the circuitry required to condition these voltages to ensure correct operation of the ICH5 RTC.

**Figure 2. Example External RTC Circuit**




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

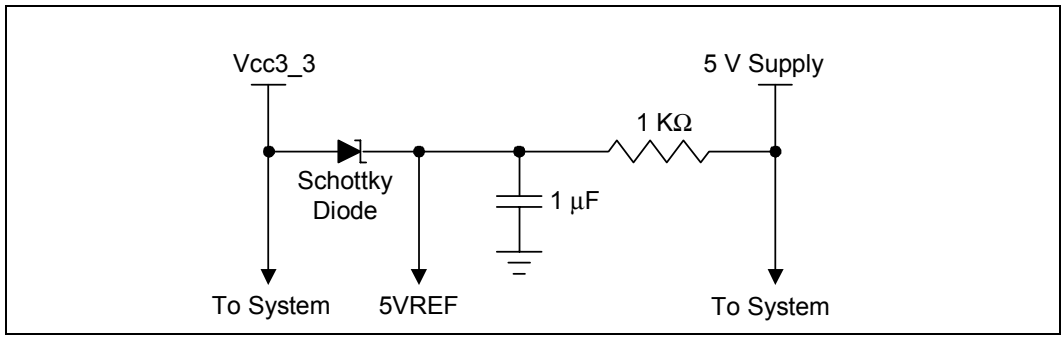
## 2.21.3 Power Sequencing Requirements

### 2.21.3.1 V5REF / Vcc3_3 Sequencing Requirements

V5REF is the reference voltage for 5 V tolerance on inputs to the ICH5. V5REF 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. The rule must be followed in order to ensure the safety of the ICH5. If the rule is violated, internal diodes will attempt to draw power sufficient to damage the diodes from the Vcc3_3 rail. Figure 3 shows a sample implementation of how to satisfy the V5REF/3.3 V sequencing rule.

This rule also applies to the standby rails, but in most platforms, the VccSus3_3 rail is derived from the VccSus5 rail; therefore, the VccSus3_3 rail will always come up after the VccSus5 rail. As a result, V5REF_Sus will always be powered up before VccSus3_3. In platforms that do not derive the VccSus3_3 rail from the VccSus5 rail, this rule must be comprehended in the platform design.

**Figure 3. Example V5REF Sequencing Circuit**



### 2.21.3.2 3.3 V/1.5 V Standby Power Sequencing Requirements

There is an integrated 1.5 V standby regulator that is used to power the resume well of the ICH5. Due to the use of this internal regulator, there are **no** power sequencing requirements for associated 3.3 V/1.5 V (standby or core) rails of the ICH5.

## 2.21.4 Test Signals

### 2.21.4.1 Test Mode Selection

When PWROK is active (high) for at least 76 PCI clocks, driving RTCRST# active (low) for a number of PCI clocks (33 MHz) will activate a particular test mode a specified in Table 24.

***Note:*** RTCRST# may be driven low any time after PCIRST is inactive. Refer to Chapter 21 for a detailed description of the ICH5 test modes.

**Table 24. Test Mode Selection**

| Number of PCI Clocks RTCRST# driven low after PWROK active | Test Mode |
|---|---|
| <4 | No Test Mode Selected |
| 4 | XOR Chain 1 |
| 5 | XOR Chain 2 |
| 6 | XOR Chain 3 |
| 7 | XOR Chain 4 |
| 8 | All “Z” |
| 9–13 | Reserved. DO NOT ATTEMPT |
| 14 | Long XOR |
| 15–42 | Reserved. DO NOT ATTEMPT |
| 43–51 | No Test Mode Selected |
| 52 | XOR Chain 6 |
| 53 | XOR Chain 4 Bandgap |
| >53 | No Test Mode Selected |

# Intel® ICH5 Power Planes and Pin States 3

## 3.1 Power Planes

**Table 25. Intel® ICH5 Power Planes**

| Plane | Description |
|---|---|
| Main I/O (3.3 V) | **Vcc3_3:** Powered by the main power supply. When the system is in the S3, S4, S5, or G3 state, this plane is assumed to be shut off. |
| Main Logic (1.5 V) | **Vcc1_5:** Powered by the main power supply. When the system is in the S3, S4, S5, or G3 state, this plane is assumed to be shut off. |
| Resume I/O (3.3 V Standby) | **VccSus3_3:** Powered by the main power supply in S0–S1 states. Powered by the trickle power supply when the system is in the S3, S4, or S5 state. Assumed to be shut off only when in the G3 state (system is unplugged). |
| Resume Logic (1.5 V Standby) | **VccSus1_5:** Powered by the main power supply in S0–S1 states. Powered by the trickle power supply when the system is in the S3, S4, or S5 state. Assumed to be shut off only when in the G3 state (system is unplugged).<br>These planes are generated from the integrated VRM. |
| Processor I/F (0.8 ~ 1.75 V) | **V_CPU_IO:** Powered by the main power supply via processor voltage regulator. When the system is in the S3, S4, S5, or G3 state, this plane is assumed to be shut off. |
| RTC | **VccRTC:** When other power is available (from the main supply), external diode coupling will provide power to reduce the drain on the RTC battery. The batter is assumed to operate from 3.3 V down to 1.0 V. |

## 3.2 Integrated Pull-Ups and Pull-Downs

### Table 26. Integrated Pull-Up and Pull-Down Resistors

| Signal | Resistor Type | Nominal Value | Notes |
|---|---|---|---|
| AC_BITCLK | pull-down | 20 kΩ | 1, 9 |
| AC_RST# | pull-down | 20 kΩ | 2, 9 |
| AC_SDIN[2:0] | pull-down | 20 kΩ | 2 |
| AC_SDOUT | pull-down | 20 kΩ | 2, 8, 9 |
| AC_SYNC | pull-down | 20 kΩ | 2, 8, 9 |
| EE_DIN | pull-up | 20 kΩ | 3 |
| EE_DOUT | pull-up | 20 kΩ | 3 |
| EE_CS | pull-up | 20 kΩ | 3 |
| GNT[B:A]# / GNT5# / GPIO[17:16] | pull-up | 20 kΩ | 3, 8 |
| LAD[3:0]# / FB[3:0]# | pull-up | 20 kΩ | 3 |
| LDRQ[1:0] / GPIO41 | pull-up | 20 kΩ | 3 |
| LAN_RXD[2:0] | pull-up | 10 kΩ | 4 |
| LAN_CLK | pull-down | 100 kΩ | 5 |
| PME# | pull-up | 20 kΩ | 3 |
| PWRBTN# | pull-up | 20 kΩ | 3 |
| PDD7 / SDD7 | pull-down | 11.5 kΩ | 6 |
| PDDREQ / SDDREQ | pull-down | 11.5 kΩ | 6 |
| SPKR | pull-down | 20 kΩ | 2, 8 |
| TP1 | pull-down | 20 kΩ | 2, 8 |
| USB[7:0] [P,N] | pull-down | 15 kΩ | 7 |

**NOTES:**
1. Simulation data shows that these resistor values can range from 10 kΩ to 40 kΩ.
2. Simulation data shows that these resistor values can range from 9 kΩ to 50 kΩ.
3. Simulation data shows that these resistor values can range from 15 kΩ to 35 kΩ.
4. Simulation data shows that these resistor values can range from 7.5 kΩ to 16 kΩ.
5. Simulation data shows that these resistor values can range from 45 kΩ to 170 kΩ.
6. Simulation data shows that these resistor values can range from 5.7 kΩ to 28.3 kΩ.
7. Simulation data shows that these resistor values can range from 14.25 kΩ to 24.8 kΩ.
8. The pull-up or pull-down on this signal is only enabled at boot/reset for strapping function.
9. The pull-down on this signal is enabled when the ACLINK Shutoff bit in the AC ‘97 Global Control Register is set to 1.

## 3.3 IDE Integrated Series Termination Resistors

Table 27 shows the ICH5 IDE signals that have integrated series termination resistors.

**Table 27. IDE Series Termination Resistors**

| Signal | Integrated Series Termination Resistor Value |
|---|---|
| PDD[15:0], SDD[15:0], PDIOW#, SDIOW#, PDIOR#, SDIOR#, PDREQ, SDREQ, PDDACK#, SDDACK#, PIORDY, SIORDY, PDA[2:0], SDA[2:0], PDCS1#, SDCS1#, PDCS3#, SDCS3#, IRQ14, IRQ15 | approximately 33 Ω (See Note) |

**NOTE:** Simulation data indicates that the integrated series termination resistors are a nominal 33 Ω but can range from 21 Ω to 75 Ω.

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

Table 28 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. ICH5 not driving the signal high or low. |
| “High” | ICH5 is driving the signal to a logic 1 |
| “Low” | ICH5 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” | ICH5 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 ICH5 is not driving |

Note that the signal levels are the same in S4 and S5.

## Table 28. Power Plane and States for Output and I/O Signal (Sheet 1 of 3)

| Signal Name | Power Plane | During PCIRST#[4] / RSMRST#[5] | Immediately after PCIRST#[4] / RSMRST#[5] | S1 | S3 | S4/S5 |
|---|---|---|---|---|---|---|
| **PCI Bus** | | | | | | |
| AD[31:0] | Main I/O | High-Z | Undefined | Defined | Off | Off |
| C/BE[3:0]# | Main I/O | High-Z | Undefined | Defined | Off | Off |
| DEVSEL# | Main I/O | High-Z | High-Z | High-Z | Off | Off |
| FRAME# | Main I/O | High-Z | High-Z | High-Z | Off | Off |
| GNT[4:0]# | Main I/O | High | High | High | Off | Off |
| GNT[A:B]# | Main I/O | High-Z with Internal Pull-Up | High | High | Off | Off |
| IRDY#, TRDY# | Main I/O | High-Z | High-Z | High-Z | Off | Off |
| PAR | Main I/O | High-Z | Undefined | Defined | Off | Off |
| PCIRST# | Resume I/O | Low | High | High | Low | Low |
| PERR# | Main I/O | High-Z | High-Z | High-Z | Off | Off |
| PLOCK# | Main I/O | High-Z | High-Z | High-Z | Off | Off |
| STOP# | Main I/O | High-Z | High-Z | High-Z | Off | Off |
| **LPC Interface** | | | | | | |
| LAD[3:0] | Main I/O | High | High | High | Off | Off |
| LFRAME# | Main I/O | High | High | High | Off | Off |
| **LAN Connect and EEPROM Interface** | | | | | | |
| EE_CS | Resume I/O | High | Running | Defined | Defined | Defined |
| EE_DOUT | Resume I/O | High | High | Defined | Defined | Defined |
| EE_SHCLK | Resume I/O | Low | Running | Defined | Defined | Defined |
| LAN_RSTSYNC | Resume I/O | High | Low | Defined | Defined | Defined |
| LAN_TXD[2:0] | Resume I/O | Low | Low | Defined | Defined | Defined |
| **IDE Interface** | | | | | | |
| PDA[2:0], SDA[2:0] | Main I/O | Undefined | Undefined | Undefined | Off | Off |
| PDCS1#, PDCS3# | Main I/O | High | High | High | Off | Off |
| PDD[15:8], SDD[15:8], PDD[6:0], SDD[6:0] | Main I/O | High-Z | High-Z | High-Z | Off | Off |
| PDD7, SDD7 | Main I/O | Low | Low | Low | Off | Off |
| PDDACK#, SDDACK# | Main I/O | High | High | High | Off | Off |
| PDIOR#, PDIOW# | Main I/O | High | High | High | Off | Off |
| SDCS1#, SDCS3# | Main I/O | High | High | High | Off | Off |
| SDIOR#, SDIOW# | Main I/O | High | High | High | Off | Off |

## Table 28. Power Plane and States for Output and I/O Signal (Sheet 2 of 3)

| Signal Name | Power Plane | During PCIRST#[4] / RSMRST#[5] | Immediately after PCIRST#[4] / RSMRST#[5] | S1 | S3 | S4/S5 |
|---|---|---|---|---|---|---|
| **SATA Interface** | | | | | | |
| SATA0TXP, SATA0TXN SATA1TXP, SATA1TXN | Main I/O | High-Z | High-Z | Defined | Off | Off |
| SATARBIAS | Main I/O | High-Z | High-Z | Defined | Defined | Defined |
| SATALED# | Main I/O | Low | High-Z | Defined | Off | Off |
| **Interrupts** | | | | | | |
| PIRQ[A:H]# | Main I/O | High-Z | High-Z | High-Z | Off | Off |
| SERIRQ | Main I/O | High-Z | High-Z | High-Z | Off | Off |
| **USB Interface** | | | | | | |
| USBP[7:0][P,N] | Resume I/O | Low | Low | Low | Low | Low |
| USBRBIAS | Resume I/O | High-Z | High-Z | Defined | Defined | Defined |
| **Power Management** | | | | | | |
| SLP_S3# | Resume I/O | Low | High | High | Low | Low |
| SLP_S4# | Resume I/O | Low | High | High | High | Low |
| SLP_S5# | Resume I/O | Low | High | High | High | Low |
| SUS_STAT# | Resume I/O | Low | High | High | Low | Low |
| SUSCLK | Resume I/O | Running | | | | |
| **Processor Interface** | | | | | | |
| A20M# | CPU I/O | See Note 1 | High | High | Off | Off |
| CPUPWRGD | CPU I/O | See Note 3 | High-Z | High-Z | Off | Off |
| CPUSLP# | CPU I/O | High | High | Defined | Off | Off |
| IGNNE# | CPU I/O | See Note 1 | High | High | Off | Off |
| INIT# | CPU I/O | High | High | High | Off | Off |
| INTR | CPU I/O | See Note 1 | Low | Low | Off | Off |
| NMI | CPU I/O | See Note 1 | Low | Low | Off | Off |
| SMI# | CPU I/O | High | High | High | Off | Off |
| STPCLK# | CPU I/O | High | High | Low | Off | Off |

## Table 28. Power Plane and States for Output and I/O Signal (Sheet 3 of 3)

| Signal Name | Power Plane | During PCIRST#[4] / RSMRST#[5] | Immediately after PCIRST#[4] / RSMRST#[5] | S1 | S3 | S4/S5 |
|---|---|---|---|---|---|---|
| **SMBus Interface** | | | | | | |
| SMBCLK, SMBDATA | Resume I/O | High-Z | High-Z | Defined | Defined | Defined |
| **System Management Interface** | | | | | | |
| SMLINK[1:0] | Resume I/O | High-Z | High-Z | Defined | Defined | Defined |
| LINKALERT# | Resume I/O | High-Z | High-Z | Defined | Defined | Defined |
| **Miscellaneous Signals** | | | | | | |
| SPKR | Main I/O | Low with Internal Pull-Down | Low | Defined | Off | Off |
| **AC '97 Interface** | | | | | | |
| AC_RST# | Resume I/O | Low | Low | Cold Reset Bit (High) | Low | Low |
| AC_SDOUT | Main I/O | Low | Running | Low | Off | Off |
| AC_SYNC | Main I/O | Low | Running | Low | Off | Off |
| **Multiplexed GPIO Signals** | | | | | | |
| GPIO[17:16] | Main I/O | High-Z with Internal Pull-Up | High | High | Off | Off |
| GPIO48 | Main I/O | High | High | High | Off | Off |
| GPIO49 | CPU I/O | See Note 6 | High-Z | High-Z | Off | Off |
| **Unmultiplexed GPIO Signals** | | | | | | |
| GPIO18 | Main I/O | High | See Note 2 | Defined | Off | Off |
| GPIO[20:19] | Main I/O | High | High | Defined | Off | Off |
| GPIO21 | Main I/O | High | High | Defined | Off | Off |
| GPIO22 | Main I/O | High-Z | High-Z | Defined | Off | Off |
| GPIO23 | Main I/O | Low | Low | Defined | Off | Off |
| GPIO24 | Resume I/O | High | High | Defined | Defined | Defined |
| GPIO25 | Resume I/O | High | High | Defined | Defined | Defined |
| GPIO[28:27] | Resume I/O | High | High | Defined | Defined | Defined |
| GPIO32 | Main I/O | High | High | Defined | Off | Off |
| GPIO34 | Main I/O | High | High | Defined | Off | Off |

**NOTES:**
1. ICH5 sets these signals at reset for processor frequency strap.
2. GPIO18 will toggle at a frequency of approximately 1 Hz when the ICH5 comes out of reset
3. CPUPWRGD is an open-drain output that represents a logical AND of the ICH5's VRMPWRGD and PWROK signals, and thus will be driven low by ICH5 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-Z.
4. The states of main I/O signals are taken at the times During PCIRST# and Immediately after PCIRST#.
5. The states of resume I/O signals are taken at the times During RSMRST# and Immediately after RSMRST#.
6. GPIO48 is an open-drain output. During boot, or during a hard reset with power cycling, GPIO48 will be expected to transition from low to High-Z.

## 3.5 Power Planes for Input Signals

Table 29 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

**Table 29. Power Plane for Input Signals (Sheet 1 of 2)**

| Signal Name | Power Well | Driver During Reset | S1 | S3 | S5 |
|---|---|---|---|---|---|
| A20GATE | Main I/O | External Microcontroller | Static | Low | Low |
| AC_BIT_CLK | Main I/O | AC '97 Codec | Low | Low | Low |
| AC_SDIN[2:0] | Resume I/O | AC '97 Codec | Low | Low | Low |
| CLK14 | Main I/O | Clock Generator | Running | Low | Low |
| CLK48 | Main I/O | Clock Generator | Running | Low | Low |
| CLK66 | Main Logic | Clock Generator | Running | Low | Low |
| CLK100P<br>CLK100N | Main Logic | Clock Generator | Running | Low | Low |
| EE_DIN | Resume I/O | EEPROM Component | Driven | Driven | Driven |
| FERR# | CPU I/O | Processor | Static | Low | Low |
| GPIO[1:0] | Main I/O | External Circuit | Driven | Low | Low |
| GPIO[5:2] | Main I/O | External Circuit | Driven | Driven | Driven |
| GPIO[10:9] | Resume I/O | External Pull-Ups | Driven | Driven | Driven |
| GPIO11 | Resume I/O | External Pull-Up | Driven | Driven | Driven |
| GPIO[15:14] | Resume I/O | External Pull-Ups | Driven | Driven | Driven |
| GPIO40 | Main I/O | External Circuit | Driven | Low | Low |
| GPIO41 | Main I/O | External Circuit | High | Low | Low |
| INTRUDER# | RTC | External Switch | Driven | Driven | Driven |
| INTVRMEN | RTC | External Pull-up | Driven | Driven | Driven |
| IRQ[15:14] | Main I/O | IDE Device | Static | Low | Low |
| LAN_CLK | Resume I/O | LAN Connect Component | Driven | Driven | Driven |
| LAN_RST# | Resume I/O | External RC Circuit | High | High | High |
| LAN_RXD[2:0] | Resume I/O | LAN Connect Component | Driven | Driven | Driven |
| LDRQ0# | Main I/O | LPC Devices | High | Low | Low |
| LDRQ1# | Main I/O | LPC Devices | High | Low | Low |
| OC[7:0]# | Resume I/O | External Pull-Ups | Driven | Driven | Driven |
| PCICLK | Main I/O | Clock Generator | Running | Low | Low |
| PDDREQ | Main I/O | IDE Device | Static | Low | Low |

### Table 29. Power Plane for Input Signals (Sheet 2 of 2)

| Signal Name | Power Well | Driver During Reset | S1 | S3 | S5 |
|---|---|---|---|---|---|
| PIORDY | Main I/O | IDE Device | Static | Low | Low |
| PME# | Resume I/O | Internal Pull-Up | Driven | Driven | Driven |
| PWRBTN# | Resume I/O | Internal Pull-Up | Driven | Driven | Driven |
| PWROK | RTC | System Power Supply | Driven | Low | Low |
| RCIN# | Main I/O | External Microcontroller | High | Low | Low |
| REQ[5:0]# | Main I/O | PCI Master | Driven | Low | Low |
| REQ[B:A]# | Main I/O | PC/PCI Devices | Driven | Low | Low |
| RI# | Resume I/O | Serial Port Buffer | Driven | Driven | Driven |
| RSMRST# | RTC | External RC Circuit | High | High | High |
| RTCRST# | RTC | External RC Circuit | High | High | High |
| SATA0RXP, SATA0RXN SATA1RXP, SATA1RXN | Main Logic | SATA Device | Driven | Driven | Driven |
| SATARBIAS# | Main Logic | External Pull-Down | Driven | Driven | Driven |
| SDDREQ | Main I/O | IDE Device | Static | Low | Low |
| SERR# | Main I/O | PCI Bus Peripherals | High | Low | Low |
| SIORDY | Main I/O | IDE Device | Static | Low | Low |
| SMBALERT# | Resume I/O | External Pull-Up | Driven | Driven | Driven |
| SYS_RESET# | Resume I/O | External Circuit | Driven | Driven | Driven |
| THRM# | Main I/O | Thermal Sensor | Driven | Low | Low |
| THRMTRIP# | CPU I/O | Thermal Sensor | Driven | Low | Low |
| USBRBIAS# | Resume I/O | External Pull-Down | Driven | Driven | Driven |
| VRMPWRGD | Main I/O | Processor Voltage Regulator | High | Low | Low |

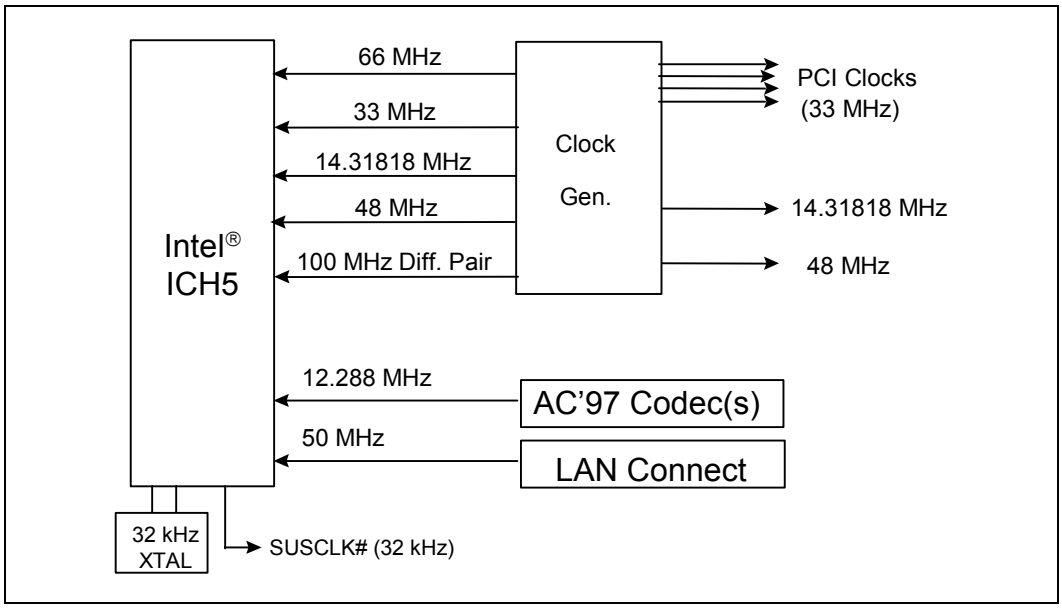
# Intel® ICH5 and System Clock Domains 4

Table 30 shows the system clock domains. Figure 4 shows the assumed connection of the various system components, including the clock generator. For complete details of the system clocking solution, refer to the system's clock generator component specification.

**Table 30. Intel® ICH5 and System Clock Domains**

| Clock Domain | Frequency | Source | Usage |
|---|---|---|---|
| ICH5 CLK100 | 100 MHz | Main Clock Generator | Differential clock pair used for SATA. |
| ICH5 CLK66 | 66 MHz | Main Clock Generator | Hub I/F, Processor I/F. Shut off during S3 or below. |
| ICH5 PCICLK | 33 MHz | Main Clock Generator | Free-running PCI Clock to the Intel® ICH5. This clock remains on during S0 and S1 state, and is expected to be shut off during S3 or below. |
| System PCI | 33 MHz | Main Clock Generator | PCI Bus, LPC I/F. These only go to external PCI and LPC devices. |
| ICH5 CLK48 | 48 MHz | Main Clock Generator | Super I/O, USB controllers. Expected to be shut off during S3 or below. |
| ICH5 CLK14 | 14.31818 MHz | Main Clock Generator | Used for ACPI timer and high-precision event timers. Expected to be shut off during S3 or below. |
| ICH5 AC_BIT_CLK | 12.288 MHz | AC '97 Codec | AC-link. Generated by AC '97 codec. Can be shut by codec in D3. Expected to be shut off during S3 or below. |
| LAN_CLK | 5 to 50 MHz | LAN Connect Component | Generated by the LAN Connect component. Expected to be shut off during S3 or below. |

**Figure 4. Conceptual System Clock Diagram**



# Functional Description 5

This chapter describes the functions and interfaces of the ICH5.

## 5.1 Hub Interface to PCI Bridge (D30:F0)

The hub interface to PCI bridge resides in PCI Device 30, Function 0 on bus #0. This portion of the ICH5 implements the buffering and control logic between PCI and the hub interface. The arbitration for the PCI bus is handled by this PCI device. The PCI decoder in this device must decode the ranges for the hub interface. All register contents are lost when core well power is removed.

### 5.1.1 PCI Bus Interface

The ICH5 PCI interface provides a 33 MHz, *PCI Local Bus Specification, Revision 2.3*-compliant implementation. All PCI signals are 5 V tolerant (except PME#). The ICH5 integrates a PCI arbiter that supports up to six external PCI bus masters in addition to the internal ICH5 requests.

Note that most transactions targeted to the ICH5 first appear on the external PCI bus before being claimed back by the ICH5. The exceptions are I/O cycles involving USB, IDE, SATA, and AC ’97. These transactions complete over the hub interface without appearing on the external PCI bus. Configuration cycles targeting USB, IDE, SATA, or AC ’97 appear on the PCI bus. If the ICH5 is programmed for positive decode, the ICH5 claims the cycles appearing on the external PCI bus in medium decode time. If the ICH5 is programmed for subtractive decode, the ICH5 claims these cycles in subtractive time. If the ICH5 is programmed for subtractive decode, these cycles can be claimed by another positive decode agent out on PCI. This architecture enables the ability to boot off of a PCI card that positively decodes the boot cycles. In order to boot off a PCI card it is necessary to keep the ICH5 in subtractive decode mode. When booting off a PCI card, the BOOT_STS bit (bit 2, TCO2 Status Register) will be set.

***Note:*** The ICH5’s AC ’97, IDE and USB controllers cannot perform peer-to-peer traffic.

***Note:*** PCI Bus Masters should not use memory area locations as a target if that area is programmed to anything but Read/Write.

***Note:*** PCI configuration write cycles, initiated by the processor, with the following characteristics are converted to a Special Cycle with the Shutdown message type.

- Device Number (AD[15:11]) = 11111
- Function Number (AD[10:8]) = 111
- Register Number (AD[7:2]) = 000000
- Data = 00h
- Bus number matches secondary bus number

*Note:* If the processor issues a locked cycle to a resource that is too slow (e.g., PCI), the ICH5 does not allow upstream requests to be performed until the cycle completes. This may be critical for isochronous buses that assume certain timing for their data flow (e.g., AC '97 or USB). Devices on these buses may suffer from underrun if the asynchronous traffic is too heavy. Underrun means that the same data is sent over the bus while ICH5 is not able to issue a request for the next data. Snoop cycles are not permitted while the front side bus is locked.

*Note:* Locked cycles are assumed to be rare. Locks by PCI targets are assumed to exist for a short duration (a few microseconds at most). If a system has a very large number of locked cycles and some that are very long, the system will definitely experience underruns and overruns. The units most likely to have problems are the AC '97 controller and the USB controllers. Other units could get underruns/overruns, but are much less likely. The IDE controller (due to its stalling capability on the cable) should not get any underruns or overruns.

### 5.1.2 PCI-to-PCI Bridge Model

From a software perspective, the ICH5 contains a PCI-to-PCI bridge. This bridge connects the hub interface to the PCI bus. By using the PCI-to-PCI bridge software model, the ICH5 can have its decode ranges programmed by existing plug-and-play software such that PCI ranges do not conflict with AGP and graphics aperture ranges in the Host controller.

### 5.1.3 IDSEL to Device Number Mapping

When addressing devices on the external PCI bus (with the PCI slots), the ICH5 asserts one address signal as an IDSEL. When accessing device 0, the ICH5 asserts AD16. When accessing Device 1, the ICH5 asserts AD17. This mapping continues all the way up to device 15 where the ICH5 asserts AD31. Note that the ICH5's internal functions (AC '97, IDE, USB, SATA and PCI Bridge) are enumerated like they are on a separate PCI bus (the hub interface) from the external PCI bus. The integrated LAN controller is Device 8 on the ICH5's PCI bus, and hence it uses AD24 for IDSEL.

### 5.1.4 SERR# Functionality

There are several internal and external sources that can cause SERR#. The ICH5 can be programmed to cause an NMI based on detecting that an SERR# condition has occurred. The NMI can also be routed to instead cause an SMI#. Note that the ICH5 does not drive the external PCI bus SERR# signal active onto the PCI bus. The external SERR# signal is an input into the ICH5 driven only by external PCI devices. The conceptual logic diagrams in Figure 5 and Figure 6 illustrate all sources of SERR#, along with their respective enable and status bits. Figure 7 shows how the ICH5 error reporting logic is configured for NMI# generation.

**Figure 5. Primary Device Status Register Error Reporting Logic**

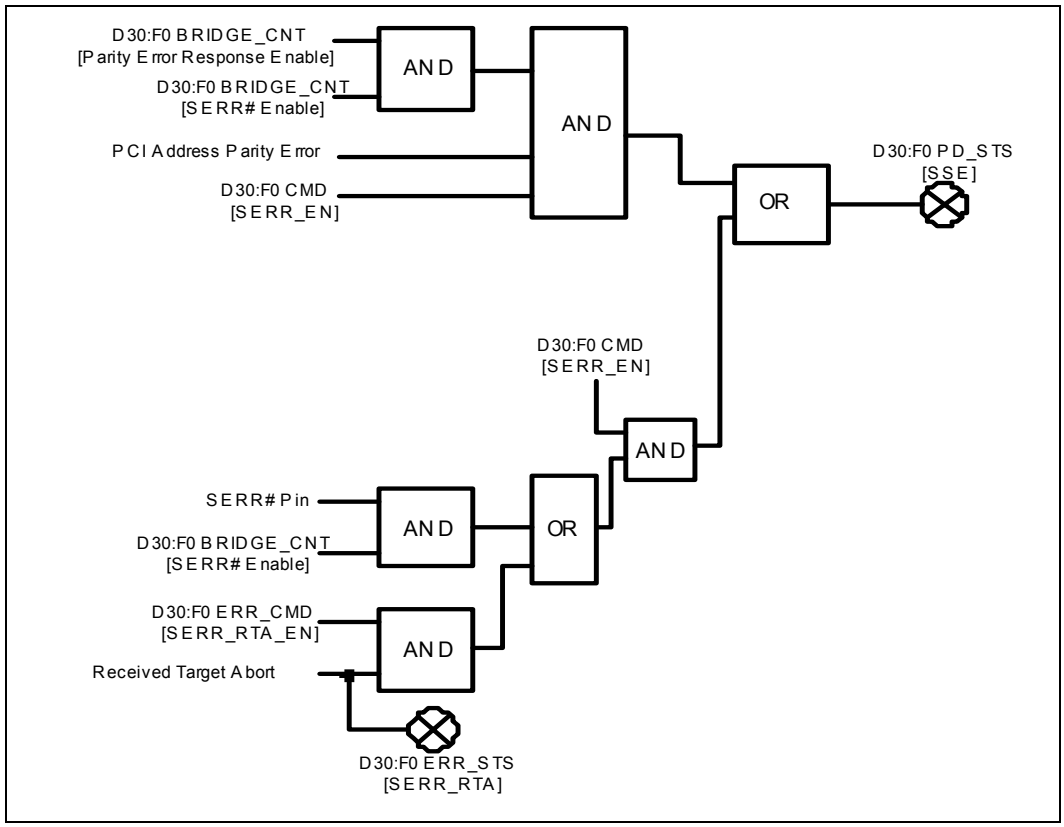



**Figure 6. Secondary Status Register Error Reporting Logic**

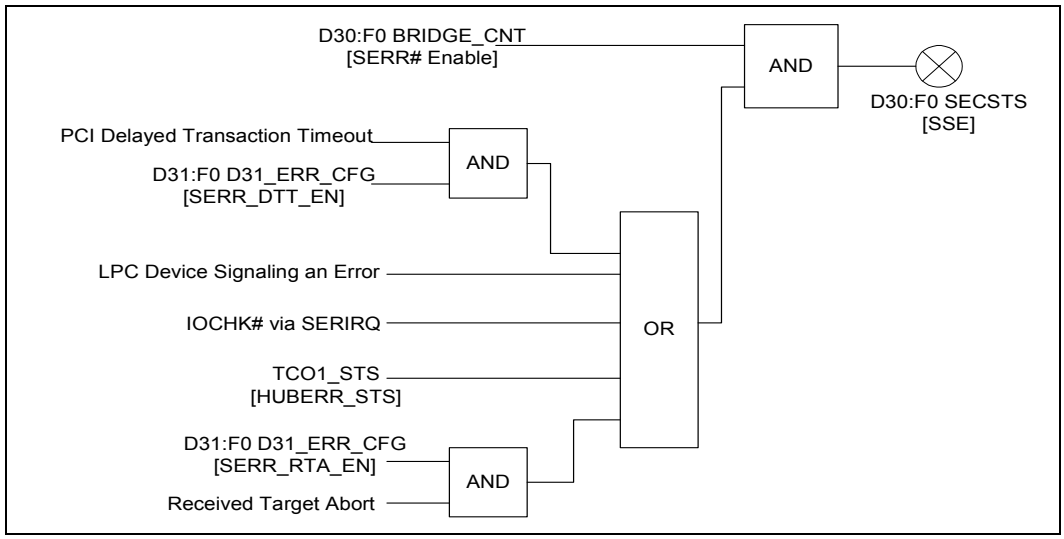
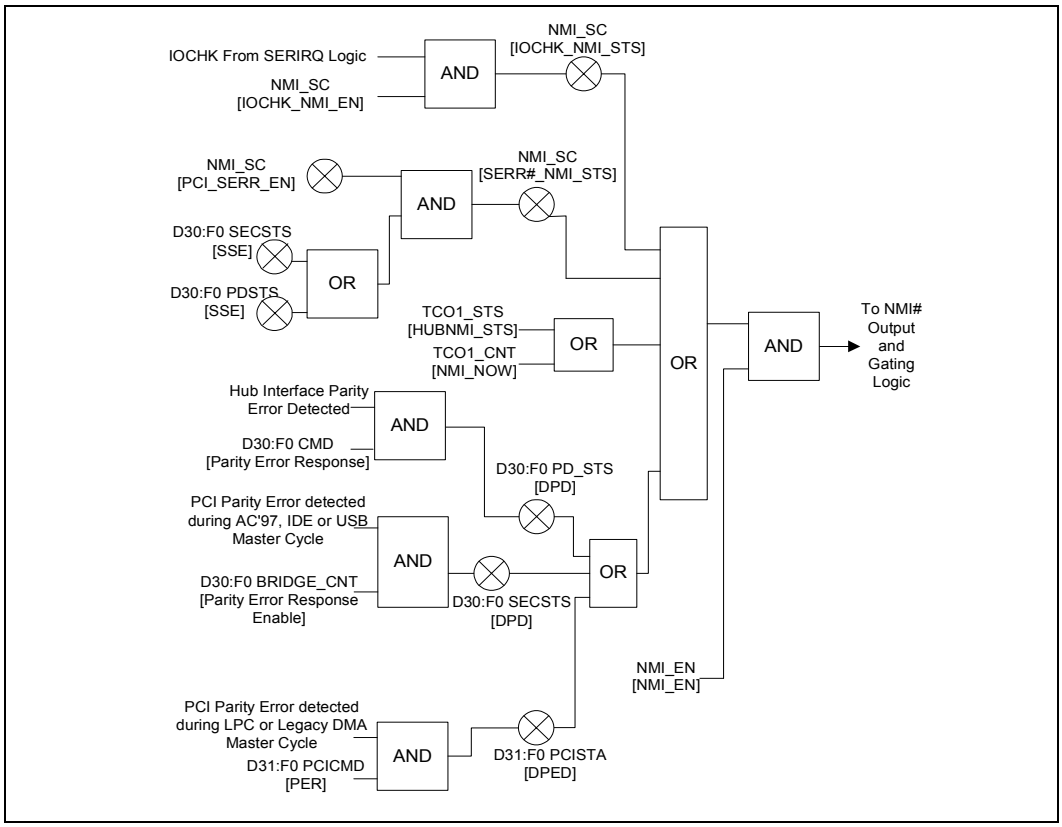


**Figure 7. NMI# Generation Logic**




## 5.1.5 Parity Error Detection

The ICH5 can detect and report different parity errors in the system. The ICH5 can be programmed to cause an NMI (or SMI# if NMI is routed to SMI#) based on detecting a parity error. The conceptual logic diagram in Figure 7 details all the parity errors that the ICH5 can detect, along with their respective enable bits, status bits, and the results.

For hub interface-to-PCI data packets, with MCH's that generate HI parity, the ICH5 provides the ability to generate bad parity on all data driven by the ICH5 when bad data parity was detected on hub interface. This prevents PCI agents that are capable of checking parity from taking corrupted data unknowingly. This state can be entered due to either hub interface-to-PCI write data or hub interface-to-PCI read completion data. This mode is enabled by D30.F0.50h.bit 19 and reported in D30.F0.92h.bit 0.

***Note:*** The HP_Unsupported bit (D30:F0:40h bit 20) must be cleared for any of the parity checking enable bits to have any effect.

***Note:*** If NMIs are enabled, and parity error checking on PCI is also enabled, then parity errors will cause an NMI. Some operating systems will not attempt to recover from this NMI, since it considers the detection of a PCI error to be a catastrophic event.

## 5.1.6 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 the ICH5. The *PCI Local Bus Specification, Revision 2.3* defines two mechanisms to access configuration space, Mechanism 1 and Mechanism 2. The ICH5 only supports Mechanism 1.

Configuration cycles for PCI Bus 0 devices 2 through 31, and for PCI Bus numbers greater than 0 are sent towards the ICH5 from the host controller. The ICH5 compares the non-zero Bus Number with the Secondary Bus Number and Subordinate Bus number registers of its PCI-to-PCI bridge to determine if the configuration cycle is meant for primary PCI or a downstream PCI bus.

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

### 5.1.6.1 Type 0 to Type 0 Forwarding

When a Type 0 configuration cycle is received on hub interface to any function other than EHCI or AC '97, the ICH5 forwards these cycles to PCI and then reclaims them. The ICH5 uses address bits AD[15:13] to communicate the ICH5 device numbers in Type 0 configuration cycles. If the Type 0 cycle on hub interface specifies any device number other than 29, 30 or 31, the ICH5 will not set any address bits in the range AD[31:11] during the corresponding transaction on PCI. Table 31 shows the device number translation.

**Table 31. Type 0 Configuration Cycle Device Number Translation**

| Device # in Hub Interface Type 0 Cycle | AD[31:11] during Address Phase of Type 0 Cycle on PCI |
|---|---|
| 0 through 28 | 0000000000000000_00000b |
| 29 | 0000000000000000_00100b |
| 30 | 0000000000000000_01000b |
| 31 | 0000000000000000_10000b |

The ICH5 logic generates single DWord configuration read and write cycles on the PCI bus. The ICH5 generates a Type 0 configuration cycle for configurations to the bus number matching the PCI bus. Type 1 configuration cycles are converted to Type 0 cycles in this case. If the cycle is targeting a device behind an external bridge, the ICH5 runs a Type 1 cycle on the PCI bus.

### 5.1.6.2 Type 1 to Type 0 Conversion

When the bus number for the Type 1 configuration cycle matches the PCI (Secondary) bus number, the ICH5 converts the address as follows:

1. For device numbers 0 through 15, only 1 bit of the PCI address [31:16] is set. If the device number is 0, AD16 is set; if the device number is 1, AD17 is set; etc.
2. The ICH5 always drives 0s on bits AD[15:11] when converting Type 1 configurations cycles to Type 0 configuration cycles on PCI.
3. Address bits [10:1] are also be passed unchanged to PCI.
4. Address bit 0 is changed to 0.

### 5.1.7 PCI Dual Address Cycle (DAC) Support

The ICH5 supports Dual Address Cycle (DAC) format on PCI for cycles from PCI initiators to main memory. This allows PCI masters to generate an address up to 44 bits. The size of the actual supported memory space is determined by the memory controller and the processor.

The DAC mode is only supported for PCI adapters and USB EHC, and is not supported for any of the internal PCI masters (IDE, LAN, USB UHC, AC '97, 8237 DMA, etc.).

When a PCI master wants to initiate a cycle with an address above 4 G, it follows the following behavioral rules (See *PCI Local Bus Specification, Revision 2.3*, Section 3.9 for more details):

1. On the first clock of the cycle (when FRAME# is first active), the peripheral uses the DAC encoding on the C/BE# signals. This unique encoding is: 1101.
2. Also during the first clock, the peripheral drives the AD[31:0] signals with the low address.
3. On the second clock, the peripheral drives AD[31:0] with the high address. The address is right justified: A[43:32] appear on AD[12:0]. The value of AD[31:13] is expected to be 0; however, the ICH5 ignores these bits. C/BE# indicate the bus command type (memory read, memory write, etc.)
4. The rest of the cycle proceeds normally.

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

The ICH5'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 ICH5 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 the ICH5's virtual PCI-to-PCI bridge (see Section 5.1.2). This is typically Bus 1, but may be assigned a different number, depending upon system configuration.

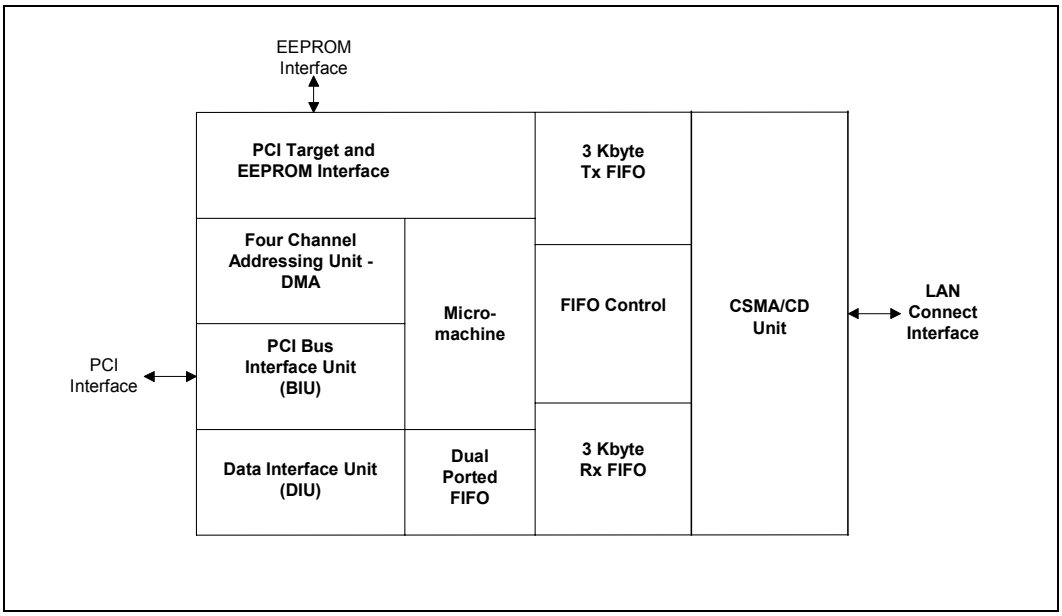
The following summarizes the ICH5 LAN controller features:

- 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
- Support of Wired for Management (WfM) Revision 2.0
- Backward compatible software with 82550, 82557, 82558 and 82559
- TCP/UDP checksum off load capabilities
- Support for Intel's Adaptive Technology

## 5.2.1 LAN Controller Architectural Overview

Figure 8 is a high-level block diagram of the ICH5 integrated LAN controller. It is divided into four main subsystems: a Parallel subsystem, a FIFO subsystem, and the Carrier-Sense Multiple Access with Collision Detect (CSMA/CD) unit.

**Figure 8. Integrated LAN Controller Block Diagram**



#### 5.2.1.1 Parallel Subsystem Overview

The parallel subsystem is broken down into several functional blocks: a PCI bus master interface, a micromachine processing unit and its corresponding microcode ROM, and a PCI Target Control/ EEPROM/ interface. The parallel subsystem also interfaces to the FIFO subsystem, passing data (such as transmit, receive, and configuration data) and command and status parameters between these two blocks.

The PCI bus master interface provides a complete interface to the PCI bus and is compliant with the *PCI Local Bus Specification, Revision 2.3*. The LAN controller provides 32 bits of addressing and data, as well as the complete control interface to operate on the PCI bus. As a PCI target, it follows the PCI configuration format which allows all accesses to the LAN controller to be automatically mapped into free memory and I/O space upon initialization of a PCI system. For processing of transmit and receive frames, the integrated LAN controller operates as a master on the PCI bus, initiating zero wait-state transfers for accessing these data parameters.

The LAN controller control/status register block is part of the PCI target element. The control/ status register block consists of the following LAN controller internal control registers: System Control Block (SCB), PORT, EEPROM Control and Management Data Interface (MDI) Control.

The micromachine is an embedded processing unit contained in the LAN controller that enables Adaptive Technology. The micromachine accesses the LAN controller's microcode ROM, working its way through the opcodes (or instructions) contained in the ROM to perform its functions. Parameters accessed from memory, such as pointers to data buffers, are also used by the micromachine during the processing of transmit or receive frames by the LAN controller. A typical micromachine function is to transfer a data buffer pointer field to the LAN controller's DMA unit for direct access to the data buffer. The micromachine is divided into two units, Receive Unit and Command Unit which includes transmit functions. These two units operate independently and concurrently. Control is switched between the two units according to the microcode instruction flow. The independence of the Receive and Command units in the micromachine allows the LAN controller to execute commands and receive incoming frames simultaneously, with no real-time processor intervention.

The LAN controller contains an interface to an external serial EEPROM. The EEPROM is used to store relevant information for a LAN connection such as node address, as well as board manufacturing and configuration information. Both read and write accesses to the EEPROM are supported by the LAN controller. Information on the EEPROM interface is detailed in Section 5.2.3.

### 5.2.1.2 FIFO Subsystem Overview

The ICH5 LAN controller FIFO subsystem consists of a 3-KB transmit FIFO and 3-KB receive FIFO. Each FIFO is unidirectional and independent of the other. The FIFO subsystem serves as the interface between the LAN controller parallel side and the serial CSMA/CD unit. It provides a temporary buffer storage area for frames as they are either being received or transmitted by the LAN controller, which improves performance:

- Transmit frames can be queued within the transmit FIFO, allowing back-to-back transmission within the minimum Interframe Spacing (IFS).
- The storage area in the FIFO allows the LAN controller to withstand long PCI bus latencies without losing incoming data or corrupting outgoing data.
- The ICH5 LAN controller's transmit FIFO threshold allows the transmit start threshold to be tuned to eliminate underruns while concurrent transmits are being performed.
- The FIFO subsection allows extended PCI zero wait-state burst accesses to or from the LAN controller for both transmit and receive frames since the transfer is to the FIFO storage area rather than directly to the serial link.
- Transmissions resulting in errors (collision detection or data underrun) are retransmitted directly from the LAN controller's FIFO, increasing performance and eliminating the need to re-access this data from the host system.
- Incoming runt receive frames (in other words, frames that are less than the legal minimum frame size) can be discarded automatically by the LAN controller without transferring this faulty data to the host system.

### 5.2.1.3 Serial CSMA/CD Unit Overview

The CSMA/CD unit of the ICH5 LAN controller allows it to be connected to the 82562ET/EM/EZ/EX 10/100 Mbps Ethernet LAN Connect components. The CSMA/CD unit performs all of the functions of the 802.3 protocol such as frame formatting, frame stripping, collision handling, deferral to link traffic, etc. The CSMA/CD unit can also be placed in a full-duplex mode that allows simultaneous transmission and reception of frames.

## 5.2.2 LAN Controller PCI Bus Interface

As a Fast Ethernet controller, the role of the ICH5 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.

### 5.2.2.1 Bus Slave Operation

The ICH5 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.

**Read Accesses:** The processor, as the initiator, drives address lines AD[31:0], the command and byte enable lines C/BE[3:0]# and the control lines IRDY# and FRAME#. As a slave, the LAN controller controls the TRDY# signal and provides valid data on each data access. The LAN controller allows the processor to issue only one read cycle when it accesses the CSR, generating a disconnect by asserting the STOP# signal. The processor can insert wait-states by deasserting IRDY# when it is not ready.

**Write Accesses:** The processor, as the initiator, drives the address lines AD[31:0], the command and byte enable lines C/BE[3:0]# and the control lines IRDY# and FRAME#. It also provides the LAN controller with valid data on each data access immediately after asserting IRDY#. The LAN controller controls the TRDY# signal and asserts it from the data access. The LAN controller allows the processor to issue only one I/O write cycle to the Control/Status Registers, generating a disconnect by asserting the STOP# signal. This is true for both memory mapped and I/O mapped accesses.

### 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 always 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.

### 5.2.2.2 Bus Master Operation

As a PCI Bus Master, the ICH5 integrated LAN controller initiates memory cycles to fetch data for transmission or deposit received data and for accessing the memory resident control structures. The LAN controller performs zero wait-state burst read and write cycles to the host main memory. For bus master cycles, the LAN controller is the initiator and the host main memory (or the PCI host bridge, depending on the configuration of the system) is the target.

The processor provides the LAN controller with action commands and pointers to the data buffers that reside in host main memory. The LAN controller independently manages these structures and initiates burst memory cycles to transfer data to and from them. The LAN controller uses the Memory Read Multiple (MR Multiple) command for burst accesses to data buffers and the Memory Read Line (MR Line) command for burst accesses to control structures. For all write accesses to the control structure, the LAN controller uses the Memory Write (MW) command. For write accesses to data structure, the LAN controller may use either the Memory Write or Memory Write and Invalidate (MWI) commands.

**Read Accesses:** The LAN controller performs block transfers from host system memory in order to perform frame transmission on the serial link. In this case, the LAN controller initiates zero wait-state memory read burst cycles for these accesses. The length of a burst is bounded by the system and the LAN controller's internal FIFO. The length of a read burst may also be bounded by the value of the Transmit DMA Maximum Byte Count in the Configure command. The Transmit DMA Maximum Byte Count value indicates the maximum number of transmit DMA PCI cycles that will be completed after an LAN controller internal arbitration.

The LAN controller, as the initiator, drives the address lines AD[31:0], the command and byte enable lines C/BE[3:0]# and the control lines IRDY# and FRAME#. The LAN controller asserts IRDY# to support zero wait-state burst cycles. The target signals the LAN controller that valid data is ready to be read by asserting the TRDY# signal.

**Write Accesses:** The LAN controller performs block transfers to host system memory during frame reception. In this case, the LAN controller initiates memory write burst cycles to deposit the data, usually without wait-states. The length of a burst is bounded by the system and the LAN controller's internal FIFO threshold. The length of a write burst may also be bounded by the value of the Receive DMA Maximum Byte Count in the Configure command. The Receive DMA Maximum Byte Count value indicates the maximum number of receive DMA PCI transfers that will be completed before the LAN controller internal arbitration.

The LAN controller, as the initiator, drives the address lines AD[31:0], the command and byte enable lines C/BE[3:0]# and the control lines IRDY# and FRAME#. The LAN controller asserts IRDY# to support zero wait-state burst cycles. The LAN controller also drives valid data on AD[31:0] lines during each data phase (from the first clock and on). The target controls the length and signals completion of a data phase by deassertion and assertion of TRDY#.

**Cycle Completion:** The LAN controller completes (terminates) its initiated memory burst cycles in the following cases:

- **Normal Completion:** All transaction data has been transferred to or from the target device (for example, host main memory).
- **Backoff:** Latency Timer has expired and the bus grant signal (GNT#) was removed from the LAN controller by the arbiter, indicating that the LAN controller has been preempted by another bus master.
- **Transmit or Receive DMA Maximum Byte Count:** The LAN controller burst has reached the length specified in the Transmit or Receive DMA Maximum Byte Count field in the Configure command block.
- **Target Termination:** The target may request to terminate the transaction with a target-disconnect, target-retry, or target-abort. In the first two cases, the LAN controller initiates the cycle again. In the case of a target-abort, the LAN controller sets the Received Target-Abort bit in the PCI Configuration Status field (PCI Configuration Status register, bit 12) and does not re-initiate the cycle.
- **Master Abort:** The target of the transaction has not responded to the address initiated by the LAN controller (in other words, DEVSEL# has not been asserted). The LAN controller simply deasserts FRAME# and IRDY# as in the case of normal completion.
- **Error Condition:** In the event of parity or any other system error detection, the LAN controller completes its current initiated transaction. Any further action taken by the LAN controller depends on the type of error and other conditions.

### Memory Write and Invalidate

The LAN controller has four Direct Memory Access (DMA) channels. Of these four channels, the Receive DMA is used to deposit the large number of data bytes received from the link into system memory. The Receive DMA uses both the Memory Write (MW) and the Memory Write and Invalidate (MWI) commands. To use MWI, the LAN controller must guarantee the following:

- Minimum transfer of one cache line.
- Active byte enable bits (or BE[3:0]# are all low) during MWI access.
- The LAN controller may cross the cache line boundary only if it intends to transfer the next cache line too.

To ensure the above conditions, the LAN controller may use the MWI command only under the following conditions:

- The Cache Line Size (CLS) written in the CLS register during PCI configuration is 8 or 16 DWords.
- The accessed address is cache line aligned.
- The LAN controller has at least 8 or 16 DWords of data in its receive FIFO.
- There are at least 8 or 16 DWords of data space left in the system memory buffer.
- The MWI Enable bit in the PCI Configuration Command register, bit 4, should is set to 1b.
- The MWI Enable bit in the LAN Controller Configure command should is set to 1b.

If any one of the above conditions does not hold, the LAN controller uses the MW command. If a MWI cycle has started and one of the conditions is no longer valid (for example, the data space in the memory buffer is now less than CLS), then the LAN controller terminates the MWI cycle at the end of the cache line. The next cycle is either a MW or MWI cycle, depending on the conditions listed above.

If the LAN controller started a MW cycle and reached a cache line boundary, it either continues or terminates the cycle depending on the Terminate Write on Cache Line configuration bit of the LAN Controller Configure command (byte 3, bit 3). If this bit is set, the LAN controller terminates the MW cycle and attempts to start a new cycle. The new cycle is a MWI cycle if this bit is set and all of the above listed conditions are met. If the bit is not set, the LAN controller continues the MW cycle across the cache line boundary if required.

### Read Align

The Read Align feature enhances the LAN controller's performance in cache line oriented systems. In these particular systems, starting a PCI transaction on a non-cache line aligned address may cause low performance.

To resolve this performance anomaly, the LAN controller attempts to terminate transmit DMA cycles on a cache line boundary and start the next transaction on a cache line aligned address. This feature is enabled when the Read Align Enable bit is set in the LAN Controller Configure command (byte 3, bit 2).

If this bit is set, the LAN controller operates as follows:

- When the LAN controller is almost out of resources on the transmit DMA (that is, the transmit FIFO is almost full), it attempts to terminate the read transaction on the nearest cache line boundary when possible.
- When the arbitration counter's feature is enabled (i.e., the Transmit DMA Maximum Byte Count value is set in the Configure command), the LAN controller switches to other pending DMAs on cache line boundary only.

***Note:***

1. This feature is not recommended for use in non-cache line oriented systems since it may cause shorter bursts and lower performance.
2. This feature should be used only when the CLS register in PCI Configuration space is set to 8 or 16.
3. The LAN controller reads all control data structures (including Receive Buffer Descriptors) from the first DWord (even if it is not required) in order to maintain cache line alignment.

### Error Handling

**Data Parity Errors:** As an initiator, the LAN controller checks and detects data parity errors that occur during a transaction. If the Parity Error Response bit is set (PCI Configuration Command register, bit 6), the LAN controller also asserts PERR# and sets the Data Parity Detected bit (PCI Configuration Status register, bit 8). In addition, if the error was detected by the LAN controller during read cycles, it sets the Detected Parity Error bit (PCI Configuration Status register, bit 15).

### 5.2.2.3 PCI Power Management

Enhanced support for the power management standard, *PCI Local Bus Specification, Revision 2.3*, is provided in the ICH5 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 sections below describe these events, the LAN controller power states, and estimated power consumption at each power state.

#### Power States

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.

- D0 Power State

  The device is fully functional in the D0 power state. In this state, the LAN controller receives full power and should be providing full functionality. In the LAN controller the D0 state is partitioned into two substates, D0 Uninitialized (D0u) and D0 Active (D0a).

  D0u is the LAN controller's initial power state following a PCI RST#. While in the D0u state, the LAN controller has PCI slave functionality to support its initialization by the host and supports Wake on LAN mode. Initialization of the CSR, Memory, or I/O Base Address Registers in the PCI Configuration space switches the LAN controller from the D0u state to the D0a state.

  In the D0a state, the LAN controller provides its full functionality and consumes its nominal power. In addition, the LAN controller supports wake on link status change (see Section 5.2.2.5). While it is active, the LAN controller requires a nominal PCI clock signal (in other words, a clock frequency greater than 16 MHz) for proper operation. The LAN controller supports a dynamic standby mode. In this mode, the LAN controller is able to save almost as much power as it does in the static power-down states. The transition to or from standby is done dynamically by the LAN controller and is transparent to the software.

- D1 Power State

  In order for a device to meet the D1 power state requirements, as specified in the *Advanced Configuration and Power Interface, Version 2.0b Specification*, it must not allow bus transmission or interrupts; however, bus reception is allowed. Therefore, device context may be lost and the LAN controller does not initiate any PCI activity. In this state, the LAN controller responds only to PCI accesses to its configuration space and system wake-up events.

  The LAN controller retains link integrity and monitors the link for any wake-up events (e.g., wake-up packets or link status change). Following a wake-up event, the LAN controller asserts the PME# signal.

- D2 Power State

  The ACPI D2 power state is similar in functionality to the D1 power state. In addition to D1 functionality, the LAN controller can provide a lower power mode with wake-on-link status change capability. The LAN controller may enter this mode if the link is down while the LAN controller is in the D2 state. In this state, the LAN controller monitors the link for a transition from an invalid to a valid link.

  The sub-10 mA state due to an invalid link can be enabled or disabled by a configuration bit in the Power Management Driver Register (PMDR). The LAN controller will consume in D2 <10 mA, regardless of the link status. It is the LAN Connect component that consumes much less power during link down; hence, the LAN controller in this state can consume <10 mA.

- D3 Power State

  In the D3 power state, the LAN controller has the same capabilities and consumes the same amount of power as it does in the D2 state. However, it enables the PCI system to be in the B3 state. If the PCI system is in the B3 state (in other words, no PCI power is present), the LAN controller provides wake-up capabilities. If PME is disabled, the LAN controller does not provide wake-up capability or maintain link integrity. In this mode the LAN controller consumes its minimal power.

  The LAN controller enables a system to be in a sub-5 Watt state (low-power state) and still be virtually connected. More specifically, the LAN controller supports full wake-up capabilities while it is in the D3 cold state. The LAN controller is in the ICH5 resume well, which enables it to provide wake-up functionality while the PCI power is off.

#### 5.2.2.4 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.

### 5.2.2.5 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 pre-defined 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.

### 5.2.2.6 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.

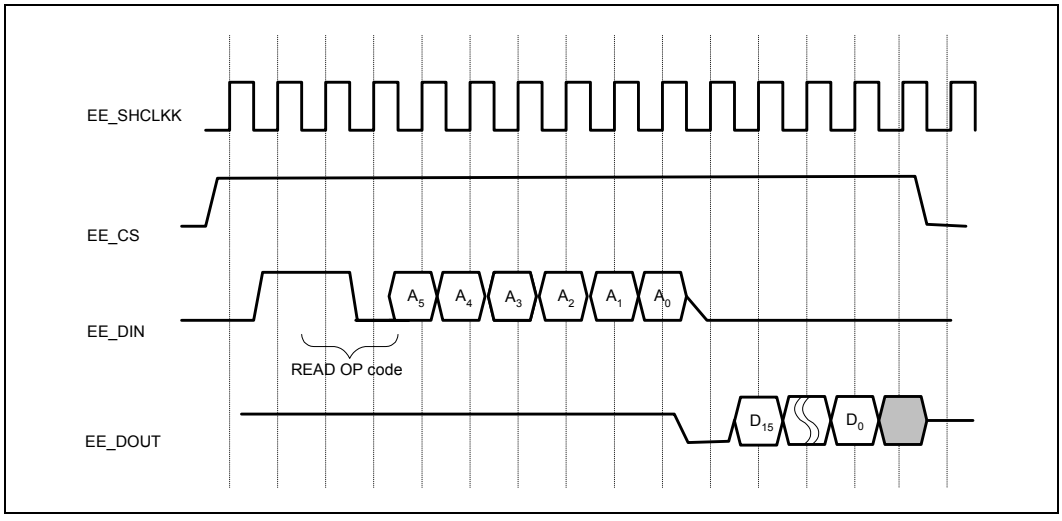
The LAN controller switches from Wake on LAN mode to the D0a power state following a setup of the Memory or I/O Base Address Registers in the PCI Configuration space.

## 5.2.3 Serial EEPROM Interface

The serial EEPROM stores configuration data for the ICH5 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, which indicates the entire address field has been transferred to the device. An EEPROM read instruction waveform is shown in Figure 9.

**Figure 9. 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.

### 5.2.4 CSMA/CD Unit

The ICH5 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 the ICH5's LAN Connect interface signals.

#### 5.2.4.1 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. The ICH5 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.

#### 5.2.4.2 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.

#### 5.2.4.3 Address Filtering Modifications

The LAN controller can be configured to ignore 1 bit when checking for its Individual Address (IA) on incoming receive frames. The address bit, known as the Upper/Lower (U/L) bit, is the second least significant bit of the first byte of the IA. This bit may be used, in some cases, as a priority indication bit. When configured to do so, the LAN controller passes any frame that matches all other 47 address bits of its IA, regardless of the U/L bit value.

This configuration only affects the LAN controller specific IA and not multicast, multi-IA or broadcast address filtering. The LAN controller does not attribute any priority to frames with this bit set, it simply passes them to memory regardless of this bit.

#### 5.2.4.4 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.

### 5.2.5 Media Management Interface

The management interface allows the processor to control the platform LAN Connect component via a control register in the ICH5 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.

### 5.2.6 TCO Functionality

The ICH5 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.

#### 5.2.6.1 Advanced TCO Mode

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

**Table 32. Advanced TCO Functionality**

| Power State | TCO Controller Functionality |
|---|---|
| D0 nominal | Transmit<br>Set Receive TCO Packets<br>Receive TCO Packets<br>Read ICH5 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>Config 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, the ICH5 LAN controller first completes the current transmit DMA, sets the TCO Request bit in the PMDR register (see Section 7.2), 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 7.2.14). Following the completion of the transmit operation, the ICH5 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

The ICH5 LAN controller supports receive flow towards the TCO controller. The ICH5 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 the ICH5 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, the ICH5 increments its nominal Receive statistic counters as well as the Receive TCO counter.

**Dx>0 Power State:** While the ICH5 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, the ICH5 may transfer TCO packets to the TCO controller. At this state, TCO packets are posted first to the host memory, then read by the ICH5, 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, the ICH5 increments the receive TCO static counter, clears the TCO request bit, and resumes normal control.

### Read ICH5 Status (PM and Link State)

The TCO controller is capable of reading the ICH5 power state and link status. Following a status change, the ICH5 asserts LINKALERT# and then the TCO can read its new power state.

### Set Force TCO Mode

The TCO controller put the ICH5 into the Force TCO mode. The ICH5 is set back to the nominal operation following a PCIRST#. Following the transition from nominal mode to a TCO mode, the ICH5 aborts transmission and reception and loses its memory structures. The TCO may configure the ICH5 before it starts transmission and reception if required.

***Note:*** The Force TCO is a destructive command. It causes the ICH5 to lose its memory structures, and during the Force TCO mode the ICH5 ignores any PCI accesses. Therefore, it is highly recommended to use this command by the TCO controller at system emergency only.

## 5.3 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, OS 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 OS 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 the ICH5) and any SMBus platform sensors. The SMBus host is accessible by the system software, including software running on the OS 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.

ICH5 provides an input and an output EEPROM interface. The EEPROM contains the LAN controller configuration and the ASF controller configuration/packet information.

### 5.3.1 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 OS 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)

## 5.3.2 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 ICH5 ASF controller will be disabled. The external ASF device will have access to the SMBus controller.

### 5.3.2.1 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.

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

To support the ICH5 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).

### 5.3.2.3 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.*

### 5.3.2.4 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.

### 5.3.2.5 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*.

## 5.3.3 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.

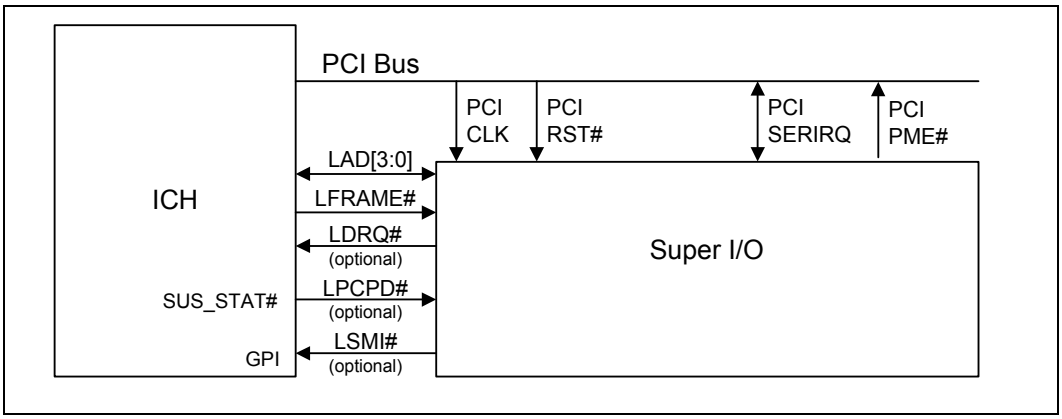
# 5.4 LPC Bridge (w/ System and Management Functions) (D31:F0)

The LPC Bridge function of the ICH5 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.

## 5.4.1 LPC Interface

The ICH5 implements an LPC interface as described in the *Low Pin Count Interface Specification, Revision 1.1*. The LPC interface to the ICH5 is shown in Figure 10. Note that the ICH5 implements all of the signals that are shown as optional, but peripherals are not required to do so.

**Figure 10. LPC Interface Diagram**



### 5.4.1.1 LPC Cycle Types

The ICH5 implements all of the cycle types described in the *Low Pin Count Interface Specification, Revision 1.0*. Table 33 shows the cycle types supported by the ICH5.

**Table 33. LPC Cycle Types Supported**

| Cycle Type | Comment |
|---|---|
| Memory Read | Single: 1 byte only |
| Memory Write | Single: 1 byte only |
| I/O Read | 1 byte only. Intel® ICH5 breaks up 16- and 32-bit processor cycles into multiple 8-bit transfers. See Note 1 below. |
| I/O Write | 1 byte only. ICH5 breaks up 16- and 32-bit processor cycles into multiple 8-bit transfers. See Note 1 below. |
| 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 2 below) |
| Bus Master Write | Can be 1, 2, or 4 bytes. (See Note 2 below) |

**NOTES:**
1. For memory cycles below 16 MB that do not target enabled flash BIOS ranges, the ICH5performs standard LPC memory cycles. It only attempts 8-bit transfers. If the cycle appears on PCI as a 16-bit transfer, it appears as two consecutive 8-bit transfers on LPC. Likewise, if the cycle appears as a 32-bit transfer on PCI, it appears as four consecutive 8-bit transfers on LPC. If the cycle is not claimed by any peripheral, it is subsequently aborted, and the ICH5 returns a value of all 1s to the processor. This is done to maintain compatibility with ISA memory cycles where pull-up resistors would keep the bus high if no device responds.
2. Bus Master Read or Write cycles must be naturally aligned. For example, a 1-byte transfer can be to any address. However, the 2-byte transfer must be word aligned (i.e., with an address where A0=0). A DWord transfer must be DWord aligned (i.e., with an address where A1and A0 are both 0).

### 5.4.1.2 Start Field Definition

**Table 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.

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

The ICH5 always drives bit 0 of this field to 0. Peripherals running bus master cycles must also drive bit 0 to 0. Table 35 shows the valid bit encodings.

**Table 35. Cycle Type Bit Definitions**

| Bits[3:2] | Bit1 | Definition |
|---|---|---|
| 00 | 0 | I/O Read |
| 00 | 1 | I/O Write |
| 01 | 0 | Memory Read |
| 01 | 1 | Memory Write |
| 10 | 0 | DMA Read |
| 10 | 1 | DMA Write |
| 11 | x | Reserved. If a peripheral performing a bus master cycle generates this value, the Intel® ICH5 aborts the cycle. |

### 5.4.1.4 SIZE

Bits[3:2] are reserved. The ICH5 always drives them to 00. Peripherals running bus master cycles are also supposed to drive 00 for bits 3:2; however, the ICH5 ignores those bits. Bits[1:0] are encoded as listed in Table 36.

**Table 36. Transfer Size Bit Definition**

| Bits[1:0] | Size |
|---|---|
| 00 | 8-bit transfer (1 byte) |
| 01 | 16-bit transfer (2 bytes) |
| 10 | Reserved. The Intel® ICH5 never drives this combination. If a peripheral running a bus master cycle drives this combination, the ICH5 may abort the transfer. |
| 11 | 32-bit transfer (4 bytes) |

### 5.4.1.5 SYNC

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

**Table 37. SYNC Bit Definition**

| Bits[3:0] | 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, the Intel® ICH5 does not use this encoding. Instead, the ICH5 uses the Long Wait encoding (see next encoding below). |
| 0110 | **Long Wait:** Part indicating wait-states, and many wait-states will be added. This encoding driven by the ICH5 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. |

**NOTE:** All other combinations are RESERVED.

### 5.4.1.6 SYNC Time-Out

There are several error cases that can occur on the LPC interface. Table 38 indicates the failing case and the ICH5 response.

**Table 38. Intel® ICH5 Response to Sync Failures**

| Possible Sync Failure | Intel® ICH5 Response |
|---|---|
| Intel®ICH5 starts a Memory, I/O, or DMA cycle, but no device drives a valid SYNC after 4 consecutive clocks. This could occur if the processor tries to access an I/O location to which no device is mapped. | ICH5 aborts the cycle after the fourth clock. |
| ICH5 drives a Memory, I/O, or DMA cycle, and a peripheral drives more than 8 consecutive valid SYNC to insert wait-states using the Short (0101b) encoding for SYNC. This could occur if the peripheral is not operating properly. | Continues waiting |
| ICH5 starts a Memory, I/O, or DMA cycle, and a peripheral drives an invalid SYNC pattern. This could occur if the peripheral is not operating properly or if there is excessive noise on the LPC I/F. | ICH5 aborts the cycle when the invalid Sync is recognized. |

There may be other peripheral failure conditions; however, these are not handled by the ICH5.

### 5.4.1.7 SYNC Error Indication

The SYNC protocol allows the peripheral to report an error via the LAD[3:0] = 1010b encoding. The intent of this encoding is to give peripherals a method of communicating errors to aid higher layers with more robust error recovery.

If the ICH5 was reading data from a peripheral, data will still be transferred in the next two nibbles. This data may be invalid, but it must be transferred by the peripheral. If the ICH5 was writing data to the peripheral, the data had already been transferred.

In the case of multiple byte cycles (e.g., for memory and DMA cycles) an error SYNC terminates the cycle. Therefore, if the ICH5 is transferring 4 bytes from a device, if the device returns the error SYNC in the first byte, the other three bytes will not be transferred.

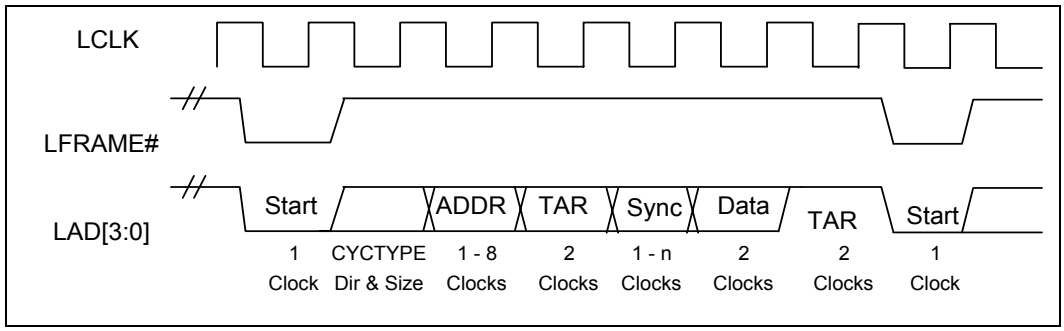
Upon recognizing the SYNC field indicating an error, the ICH5 treats this the same as IOCHK# going active on the ISA bus.

### 5.4.1.8 LFRAME# Usage

#### Start of Cycle

For Memory, I/O, and DMA cycles, the ICH5 asserts LFRAME# for one clock at the beginning of the cycle (Figure 11). During that clock, the ICH5 drives LAD[3:0] with the proper START field.

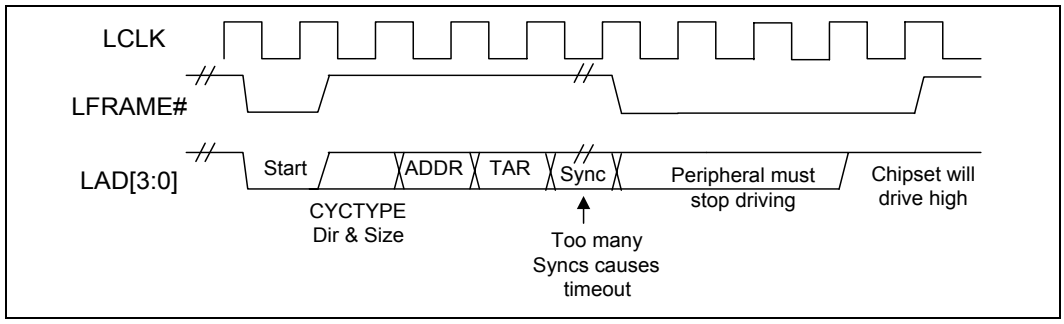
**Figure 11. Typical Timing for LFRAME#**




#### Abort Mechanism

When performing an Abort, the ICH5 drives LFRAME# active for four, consecutive clocks. On the fourth clock, it drives LAD[3:0] to 1111b.

**Figure 12. Abort Mechanism**




The ICH5 performs an abort for the following cases (possible failure cases):

- ICH5 starts a Memory, I/O, or DMA cycle, but no device drives a valid SYNC after four consecutive clocks.
- ICH5 starts a Memory, I/O, or DMA cycle, and the peripheral drives an invalid SYNC pattern.
- A peripheral drives an illegal address when performing bus master cycles.
- A peripheral drives an invalid value.

#### 5.4.1.9 I/O Cycles

For I/O cycles targeting registers specified in the ICH5's decode ranges, the ICH5 performs I/O cycles as defined in the *Low Pin Count Interface Specification, Revision 1.1*. These are 8-bit transfers. If the processor attempts a 16-bit or 32-bit transfer, the ICH5 breaks the cycle up into multiple 8-bit transfers to consecutive I/O addresses.

***Note:*** If the cycle is not claimed by any peripheral (and subsequently aborted), the ICH5 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.

#### 5.4.1.10 Bus Master Cycles

The ICH5 supports Bus Master cycles and requests (using LDRQ#) as defined in the *Low Pin Count Interface Specification, Revision 1.1*. The ICH5 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:*** The ICH5 does not support LPC Bus Masters performing I/O cycles. LPC Bus Masters should only perform memory read or memory write cycles.

#### 5.4.1.11 LPC Power Management

##### LPCPD# Protocol

Same timings as for SUS_STAT#. Upon driving SUS_STAT# low, LPC peripherals drive LDRQ# low or tri-state it. ICH5 shuts off the LDRQ# input buffers. After driving SUS_STAT# active, the ICH5 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. The ICH5 asserts both SUS_STAT# (connects to LPCPD#) and PCIRST# (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.

#### 5.4.1.12 Configuration and Intel® ICH5 Implications

##### LPC I/F Decoders

To allow the I/O cycles and memory mapped cycles to go to the LPC interface, the ICH5 includes several decoders. During configuration, the ICH5 must be programmed with the same decode ranges as the peripheral. The decoders are programmed via the Device 31:Function 0 configuration space.

***Note:*** The ICH5 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

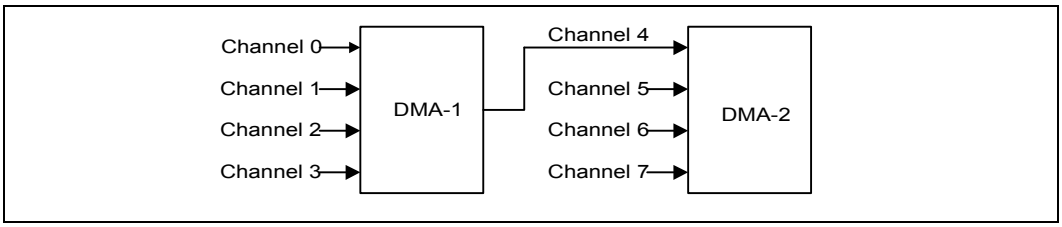
Bus Masters must have a unique START field. In the case of the ICH5 that supports 2 LPC bus masters, it drives 0010 for the START field for grants to bus master #0 (requested via LDRQ0#) and 0011 for grants to bus master #1 (requested via LDRQ1#.). Thus, no registers are needed to configure the START fields for a particular bus master.

## 5.5 DMA Operation (D31:F0)

The ICH5 supports two types of DMA: LPC, and PC/PCI. DMA via LPC is similar to ISA DMA. LPC DMA and PC/PCI DMA use the ICH5'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 individual channels for use by LPC or PC/PCI DMA.

The DMA circuitry incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels (Figure 13). 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 13. Intel® ICH5 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.

ICH5 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.

## 5.5.1 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 the Section 9.2.

### 5.5.1.1 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.

### 5.5.1.2 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 always 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.

## 5.5.2 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.

## 5.5.3 Summary of DMA Transfer Sizes

Table 39 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.

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

**Table 39. 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 |

The ICH5 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 that 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 40.

**Table 40. 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.

## 5.5.4 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.

### 5.5.5 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.

#### 5.5.5.1 Clear Byte Pointer Flip-Flop

This command is executed prior to writing or reading new address or word count information to/ from the DMA controller. This initializes the flip-flop to a known state so that subsequent accesses to register contents by the microprocessor will address upper and lower bytes in the correct sequence.

When the host processor is reading or writing DMA registers, two Byte Pointer flip-flops are used; one for channels 0–3 and one for channels 4–7. Both of these act independently. There are separate software commands for clearing each of them (0Ch for channels 0–3, 0D8h for channels 4–7).

#### 5.5.5.2 DMA Master Clear

This software instruction has the same effect as the hardware reset. The Command, Status, Request, and Internal First/Last Flip-Flop registers are cleared and the Mask register is set. The DMA controller enters the idle cycle.

There are two independent master clear commands; 0Dh that acts on channels 0–3, and 0DAh that acts on channels 4–7.

#### 5.5.5.3 Clear Mask Register

This command clears the mask bits of all four channels, enabling them to accept DMA requests. I/O port 00Eh is used for channels 0–3 and I/O port 0DCh is used for channels 4–7.

## 5.6 PCI DMA

ICH5 provides support for the PC/PCI DMA protocol. PC/PCI DMA uses dedicated REQUEST and GRANT signals to permit PCI devices to request transfers associated with specific DMA channels. Upon receiving a request and getting control of the PCI bus, ICH5 performs a two-cycle transfer. For example, if data is to be moved from the peripheral to main memory, ICH5 first reads data from the peripheral and then writes it to main memory. The location in main memory is the Current Address Registers in the 8237.

ICH5 supports up to two PC/PCI REQ/GNT pairs, REQ[A:B]# and GNT[A:B]#. A 16-bit register is included in the ICH5 Function 0 configuration space at offset 90h. It is divided into seven 2-bit fields that are used to configure the seven DMA channels. Each DMA channel can be configured to one of two options:

- LPC DMA
- PC/PCI style DMA using the REQ/GNT signals

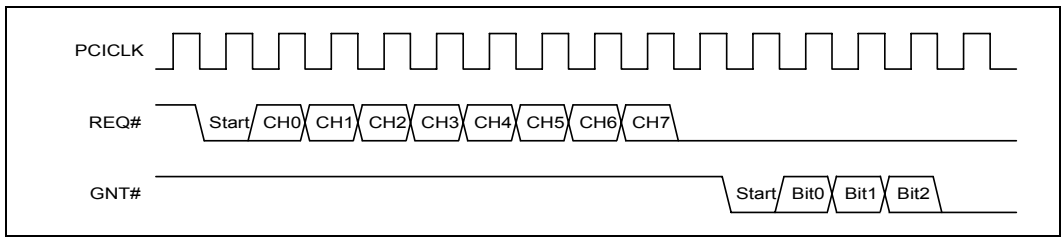
It is not possible for a particular DMA channel to be configured for more than one style of DMA; however, the seven channels can be programmed independently. For example, channel 3 could be set up for PC/PCI and channel 5 set up for LPC DMA.

The ICH5 REQ[A:B]# and GNT[A:B]# can be configured for support of a PC/PCI DMA Expansion agent. The PCI DMA Expansion agent can then provide DMA service or ISA Bus Master service using the ICH5 DMA controller. The REQ#/GNT# pair must follow the PC/PCI serial protocol described below.

### 5.6.1 PCI DMA Expansion Protocol

The PCI expansion agent must support the PCI expansion Channel Passing Protocol defined in Figure 14 for both the REQ# and GNT# pins.

**Figure 14. DMA Serial Channel Passing Protocol**




The requesting device must encode the channel request information as shown above, where CH0–CH7 are one clock active high states representing DMA channel requests 0–7.

ICH5 encodes the granted channel on the GNT# line as shown above, where the bits have the same meaning as shown in Figure 14. For example, the sequence [start, bit 0, bit 1, bit 2=0,1,0,0] grants DMA channel 1 to the requesting device, and the sequence [start, bit 0, bit 1, bit 2=0,0,1,1] grants DMA channel 6 to the requesting device.

All PCI DMA expansion agents must use the channel passing protocol described above. They must also work as follows:

- If a PCI DMA expansion agent has more than one request active, it must resend the request serial protocol after one of the requests has been granted the bus and it has completed its transfer. The expansion device should drive its REQ# inactive for two clocks and then transmit the serial channel passing protocol again, even if there are no new requests from the PCI expansion agent to ICH5. For example: If a PCI expansion agent had active requests for DMA channel 1 and channel 5, it would pass this information to ICH5 through the expansion channel passing protocol. If after receiving GNT# (assume for CH5) and having the device finish its transfer (device stops driving request to PCI expansion agent) it would then need to re-transmit the expansion channel passing protocol to inform ICH5 that DMA channel 1 was still requesting the bus, even if that was the only request the expansion device had pending.
- If a PCI DMA expansion agent has a request go inactive before ICH5 asserts GNT#, it must resend the expansion channel passing protocol to update ICH5 with this new request information. For example: If a PCI expansion agent has DMA channel 1 and 2 requests pending it sends them serially to ICH5 using the expansion channel passing protocol. If, however, DMA channel 1 goes inactive into the expansion agent before the expansion agent receives a GNT# from ICH5, the expansion agent MUST pull its REQ# line high for **one** clock and resend the expansion channel passing information with only DMA channel 2 active. Note that ICH5 does not do anything special to catch this case because a DREQ going inactive before a DACK# is received is not allowed in the ISA DMA protocol and, therefore, does not need to work properly in this protocol either. This requirement is needed to be able to support Plug-n-Play ISA devices that toggle DREQ# lines to determine if those lines are free in the system.
- If a PCI expansion agent has sent its serial request information and receives a new DMA request before receiving GNT# the agent must resend the serial request with the new request active. For example: If a PCI expansion agent has already passed requests for DMA channel 1 and 2 and sees DREQ 3 active before a GNT is received, the device must pull its REQ# line high for one clock and resend the expansion channel passing information with all three channels active.

The three cases above require the following functionality in the PCI DMA expansion device:

- Drive REQ# inactive for one clock to signal new request information.
- Drive REQ# inactive for two clocks to signal that a request that had been granted the bus has gone inactive.
- The REQ# and GNT# state machines must run independently and concurrently (i.e., a GNT# could be received while in the middle of sending a serial REQ# or a GNT# could be active while REQ# is inactive).

## 5.6.2 PCI DMA Expansion Cycles

ICH5's support of the PC/PCI DMA Protocol currently consists of four types of cycles: Memory-to-I/O, I/O-to-Memory, Verify, and ISA Master cycles. ISA Masters are supported through the use of a DMA channel that has been programmed for cascade mode.

The DMA controller does a two cycle transfer (a load followed by a store) as opposed to the ISA "fly-by" cycle for PC/PCI DMA agents. The memory portion of the cycle generates a PCI memory read or memory write bus cycle, its address representing the selected memory.

The I/O portion of the DMA cycle generates a PCI I/O cycle to one of four I/O addresses (Table 41). Note that these cycles must be qualified by an active GNT# signal to the requesting device.

**Table 41. DMA Cycle vs. I/O Address**

| DMA Cycle Type | DMA I/O Address | PCI Cycle Type |
|---|---|---|
| Normal | 00h | I/O Read/Write |
| Normal TC | 04h | I/O Read/Write |
| Verify | 0C0h | I/O Read |
| Verify TC | 0C4h | I/O Read |

## 5.6.3 DMA Addresses

The memory portion of the cycle generates a PCI memory read or memory write bus cycle, its address representing the selected memory. The I/O portion of the DMA cycle generates a PCI I/O cycle to one of the four I/O addresses listed in Table 41.

## 5.6.4 DMA Data Generation

The data generated by PC/PCI devices on I/O reads when they have an active GNT# is on the lower two bytes of the PCI AD bus. Table 42 lists the PCI pins that the data appears on for 8- and 16-bit channels. Each I/O read results in one memory write, and each memory read results in one I/O write. If the I/O device is 8 bit, the ICH5 performs an 8-bit memory write. The ICH5 does not assemble the I/O read into a DWord for writing to memory. Similarly, the ICH5 does not disassemble a DWord read from memory to the I/O device.

**Table 42. PCI Data Bus vs. DMA I/O Port Size**

| PCI DMA I/O Port Size | PCI Data Bus Connection |
|---|---|
| Byte | AD[7:0] |
| Word | AD[15:0] |

## 5.6.5 DMA Byte Enable Generation

The byte enables generated by the ICH5 on I/O reads and writes must correspond to the size of the I/O device. Table 43 defines the byte enables asserted for 8- and 16-bit DMA cycles.

**Table 43. DMA I/O Cycle Width vs. BE[3:0]#**

| BE[3:0]# | Description |
|---|---|
| 1110b | 8-bit DMA I/O Cycle: Channels 0–3 |
| 1100b | 16-bit DMA I/O Cycle: Channels 5–7 |

**NOTE:** For verify cycles the value of the Byte Enables (BEs) is a “don’t care.”

## 5.6.6 DMA Cycle Termination

DMA cycles are terminated when a terminal count is reached in the DMA controller and the channel is not in autoinitialize mode, or when the PC/PCI device deasserts its request. The PC/PCI device must follow explicit rules when deasserting its request, or the ICH5 may not see it in time and run an extra I/O and memory cycle.

The PC/PCI device must deassert its request 7 PCICLKs before it generates TRDY# on the I/O read or write cycle, or the ICH5 is allowed to generate another DMA cycle. For transfers to memory, this means that the memory portion of the cycle will be run without an asserted PC/PCI REQ#.

## 5.6.7 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.

## 5.6.8 Asserting DMA Requests

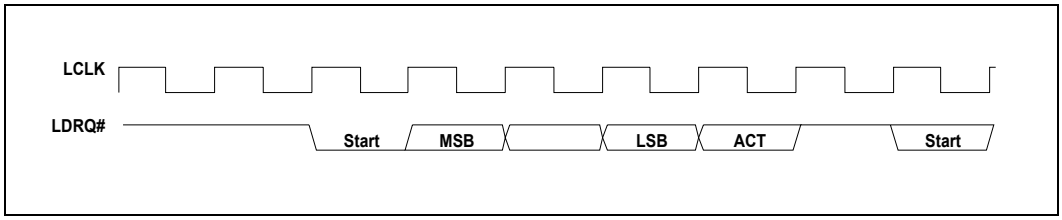
Peripherals that need DMA service encode their requested channel number on the LDRQ# signal. To simplify the protocol, each peripheral on the LPC I/F has its own dedicated LDRQ# signal (they may not be shared between two separate peripherals). The ICH5 has two LDRQ# inputs, allowing at least two devices to support DMA or bus mastering.

LDRQ# is synchronous with LCLK (PCI clock). As shown in Figure 15, 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 15. DMA Request Assertion through LDRQ#**



## 5.6.9 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 the ICH5, there is no guarantee 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 the ICH5 and the peripheral.

## 5.6.10 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. ICH5 starts transfer by asserting 0000b on LAD[3:0] with LFRAME# asserted.
2. ICH5 asserts 'cycle type' of DMA, direction based on DMA transfer direction.
3. ICH5 asserts channel number and, if applicable, terminal count.
4. ICH5 indicates the size of the transfer: 8 or 16 bits.
5. If a DMA read…
   - The ICH5 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…
   - The ICH5 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.

## 5.6.11 Terminal Count

Terminal count is communicated through LAD3 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.

## 5.6.12 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.

## 5.6.13 DMA Request Deassertion

An end of transfer is communicated to the ICH5 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) the ICH5 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 the ICH5 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 the ICH5 that this is the last piece of data transferred on a DMA read (ICH5 to peripheral), or the byte that follows is the last piece of data transferred on a DMA write (peripheral to ICH5).

When the ICH5 sees one of these two encodings, it ends the DMA transfer after this byte and deasserts the DMA request to the 8237. Therefore, if the ICH5 indicated a 16 bit transfer, the peripheral can end the transfer after one byte by indicating a SYNC value of 0000b or 1010b. The ICH5 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 the ICH5 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 the ICH5 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 the ICH5, the data will be transferred and the DMA request will remain active to the 8237. At a later time, the ICH5 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 the ICH5 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 guaranteed that they will receive the next START indication from the ICH5.

***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.

## 5.6.14 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.

# 5.7 8254 Timers (D31:F0)

The ICH5 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. The counter negates refresh request for 1 counter period (838 ns) during each count cycle. The initial count value is loaded 1 counter period after being written to the counter I/O address. The counter initially asserts refresh request, and negates it for 1 counter period when the count value reaches 1. The counter then asserts refresh request and continues counting from the initial count value.

### 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).

## 5.7.1 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 44 lists the six operating modes for the interval counters.

**Table 44. Counter Operating Modes**

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

## 5.7.2 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.

#### 5.7.2.1 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.

#### 5.7.2.2 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.

#### 5.7.2.3 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.

# 5.8 8259 Interrupt Controllers (PIC) (D31:F0)

The ICH5 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 45 shows how the cores are connected.

**Table 45. 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 |
| | 4 | Serial Port B | IRQ4 via SERIRQ |
| | 5 | Parallel Port / Generic | IRQ5 via SERIRQ |
| | 6 | Floppy Disk | IRQ6 via SERIRQ |
| | 7 | Parallel Port / Generic | IRQ7 via SERIRQ |
| Slave | 0 | Internal Real Time Clock | Internal RTC / HPET #1 |
| | 1 | Generic | IRQ9 via SERIRQ |
| | 2 | Generic | IRQ10 via SERIRQ |
| | 3 | Generic | IRQ11 via SERIRQ |
| | 4 | PS/2 Mouse | IRQ12 via SERIRQ |
| | 5 | Internal | State Machine output based on processor FERR# assertion. |
| | 6 | Primary IDE cable | IRQ14 from input signal (primary IDE in legacy mode only) or via SERIRQ |
| | 7 | Secondary IDE Cable | IRQ15 from input signal (secondary IDE in legacy mode only) or via SERIRQ |

The ICH5 cascades the slave controller onto the master controller through master controller interrupt input 2. This means there are only 15 possible interrupts for the ICH5 PIC.

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

Note that previous PIIXn devices internally latched IRQ12 and IRQ1 and required a port 60h read to clear the latch. The ICH5 can be programmed to latch IRQ12 or IRQ1 (see bit 11 and bit 12 in General Control Register, D31:F0, offset D0h).

***Note:*** Active-low interrupt sources (e.g., the PIRQ#s) are inverted inside the ICH5. 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#.

## 5.8.1 Interrupt Handling

### 5.8.1.1 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 46 defines the IRR, ISR, and IMR.

**Table 46. 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. |

### 5.8.1.2 Acknowledging Interrupts

The processor generates an interrupt acknowledge cycle that is translated by the host bridge into a PCI Interrupt Acknowledge Cycle to the ICH5. 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 47. 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 |

#### 5.8.1.3 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 the ICH5.
4. Upon observing its own interrupt acknowledge cycle on PCI, the ICH5 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.

### 5.8.2 Initialization Command Words (ICWx)

Before operation can begin, each 8259 must be initialized. In the ICH5, 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.

#### 5.8.2.1 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, the ICH5 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.

#### 5.8.2.2 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.

#### 5.8.2.3 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 the ICH5, 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.

#### 5.8.2.4 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.

### 5.8.3 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.

### 5.8.4 Modes of Operation

#### 5.8.4.1 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.

### 5.8.4.2 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.

### 5.8.4.3 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).

### 5.8.4.4 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.

### 5.8.4.5 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.

### 5.8.4.6 Cascade Mode

The PIC in the ICH5 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 the ICH5, 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.

### 5.8.4.7 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 the ICH5, 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.

### 5.8.4.8 End of Interrupt 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.

### 5.8.4.9 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 the ICH5, 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.

### 5.8.4.10 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.

## 5.8.5 Masking Interrupts

### 5.8.5.1 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.

### 5.8.5.2 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.

## 5.8.6 Steering PCI Interrupts

The ICH5 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 PIRQx Route Control registers, located at 60–63h and 68–6Bh in 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. The ICH5 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 ISA device (through SERIRQ). However, active low non-ISA 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. The ICH5 receives the PIRQ input, like all of the other external sources, and routes it accordingly.

# 5.9 Advanced Interrupt Controller (APIC) (D31:F0)

In addition to the standard ISA-compatible PIC described in the previous chapter, the ICH5 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.

## 5.9.1 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 the ICH5 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.

## 5.9.2 Interrupt Mapping

The I/O APIC within the ICH5 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 48. APIC Interrupt Mapping (Sheet 1 of 2)**

| IRQ # | 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 |
| 12 | Yes | No | Yes | |
| 13 | No | No | No | FERR# logic |
| 14 | Yes | Yes[1] | Yes | Storage (IDE/SATA) Primary (legacy mode) |
| 15 | Yes | Yes[1] | Yes | Storage (IDE/SATA) Secondary (legacy mode) |

**Table 48. APIC Interrupt Mapping (Sheet 2 of 2)**

| IRQ # | Via SERIRQ | Direct from Pin | Via PCI Message | Internal Modules |
|---|---|---|---|---|
| 16 | PIRQA# | PIRQA# | No[4] | USB UHCI Controller #1, USB UHCI Controller #4 |
| 17 | PIRQB# | PIRQB# | No[4] | AC '97 Audio, Modem, option for SMbus |
| 18 | PIRQC# | PIRQC# | No[4] | USB UHCI Controller #3, Storage (IDE/SATA) Native mode |
| 19 | PIRQD# | PIRQD# | No[4] | USB UHCI Controller #2 |
| 20 | N/A | PIRQE# | No[4] | LAN, option for SCI, TCO, HPET #0,1,2 |
| 21 | N/A | PIRQF# | Yes | Option for SCI, TCO, HPET #0,1,2 |
| 22 | N/A | PIRQG# | Yes | Option for SCI, TCO, HPET #0,1,2 |
| 23 | N/A | PIRQH# | No[4] | USB EHCI Controller, option for SCI, TCO, HPET #0,1,2 |

**NOTES:**
1. IRQ 14 and 15 can only be driven directly from the pins when in legacy IDE mode.
2. 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.
3. If IRQ 11 is used for HPET #2, software should ensure IRQ 11 is not shared with any other devices to guarantee the proper operation of HPET #2. ICH5 hardware does not prevent sharing of IRQ 11.
4. PCI Message interrupts are not prevented by hardware in these cases. However, the system must not program these interrupts as edge-triggered (as required for PCI message interrupts) because the internal and external PIRQs on these inputs must be programmed in level-triggered modes.

## 5.9.3 PCI Message-Based Interrupts

The following scheme is only supported when the internal I/O(x) APIC is used (rather than just the 8259). The ICH5 supports the new method for PCI devices to deliver interrupts as write cycles, rather than using the traditional PIRQ[A:D] signals. Essentially, the PCI devices are given a write path directly to a register that will cause the desired interrupt. This mode is only supported when the ICH5's internal I/O APIC is enabled.

The interrupts associated with the PCI Message-based interrupt method must be set up for edge triggered mode, rather than level triggered, since the peripheral only does the write to indicate the edge.

The following sequence is used:

1. During PCI PnP, the PCI peripheral is first programmed with an address (MESSAGE_ADDRESS) and data value (MESSAGE_DATA) that will be used for the interrupt message delivery. For the ICH5, the MESSAGE_ADDRESS is the IRQ Pin Assertion Register, which is mapped to memory location: FEC0_0020h.
2. To cause the interrupt, the PCI peripheral requests the PCI bus and when granted, writes the MESSAGE_DATA value to the location indicated by the MESSAGE_ADDRESS. The MESSAGE_DATA value indicates which interrupt occurred. This MESSAGE_DATA value is a binary encoded. For example, to indicate that interrupt 7 should go active, the peripheral will write a binary value of 0000111. The MESSAGE_DATA is a 32-bit value, although only the lower 5 bits are used.
3. If the PRQ bit in the APIC Version Register is set, the ICH5 positively decodes the cycles (as a slave) in Medium time.
4. The ICH5 decodes the binary value written to MESSAGE_ADDRESS and sets the appropriate IRR bit in the internal I/O APIC. The corresponding interrupt must be set up for edge-triggered interrupts. The ICH5 supports interrupts 00h through 23h. Binary values outside this range do not cause any action.
5. After sending the interrupt message to the processor, the ICH5 automatically clears the interrupt.

Because they are edge triggered, the interrupts that are allocated to the PCI bus for this scheme may not be shared with any other interrupt (such as the standard PCI PIRQ[A:D], those received via SERIRQ#, or the internal level-triggered interrupts such as SCI or TCO).

The ICH5 ignores interrupt messages sent by PCI masters that attempt to use IRQ0, 2, 8, or 13.

#### 5.9.3.1 Registers and Bits Associated with PCI Interrupt Delivery

##### Capabilities Indication

The capability to support PCI interrupt delivery are indicated via ACPI configuration techniques. This involves the BIOS creating a data structure that gets reported to the ACPI configuration software. The OS reads the PRQ bit in the APIC Version Register to see if the ICH5 is capable of support PCI-based interrupt messages. As a precaution, the PRQ bit is not set if the XAPIC_EN bit is not set.

##### Interrupt Message Register

The PCI devices all write their message into the IRQ Pin Assertion Register, which is a memory-Mapped register located at the APIC base memory location + 20h.

## 5.9.4 Front Side Bus Interrupt Delivery

For processors that support Front Side Bus (FSB) interrupt delivery, the ICH5 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 the ICH5 writing (via the hub interface) 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 the ICH5 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, the ICH5 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. The ICH5 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 5.9.4.4.

***Note:*** FSB Interrupt Delivery compatibility with processor clock control depends on the processor, not the ICH5.

### 5.9.4.1 Edge-Triggered Operation

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

### 5.9.4.2 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.

### 5.9.4.3 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.

### 5.9.4.4 Interrupt Message Format

The ICH5 writes the message to PCI (and to the Host controller) as a 32-bit memory write cycle. It uses the formats shown in Table 49 and Table 50 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 the ICH5 has any way to have a SMI source from ICH5 power management logic cause the I/O APIC to send an SMI message (there is no way to do this). The ICH5’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 ICH5.

### Table 49. 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 50. 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.<br>If using edge-triggered interrupts, then bit is always 1, since only the assertion is sent.<br>If using level-triggered interrupts, then this bit indicates the state of the interrupt input. |
| 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. |

## 5.10 Serial Interrupt (D31:F0)

The ICH5 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, the ICH5, 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

The ICH5 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).

***Note:*** When the IDE primary and secondary controllers are configured for native IDE mode, the only way to use the internal IRQ14 and IRQ15 connections to the Interrupt controllers is through the Serial Interrupt pin.

### 5.10.1 Start Frame

The serial IRQ protocol has two modes of operation which affect the start frame. These two modes are: Continuous, where the ICH5 is solely responsible for generating the start frame; and Quiet, where a serial IRQ peripheral is responsible for beginning the start frame.

The mode that must first be entered when enabling the serial IRQ protocol is continuous mode. In this mode, the ICH5 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. The ICH5 senses the line low and continues to drive it low for the remainder of the Start Frame. Since the first PCI clock of the start frame was driven by the peripheral in this mode, the ICH5 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.

## 5.10.2 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

## 5.10.3 Stop Frame

After all data frames, a Stop Frame is driven by the ICH5. The SERIRQ signal is driven low by the ICH5 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 51. 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 (Intel® ICH5) may initiate a Start Frame |

## 5.10.4 Specific Interrupts Not Supported via SERIRQ

There are three interrupts seen through the serial stream that are not supported by the ICH5. 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#.

The ICH5 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. In addition, the interrupts IRQ14 and IRQ15 from the serial stream are treated differently than their ISA counterparts. These two frames are not passed to the Bus Master IDE logic. The Bus Master IDE logic expects IDE to be behind the ICH5.

### 5.10.5 Data Frame Format

Table 52 shows the format of the data frames. For the PCI interrupts (A–D), the output from the ICH5 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 52. 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 or on ISA. |
| 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 | Do not include in BM IDE interrupt logic |
| 16 | IRQ15 | 47 | Do not include in BM IDE interrupt 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# |

## 5.11 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 optional. The hour is represented in twelve or twenty-four hour format, and data can be represented in BCD or binary format. The design is meant to be 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–FF 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.

The ICH5 does not implement month/year alarms.

## 5.11.1 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.

## 5.11.2 Interrupts

The real-time clock interrupt is internally routed within the ICH5 both to the I/O APIC and the 8259. It is mapped to interrupt vector 8. This interrupt does not leave the ICH5, nor is it shared with any other interrupt. IRQ8# from the SERIRQ stream is ignored.

## 5.11.3 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 (may be all 0s or all 1s).

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.

## 5.11.4 Century Rollover

The ICH5 detects a rollover when the Year byte (RTC I/O space, index offset 09h) transitions form 99 to 00. Upon detecting the rollover, the ICH5 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, the ICH5 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.

## 5.11.5 Clearing Battery-Backed RTC RAM

Clearing CMOS RAM in an ICH5-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.

### 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 53 shows which bits are set to their default state when RTCRST# is asserted.

### Table 53. Configuration Bits Reset by RTCRST# Assertion

| Bit Name | Default State | Register | Location | Bit(s) |
|---|---|---|---|---|
| FREQ_STRAP[3:0] | GEN_STS | D31:F0:D4h | 11:8 | 1111b |
| AIE | RTC Reg B | I/O space | 5 | 0 |
| AF | RTC Reg C | I/O space | 5 | 0 |
| PWR_FLR | GEN_PMCON_3 | D31:F0:A4h | 1 | 0 |
| AFTERG3_EN | GEN_PMCON_3 | D31:F0:A4h | 0 | 0 |
| RTC_PWR_STS | GEN_PMCON_3 | D31:F0:A4h | 2 | 1 |
| PRBTNOR_STS | PM1_STS | PMBase + 00h | 11 | 0 |
| PME_EN | GPE0_EN | PMBase + 2Ah | 11 | 0 |
| RI_EN | GPE0_EN | PMBase + 2Ah | 8 | 0 |
| NEW_CENTURY_STS | TCO1_STS | TCOBase + 04h | 7 | 0 |
| INTRD_DET | TCO2_STS | TCOBase + 06h | 0 | 0 |
| TOP_SWAP | GEN_STS | D31:F0:D4h | 13 | 0 |
| RTC_EN | PM1_EN | PMBase + 02h | 10 | 0 |
| BATLOW_EN | GPE0_EN | PMBase + 2Ah | 10 | 0 |

### 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.

### Using the SAFEMODE Strap to Clear CMOS

A jumper on AC_SDOUT (SAFEMODE strap) can also be used to clear CMOS values. BIOS would detect the setting of the SAFE_MODE status bit (D31:F0: Offset D4h bit 2) on system boot-up, and manually clear the CMOS array.

***Note:*** Both the GPI and SAFEMODE strap techniques to clear CMOS require 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. The 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.

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

# 5.12 Processor Interface (D31:F0)

The ICH5 interfaces to the processor with a variety of signals

- Standard Outputs to processor: A20M#, SMI#, NMI, INIT#, INTR, STPCLK#, IGNNE#, CPUSLP#, CPUPWRGD
- Standard Input from processor: FERR#

Most ICH5 outputs to the processor use standard buffers. The ICH5 has separate V_CPU_IO signals that are pulled up at the system level to the processor voltage, and thus determines $V_{OH}$ for the outputs to the processor. Note that this is different than previous generations of chips, that have used open-drain outputs. This new method saves up to 12 external pull-up resistors.

The ICH5 also handles the speed setting for the processor by holding specific signals at certain states just prior to CPURST going inactive. This avoids the glue often required with other chipsets.

The ICH5 does not support the processor's FRC mode.

## 5.12.1 Processor Interface Signals

This section describes each of the signals that interface between the ICH5 and the processor(s). Note that the behavior of some signals may vary during processor reset, as the signals are used for frequency strapping.

### 5.12.1.1 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).

### 5.12.1.2 INIT# (Initialization)

The INIT# signal is active (driven low) based on any one of several events described in Table 54. 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.

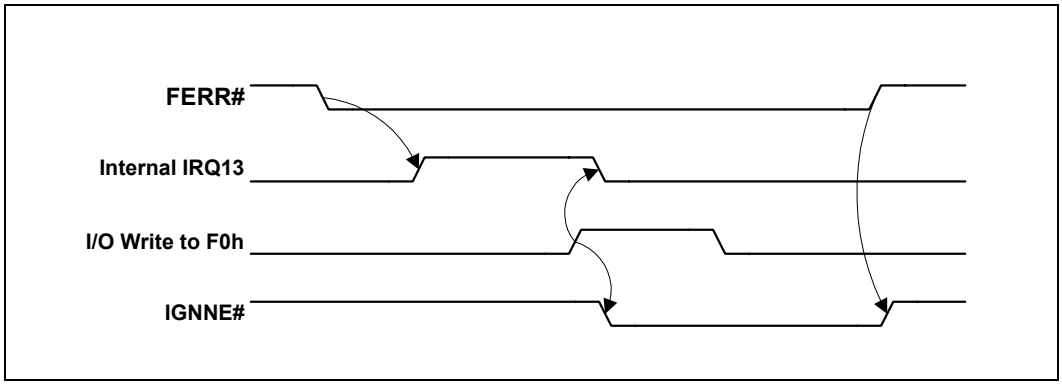
**Table 54. INIT# Going Active**

| 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 the Intel® ICH5 will arm INIT# to be generated again.<br>**NOTE:** RCIN# signal is expected to be low during S3, 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. |

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

The ICH5 supports the coprocessor error function with the FERR#/IGNNE# pins. The function is enabled via the COPROC_ERR_EN bit (Device 31:Function 0, Offset D0, bit 13). 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, the ICH5 negates the internal IRQ13 and drives IGNNE# active. IGNNE# remains active until FERR# is driven inactive. IGNNE# is never driven active unless FERR# is active.

**Figure 16. 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#.

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

Non-Maskable Interrupts (NMIs) can be generated by several sources, as described in Table 55.

**Table 55. NMI Sources**

| Cause of NMI | Comment |
|---|---|
| SERR# goes active (either internally, externally via SERR# signal, or via message from MCH) | Can instead be routed to generate an SCI, through the NMI2SCI_EN bit (Device 31:Function 0, offset 4E, 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, offset 4E, bit 11). |

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

The ICH5 power management logic controls these active-low signals. Refer to Section 5.13 for more information on the functionality of these signals.

#### 5.12.1.6 CPU Power Good (CPUPWRGOOD)

This signal is connected to the processor's PWRGOOD input. This is an open-drain output signal (external pull-up resistor required) that represents a logical AND of the ICH5's PWROK and VRMPWRGD signals.

### 5.12.2 Dual-Processor Issues

#### 5.12.2.1 Signal Differences

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

**Table 56. DP Signal Differences**

| Signal | Difference |
|---|---|
| A20M# / A20GATE | Generally not used, but still supported by Intel® ICH5. |
| 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 ICH5. |

### 5.12.2.2 Power Management

Attempting clock control with more than one processor is not feasible, because the Host controller does not provide any indication as to which processor is executing a particular Stop-Grant cycle. Without this information, there is no way for the ICH5 to know when it is safe to deassert STPCLK#.

Because the S1 state will have the STPCLK# signal active, the STPCLK# signal can be connected to both processors. The BIOS must indicate that the ICH5 only supports the C1 state for dual-processor designs. However, the THRM# signal can be used for overheat conditions to activate thermal throttling.

When entering S1, the ICH5 asserts STPCLK# to both processors. To meet the processor specifications, the CPUSLP# signal will have to be delayed until the second Stop-Grant cycle occurs. To ensure this, the ICH5 waits a minimum or 60 PCI clocks after receipt of the first Stop-Grant cycle before asserting CPUSLP# (if the SLP_EN bit is set to 1).

Both processors must immediately respond to the STPCLK# assertion with stop grant acknowledge cycles before the ICH5 asserts CPUSLP# in order to meet the processor setup time for CPUSLP#. Meeting the processor setup time for CPUSLP# is not an issue if both processors are idle when the system is entering S1. If you cannot guarantee that both processors will be idle, do not enable the SLP_EN bit. Note that setting SLP_EN to 1 is not required to support S1 in a dual-processor configuration.

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.

## 5.12.3 Speed Strapping for Processor

The ICH5 directly sets the speed straps for the processor, saving the external logic that has been needed with prior PCIsets. Refer to processor specification for speed strapping definition.

The ICH5 performs the following to set the speed straps for the processor:

1. While PCIRST# is active, the ICH5 drives A20M#, IGNNE#, NMI, and INTR high.
2. As soon as PWROK goes active, the ICH5 reads the FREQ_STRAP field contents.
3. The next step depends on the power state being exited as described in Table 57.

**Table 57. Frequency Strap Behavior Based on Exit State**

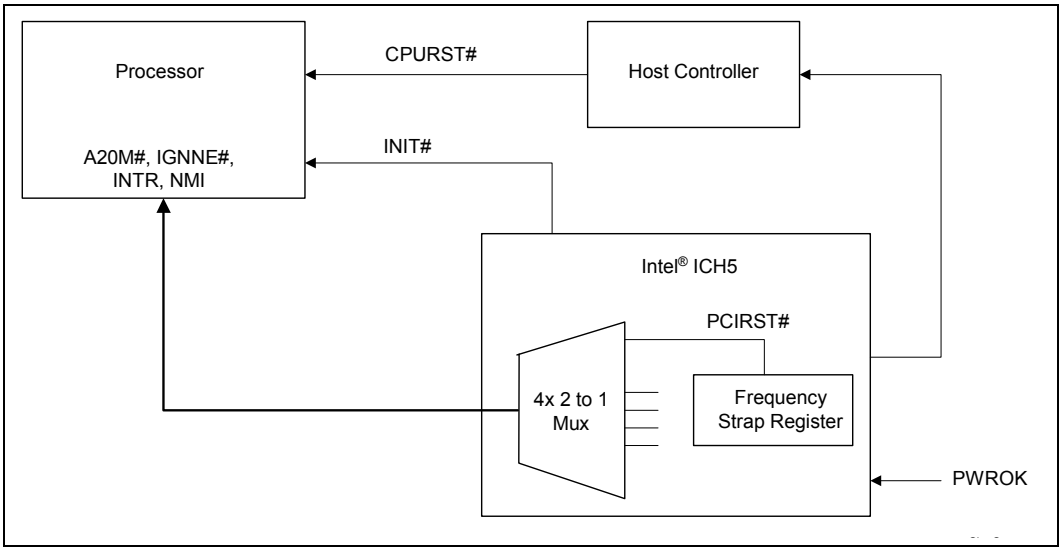
| State Exiting | Intel® ICH5 |
|---|---|
| S1 | There is no processor reset, so no frequency strap logic is used. |
| S3, S4, S5, or G3 | Based on PWROK going active, the Intel® ICH5 deasserts PCIRST#, and based on the value of the FREQ_STRAP field (D31:F0,Offset D4), the ICH5 drives the intended core frequency values on A20M#, IGNNE#, NMI, and INTR. |

### Table 58. Frequency Strap Bit Mapping

| FREQ_STRAP Bits [3:0] | Sets High/Low Level for the Corresponding Signal |
|---|---|
| 3 | NMI |
| 2 | INTR |
| 1 | IGNNE# |
| 0 | A20M# |

**NOTE:** The FREQ_STRAP register is in the RTC well. The value in the register can be forced to 1111h via a pinstrap (AC_SDOUT signal), or the ICH5 can automatically force the speed strapping to 1111h if the processor fails to boot.

### Figure 17. Signal Strapping



# 5.13 Power Management (D31:F0)

## 5.13.1 Features

- ACPI Power and Thermal Management Support
  - — 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 Sleeping State Control
  - — ACPI S1 state: Stop Grant or Quickstart state (using STPCLK# signal) halts processor's instruction stream (only STPCLK# active, and SLP# 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 Support for APM-Based Systems

## 5.13.2 Intel® ICH5 and System Power States

Table 59 shows the power states defined for ICH5-based platforms. The state names generally match the corresponding ACPI states.

**Table 59. General Power States for Systems Using Intel® ICH5**

| 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 60. Within the C0 state, the Intel® ICH5 can throttle the STPCLK# signal to reduce power consumption. The throttling can be initiated by software or by the THRM# input signal. |
| G0/S0/C1 | **Auto-Halt:** Processor has executed a AutoHalt instruction and is not executing code. The processor snoops the bus and maintains cache coherency. |
| G1/S1 | **Stop-Grant:** ICH5 also has the option to assert the CPUSLP# signal to further reduce processor power consumption.<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 66 for more details. |

Table 60 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 states. These intermediate transitions and states are not listed in the table.

### Table 60. State Transition Rules for Intel® ICH5

| Present State | Transition Trigger | Next State |
|---|---|---|
| G0/S0/C0 | • Processor halt instruction<br>• SLP_EN bit set<br>• Power Button Override<br>• Mechanical Off/Power Failure | • G0/S0/C1<br>• G0/S0<br>• G0/S0<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<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<br>• G2/S5<br>• G3 |
| G2/S5 | • Any Enabled Wake Event<br>• Power Failure | • G0/S0/C0<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) |

**NOTES:**
1. Some wake events can be preserved through power failure.

### 5.13.3 System Power Planes

The system has several independent power planes, as described in Table 61. Note that when a particular power plane is shut off, it should go to a 0 V level.

**Table 61. 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. |
| MAIN | SLP_S3# signal | When SLP_S3# goes active, power can be shut off to any circuit not required to wake the system from the S3 state. Since the S3 state requires that the memory context be preserved, power must be retained to the main memory.<br>The processor, devices on the PCI bus, LPC I/F downstream hub interface and AGP will typically be shut off when the Main power plane is shut, although there may be small subsections powered. |
| MEMORY | SLP_S4# 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. |
| 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. |

### 5.13.4 Intel® ICH5 Power Planes

The ICH5 power planes are previously defined in Section 3.1. Although not specific power planes within the ICH5, there are many interface signals that go to devices that may be powered down. These include:

- IDE: ICH5 can tri-state or drive low all IDE output signals and shut off input buffers.
- USB: ICH5 can tri-state USB output signals and shut off input buffers if USB wakeup is not desired
- AC ’97: ICH5 can drive low the outputs and shut off inputs

### 5.13.5 SMI#/SCI Generation

On any SMI# event taking place, ICH5 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 9.1.11 ACPI Control Register for details.) The interrupt remains asserted until all SCI sources are removed.

Table 62 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 62. Causes of SMI# and SCI (Sheet 1 of 2)**

| Cause | 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 |
| Power Button Press | Yes | Yes | PWRBTN_EN=1 | PWRBTN_STS |
| RTC Alarm | Yes | Yes | RTC_EN=1 | RTC_STS |
| Ring Indicate | Yes | Yes | RI_EN=1 | RI_STS |
| AC '97 wakes | Yes | Yes | AC97_EN=1 | AC97_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 | 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 MCH | 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 MCH | 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 |
| 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 | none | 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 |

**Table 62. Causes of SMI# and SCI (Sheet 2 of 2)**

| Cause | SCI | SMI | Additional Enables | Where Reported |
|---|---|---|---|---|
| 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 | DEV[n]_TRAP_EN=1 | DEVMON_STS, DEV[n]_TRAP_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 |
| 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.

## 5.13.6 Dynamic Processor Clock Control

The ICH5 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.

The ICH5 supports the ACPI C0 and C1 states.

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

- STPCLK#: Used to halt processor instruction stream.

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

A C1 state ends due to a Break event. Based on the break event, the ICH5 returns the system to C0 state.

#### 5.13.6.1 Throttling Using STPCLK#

Throttling is used to lower power consumption or reduce heat. The ICH5 asserts STPCLK# to throttle the processor clock. After a programmable time, the ICH5 deasserts STPCLK# and the processor appears to return to the C0 state. This allows the processor to operate at reduced average power, with a corresponding decrease in performance. Two methods are included to start throttling:

1. Software enables a timer with a programmable duty cycle. The duty cycle is set by the THTL_DTY field and the throttling is enabled using the THTL_EN field. This is known as Manual Throttling. The period is fixed to be in the non-audible range, due to the nature of switching power supplies.
2. A Thermal Override condition (THRM# signal active for >2 seconds) occurs that unconditionally forces throttling, independent of the THTL_EN bit. The throttling due to Thermal Override has a separate duty cycle (THRM_DTY) which may vary by field and system. The Thermal Override condition will end when THRM# goes inactive.

Throttling due to the THRM# signal has higher priority than the software initiated throttling.

#### 5.13.6.2 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 sleep state (S1–S5), the THTL_EN bit can be internally treated as being disabled (no throttling while going to sleep state). Note that thermal throttling (based on THRM# signal) cannot be disabled in an S0 state. However, once the SLP_EN bit is set, the thermal throttling is shut off (since STPCLK# will be active in S1–S5 states).
- Level 2 C2 Level 2 Level 2 C2 Level 2 C2 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 the ICH5 observing the Stop-Grant cycle. This ensures that the STPCLK# signals stays active for a sufficient period after the processor observes the response phase.

### 5.13.7 Sleep States

#### 5.13.7.1 Sleep State Overview

The ICH5 directly supports different sleep states (S1–S5), which 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 always 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.

### 5.13.7.2 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 63. Sleep Types**

| Sleep Type | Comment |
|---|---|
| S1 | Intel® ICH5 asserts the STPCLK# signal. It also has the option to assert CPUSLP# signal. This lowers the processor's power consumption. No snooping is possible in this state. |
| S3 | ICH5 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 | ICH5 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. ICH5 asserts SLP_S3#, SLP_S4# and SLP_S5#. |

### 5.13.7.3 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 the ICH5-controlled Sleep states, the WAK_STS bit is set. The possible causes of Wake Events (and their restrictions) are shown in Table 64.

### Table 64. Causes of Wake Events

| Cause | States Can Wake From | How Enabled |
|---|---|---|
| RTC Alarm | S1–S5 (Note 1) | Set RTC_EN bit in PM1_EN register |
| Power Button | S1–S5 | Always enabled as Wake event |
| GPI[0:n] | S1–S5 (Note 1) | GPE0_EN register |
| 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 (Note 1) | Set RI_EN bit in GPE0_EN register |
| AC97 | S1–S5 | Set AC97_EN bit in GPE0_EN register |
| Primary PME# | S1–S5 | PME_B0_EN bit in GPE0_EN register |
| Secondary PME# | S1–S5 (Note 1) | Set PME_EN bit in GPE0_EN register. |
| GST Timeout | S1M | Setting the GST Timeout range to a value other than 00h. |
| SMBALERT# | S1–S5 | Always enabled as Wake event |
| SMBus Slave Message | S1–S5 | Wake/SMI# command always enabled as a Wake event.<br>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. |
| PME_B0 (internal USB2.0 EHCI controller) | S1–S5 (Note 1) | Set PME_B0_EN bit in GPE0_EN register. |

**NOTES:**
1. 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.
2. If in the S5 state due to a powerbutton override, the possible wake events are due to Power Button, Hard Reset Without Cycling (See Command Type 3 in Table 121), and Hard Reset System (See Command Type 4 in Table 121).

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 an S1 state. Also, only certain GPIs are “ACPI Compliant,” meaning that their Status and Enable bits reside in ACPI I/O space. Table 65 summarizes the use of GPIs as wake events.

### Table 65. GPI Wake Events

| GPI | Power Well | Wake From | Notes |
|---|---|---|---|
| GPI[7:0] | Core | S1 | |
| GPI[13:11], GPI8 | 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 the ICH5 are insignificant.

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

Power failures can occur if the AC power is cut (a real power failure) or if the system is unplugged. In either case, PWROK and RSMRST# are assumed to go low.

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 the ICH5 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.

The ICH5 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 66. 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 |

## 5.13.8 Thermal Management

The ICH5 has mechanisms to assist with managing thermal problems in the system.

### 5.13.8.1 THRM# Signal

The THRM# signal is used as a status input for a thermal sensor. Based on the THRM# signal going active, the ICH5 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.

### 5.13.8.2 THRM# Initiated Passive Cooling

If the THRM# signal remains active for some time greater than 2 seconds and the ICH5 is in the S0/G0/C0 state, then the ICH5 enters an auto-throttling mode, in which it provides a duty cycle on the STPCLK# signal. This reduces the overall power consumption by the system, and should cool the system. The intended result of the cooling is that the THRM# signal should go back inactive.

For all programmed values (001–111), THRM# going active 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, the ICH5 waits for the STOP-GRANT cycle before starting the count of the time the STPCLK# signal is active.

When THRM# goes inactive, the throttling stops.

In case that the ICH5 is already attempting throttling because the THTL_EN bit is set, the duty cycle associated with the THRM# signal has higher priority.

If the ICH5 is in the S1–S5 states, then no throttling will be caused by the THRM# signal being active.

### 5.13.8.3 THRM# Override Software Bit

The FORCE_THTL bit allows the BIOS to force passive cooling, just as if the THRM# signal had been active for 2 seconds. If this bit is set, the ICH5 starts throttling using the ratio in the THRM_DTY field.

When this bit is cleared the ICH5 stops throttling, unless the THRM# signal has been active for 2 seconds or if the THTL_EN bit is set (indicating that ACPI software is attempting throttling).

#### 5.13.8.4 Processor Initiated Passive Cooling (Via Programmed Duty Cycle on STPCLK#)

Using the THTL_EN and THTL_DTY bits, the ICH5 can force a programmed duty cycle on the STPCLK# signal. This reduces the effective instruction rate of the processor and cuts its power consumption and heat generation.

#### 5.13.8.5 Active Cooling

Active cooling involves fans. The GPIO signals from the ICH5 can be used to turn on/off a fan.

### 5.13.9 Event Input Signals and Their Usage

The ICH5 has various input signals that trigger specific events. This section describes those signals and how they should be used.

#### 5.13.9.1 PWRBTN# (Power Button)

The ICH5 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 67. 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 67. 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). 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.

New: A power button override forces a transition to S5, even if PWROK is not active.

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 the ICH5 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 the ICH5 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.

## 5.13.9.2 RI# (Ring Indicator)

The Ring Indicator can cause a wake event (if enabled) from the S1–S5 states. Table 68 shows when the wake event is generated or ignored in different states. If in the G0/S0/Cx states, the ICH5 generates an interrupt based on RI# active, and the interrupt will be set up as a Break event.

**Table 68. 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 ICH5. Can be in modem or external.

### 5.13.9.3 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.

In the EHCI controller, there is an internal PME_B0 bit. This is separate from the external PME# signal and can cause the same effect.

### 5.13.9.4 SYS_RESET# Signal

SYS_RESET# on the ICH5 is used to eliminate extra glue logic on the board. Before the addition of this pin, a system reset was activated by external glue forcing the PWROK signal low after the reset button was pressed. This pin eliminates the need for that glue.

When the SYS_RESET# pin is detected as active after the 16 ms debounce logic, the ICH5 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 approximately 1 ms regardless of whether the SYSRESET# 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 PCIRST# inactive.

### 5.13.9.5 THRMTRIP# Signal

If THRMTRIP# goes active, the processor is indicating an overheat condition, and the ICH5 immediately transitions to an S5 state. However, since the processor has overheated, it does not respond to the ICH5’s STPCLK# pin with a stop grant special cycle. Therefore, the ICH5 does not wait for one. Immediately upon seeing THRMTRIP# low, the ICH5 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, the ICH5 power down immediately without following the normal S0 -> S5 path. This path may be taken in parallel, but ICH5 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 the ICH5, are no longer executing cycles properly. Therefore, if THRMTRIP# fires, and the ICH5 is relying on state machine logic to perform the power down, the state machine may not be working, and the system will not power down.

The ICH5 follows this flow for THRMTRIP#.

1. At boot (PCIRST# low), THRMTRIP# ignored.
2. After power-up (PCIRST# high), if THRMTRIP# sampled active, SLP_S3#, SLP_S4#, and SLP_S5# fire, 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 the ICH5 never reaches S5, the ICH5 does not reboot until power is cycled.

## 5.13.10 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, the ICH5 implements an ALT access mode.

If the ALT access mode is entered and exited after reading the registers of the ICH5 timer (8254), the timer starts counting faster (13.5 ms). The following steps listed below can cause problems:

1. BIOS enters ALT access mode for reading the ICH5 timer related registers.
2. BIOS exits ALT access mode.
3. BIOS continues through the execution of other needed steps and passes control to the OS.

After getting control in step #3, if the OS 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 OS. 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.

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

The registers described in Table 69 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 69. 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] |

## Table 69. 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** |
| 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,<br>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 |
| 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:<br>Bits(1:0) = 00 | CCh | 2 | 1 | DMA Chan 7 base address low byte |
| | | 4 | DMA Chan 1 Mode:<br>Bits(1:0) = 01 | | | 2 | DMA Chan 7 base address high byte |
| | | 5 | DMA Chan 2 Mode:<br>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[2] |
| | | 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[1] | | | 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. The OCW1 register must be read before entering ALT access mode.
2. Bits 5, 3, 1, and 0 return 0.

### 5.13.10.2 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 70.

**Table 70. 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 |

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

The registers described in Table 71 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 71. 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. |

## 5.13.11 System Power Supplies, Planes, and Signals

### 5.13.11.1 Power Plane Control with SLP_S3#, SLP_S4# and SLP_S5#

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 the ICH5 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.

### 5.13.11.2 SLP_S4# and Suspend-To-RAM Sequencing

The system memory suspend voltage regulator is controlled by the Glue logic "LATCHED_BACKFEED_CUT" signal. This signal should be generated using the SLP_S4# signal rather than the SLP_S5# signal, even if the platform does not support S4 Sleep State. The SLP_S4# logic in the ICH5 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.

### 5.13.11.3 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. SYSRESET# 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, the ICH5 does not provide any mechanism to limit the amount of time that the processor is held in reset. The platform must externally guarantee that maximum reset assertion specs 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 the ICH5 PWROK_FLR bit will be set. The ICH5 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 the ICH5 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 the ICH5.
5. In the case of true PWROK failure, PWROK goes low first before the VRMPWRGD.

### 5.13.11.4 VRMPWRGD Signal

This signal is connected to the processor's VRM and is internally AND'd with the PWROK signal that comes from the system power supply. This saves the external AND gate found in desktop systems.

### 5.13.11.5 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.

## 5.13.12 Clock Generators

The clock generator is expected to provide the frequencies shown in Table 72.

**Table 72. Intel® ICH5 Clock Inputs**

| Clock Domain | Frequency | Source | Usage |
|---|---|---|---|
| CLK100 | 100 MHz Differential | Main Clock Generator | Used by SATA controller. Stopped in S3 ~ S5 based on SLP_S3# assertion. |
| CLK66 | 66 MHz | Main Clock Generator | Should be running in all Cx states. Stopped in S3 ~ S5 based on SLP_S3# assertion. |
| PCICLK | 33 MHz | Main Clock Generator | Free-running PCI Clock to ICH5. Stopped in S3 ~ S5 based on SLP_S3# assertion. |
| CLK48 | 48 MHz | Main Clock Generator | Used by USB controllers. 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. |
| AC_BIT_CLK | 12.288 MHz | AC '97 Codec | AC-link. Control policy is determined by the clock source. |
| LAN_CLK | 0.8 to 50 MHz | LAN Connect | LAN Connect link. Control policy is determined by the clock source. |

### 5.13.13 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 OS is assumed to be at least APM enabled. Without APM calls, there is no quick way to know when the system is idle between keystrokes. The ICH5 does not support burst modes.

#### 5.13.13.1 APM Power Management

The ICH5 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 System Management (D31:F0)

The ICH5 provides various functions to make a system easier to manage and to lower the Total Cost of Ownership (TCO) of the system. In addition, ICH5 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 the ICH5:

- 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 flash BIOS programming
  - Detects if data on first read is FFh (indicates unprogrammed flash BIOS)
- 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 8.1.26)
- Integrated ASF Management support

***Note:*** Voltage ID from the processor can be read via GPI signals.

## 5.14.1 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.

### 5.14.1.1 Detecting a System Lockup

When the processor is reset, it is expected to fetch its first instruction. If the processor fails to fetch the first instruction after reset, the TCO timer times out twice and the ICH5 asserts PCIRST#.

### 5.14.1.2 Handling an Intruder

The ICH5 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 the ICH5 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 the ICH5'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 guarantee 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.

### 5.14.1.3 Detecting Improper Flash BIOS Programming

The ICH5 can detect the case where the flash BIOS is not programmed. This results in the first instruction fetched to have a value of FFh. If this occurs, the ICH5 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 5.14.2).

#### 5.14.1.4 Handling an ECC Error or Other Memory Error

The Host controller provides a message to indicate that it would like to cause an SMI#, SCI, SERR#, or NMI. The software must check the host controller as to the exact cause of the error.

### 5.14.2 Heartbeat and Event Reporting via SMBUS

The ICH5 integrated LAN controller supports ASF heartbeat and event reporting functionality when used with the 82562EM or 82562EZ 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 the ICH5 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 the ICH5.

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, the ICH5 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 the ICH5 **to reboot** the system after a hardware lockup:

1. On detecting the lockup, the SECOND_TO_STS bit is set. The ICH5 may send up to 1 Event message to the LAN controller. The ICH5 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. The ICH5 does not attempt to automatically reboot again. The ICH5 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. The ICH5 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, the ICH5 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, the ICH5 continues sending a message every heartbeat period. The ICH5 does not attempt to automatically reboot again. The ICH5 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), the ICH5 attempts to reset the system.
9. After step 8 (reset attempt) if the reset is successful, the BIOS is run. The ICH5 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, the ICH5 continues sending a message every heartbeat period. The ICH5 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 the ICH5 to **not reboot** the system after a hardware lockup.

1. On detecting the lockup the SECOND_TO_STS bit is set. The ICH5 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, the ICH5 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/step11.
4. After step 3 (third timeout), if the user does a Power Button Override, the system goes to an S5 state. The ICH5 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, the ICH5 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, the ICH5 continues sending heartbeats. The ICH5 does not attempt to reboot the system again until some external intervention occurs (reset, power failure, etc.). (See note 3)
8. After step 3 (3$^{rd}$ timeout), if a reset is attempted (using a button that pulses PWROK low or via the message on the SMBus slave I/F), the ICH5 attempts to reset the system.
9. If step 8 (reset attempt) is successful, the BIOS is run. The ICH5 continues sending heartbeats until the BIOS clears the SECOND_TO_STS bit. (See note 2)
10. If step 8 (reset attempt), is unsuccessful, the ICH5 continues sending heartbeats. The ICH5 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. The ICH5 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, the ICH5 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, the ICH5 continues sending heartbeats. The ICH5 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), the ICH5 attempts to reset the system.
17. If step 16 (reset attempt) is successful, the BIOS is run. The ICH5 continues sending heartbeats until the BIOS clears the SECOND_TO_STS bit. (See note 2)
18. If step 16 (reset attempt), is unsuccessful, the ICH5 continues sending heartbeats. The ICH5 does not attempt to reboot the system again without external intervention. (See note 3)

If the system is in a G1 (S1–S4) state, the ICH5 sends a heartbeat message every 30–32 seconds. If an event occurs prior to the system being shutdown, the ICH5 immediately sends an event message with the next incremented sequence number. After the event message, the ICH5 resumes sending heartbeat messages.

***Note:*** Notes for previous two numbered lists.

1. Normally, the ICH5 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. The ICH5 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 the ICH5 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 OS'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 ICH5'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 GPI11 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 73 shows the data included in the Alert on LAN messages.

**Table 73. 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 the Intel® ICH5 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 Flash BIOS Event Status | 1 = First BIOS fetch returned a value of FFh, indicating that the flash BIOS 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 the ICH5 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. |

# 5.15 General Purpose I/O

## 5.15.1 GPIO Mapping

**Table 74. GPIO Implementation (Sheet 1 of 4)**

| GPIO | Type | Alternate Function (Note 1) | Power Well | Tolerant | Notes |
|---|---|---|---|---|---|
| GPI0 | Input Only | REQA# | Core | 5.0 V | • GPIO_USE_SEL bit 0 enables REQ/GNTA# pair.<br>• Input active status read from GPE0_STS register bit 0.<br>• Input active high/low set through GPI_INV register bit 0. |
| GPI1 | Input Only | REQB# or REQ5# | Core | 5.0 V | • GPIO_USE_SEL bit 1 enables REQ/GNTB# pair (See note 4).<br>• Input active status read from GPE0_STS register bit 1.<br>• Input active high/low set through GPI_INV register bit 1. |
| GPI[5:2] | Input Only | PIRQ[E:H]# | Core | 5.0 V | • GPIO_USE_SEL bits [2:5] enable PIRQ[E:H]#.<br>• Input active status read from GPE0_STS reg. bits [2:5].<br>• Input active high/low set through GPI_INV reg. bit [2:5]. |
| GPI6 | Input Only | N/A | Core | 5.0 V | • Input active status read from GPE0_STS register bit 6.<br>• Input active high/low set through GPI_INV register bit 6. |
| GPI7 | Input Only | Unmuxed | Core | 5.0 V | • Input active status read from GPE0_STS register bit 7.<br>• Input active high/low set through GPI_INV register bit 7 |
| GPI8 | Input Only | Unmuxed | Resume | 3.3 V | • Input active status read from GPE0_STS register bit 8.<br>• Input active high/low set through GPI_INV register bit 8. |
| GPIO[10:9] | Input Only | OC[4:5]# | Resume | 3.3 V | • GPIO_USE_SEL bits [9:10] enable OC[4:5]#.<br>• Input active status read from GPE0_STS register bits [9:10].<br>• Input active high/low set through GPI_INV register bits [9:10]. |
| GPI11 | Input Only | SMBALERT# | Resume | 3.3 V | • GPIO_USE_SEL bit 11 enables SMBALERT#<br>• Input active status read from GPE0_STS register bit 11.<br>• Input active high/low set through GPI_INV register bit 11. |

### Table 74. GPIO Implementation (Sheet 2 of 4)

| GPIO | Type | Alternate Function (Note 1) | Power Well | Tolerant | Notes |
|---|---|---|---|---|---|
| GPI12 | Input Only | Unmuxed | Resume | 3.3 V | • Input active status read from GPE0_STS register bit 12.<br>• Input active high/low set through GPI_INV register bit 12. |
| GPI13 | Input Only | Unmuxed | Resume | 3.3 V | • Input active status read from GPE0_STS register bit 13.<br>• Input active high/low set through GPI_INV register bit 13. |
| GPIO[15:14] | Input Only | OC[6:7]# | Resume | 3.3 V | • GPIO_USE_SEL bits [14:15] enable OC[6:7]#.<br>• Input active status read from GPE0_STS register bits [14:15].<br>• Input active high/low set through GPI_INV register bits[14:15]. |
| GPO16 | Output Only | GNTA# | Core | 3.3 V | • Output controlled via GP_LVL register bit 16.<br>TTL driver output |
| GPO17 | Output Only | GNTB# or GNT5# | Core | 3.3 V | • Output controlled via GP_LVL register bit 17.<br>• TTL driver output |
| GPO18 | Output Only | N/A | Core | 3.3 V | • Blink enabled via GPO_BLINK register (D31:F0: Offset GPIOBASE+18h) bits [19:18]<br>• GPO18 will blink by default immediately after reset.<br>• Output controlled via GP_LVL register (D31:F0: Offset GPIOBASE+0Ch) bits [18:19].<br>• TTL driver output |
| GPO19 | Output Only | N/A | Core | 3.3 V | • Blink enabled via GPO_BLINK register (D31:F0: Offset GPIOBASE+18h) bits [19:18]<br>• Output controlled via GP_LVL register (D31:F0: Offset GPIOBASE+0Ch) bits [18:19].<br>• TTL driver output |
| GPO20 | Output Only | N/A | Core | 3.3 V | • Output controlled via GP_LVL register bit 20.<br>• TTL driver output |
| GPIO21 | Output Only | N/A | Core | 3.3 V | • Output controlled via GP_LVL register bit 21.<br>• TTL driver output |
| GPIO22 | Output Only | N/A | Core | 3.3 V | • Output controlled via GP_LVL register bit 22.<br>• Open-drain output |
| GPIO23 | Output Only | N/A | Core | 3.3 V | • Output controlled via GP_LVL register bit 23.<br>• TTL driver output |

### Table 74. GPIO Implementation (Sheet 3 of 4)

| GPIO | Type | Alternate Function (Note 1) | Power Well | Tolerant | Notes |
|---|---|---|---|---|---|
| GPIO24 | I/O | N/A | Resume | 3.3 V | • Input active status read from GP_LVL register bit 24.<br>• Output controlled via GP_LVL register bit 24.<br>• TTL driver output |
| GPIO25 | I/O | Unmuxed | Resume | 3.3 V | • Blink enabled via GPO_BLINK register (D31:F0: Offset GPIOBASE+18h) bit 25<br>• Input active status read from GP_LVL register bit 25<br>• Output controlled via GP_LVL register bit 25.<br>• TTL driver output |
| GPIO26 | N/A | N/A | N/A | | • Not implemented |
| GPIO[28:27] | I/O | Unmuxed | Resume | 3.3 V | • Blink enabled via GPO_BLINK register (D31:F0: Offset GPIOBASE+18h) bits [28:27]<br>• Input active status read from GP_LVL register bits [27:28]<br>• Output controlled via GP_LVL register bits [27:28]<br>• TTL driver output |
| GPIO[31:29] | N/A | N/A | N/A | | • Not implemented |
| GPIO32 | I/O | Unmuxed | Core | 3.3 V | • Input active status read from GP_LVL register bit 32<br>• Output controlled via GP_LVL2 register bit 32<br>• TTL driver output |
| GPIO33 | N/A | N/A | N/A | | • Not implemented |
| GPIO34 | I/O | Unmuxed | Core | 3.3 V | • Input active status read from GP_LVL register bit 34<br>• Output controlled via GP_LVL2 register bit 34<br>• TTL driver output |
| GPIO[39:35] | N/A | N/A | N/A | | • Not implemented |
| GPIO40 | Input Only | REQ4# | Core | 3.3 V | • GPIO_USE_SEL bit 40 enables REQ/GNT4# pair.<br>• Input active status read from GP_LVL2 register bit 40<br>• TTL driver output |
| GPIO41 | Input Only | LDRQ1# | Core | 3.3 V | • GPIO_USE_SEL bit 41 enables LDRQ1#. Input active status read from GP_LVL2 register bit 41<br>• TTL driver output |

**Table 74. GPIO Implementation (Sheet 4 of 4)**

| GPIO | Type | Alternate Function (Note 1) | Power Well | Tolerant | Notes |
|---|---|---|---|---|---|
| GPIO[47:42] | N/A | N/A | N/A | | • Not implemented |
| GPIO48 | Output Only | GNT4# | Core | 3.3 V | • Output controlled via GP_LVL2 register bit 48<br>• TTL driver output |
| GPIO49 | Output Only | CPUPWRGD | CPU I/F | 3.3 V | • GPIO_USE_SEL bit 49 enables CPUPWRGD.<br>• Output controlled via GP_LVL2 register bit 49<br>• TTL driver output |

**NOTES:**
1. All GPIOs default to their alternate function.
2. All inputs are sticky. The status bit remains set as long as the input was asserted for two clocks. GPIs are sampled on PCI clocks in S0/S1. GPIs are sampled on RTC clocks in S3/S4/S5.
3. GPIO[0:7] are 5 V tolerant, and all GPIs can be routed to cause an SCI or SMI#.
4. If GPIO_USE_SEL bit 1 is set to 1 and GEN_CNT bit 25 is also set to 1 then REQ/GNT5# is enabled. See Section 9.1.22.

## 5.15.2 Power Wells

Some GPIOs exist in the resume power plane. Care must be taken to make sure GPIO signals are not driven high into powered-down planes.

Some ICH5 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 results in the ICH5 driving a pin to a logic 1 to another device that is powered down.

GPIO[1:15] have “sticky” bits on the input. Refer to the GPE0_STS register. As long as the signal goes active for at least 2 clocks, the ICH5 keeps the sticky status bit active. The active level can be selected in the GP_LVL register.

If the system is in an S0 or an S1 state, the GPI inputs are sampled at 33 MHz, so the signal only needs to be active for about 60 ns to be latched. In the S3–S5 states, the GPI inputs are sampled at 32.768 kHz, and thus must be active for at least 61 microseconds to be latched.

If the input signal is still active when the latch is cleared, it will again be set. Another edge trigger is not required. This makes these signals “level” triggered inputs.

## 5.15.3 SMI# and SCI Routing

The routing bits for GPIO[0:15] allow an input to be routed to SMI# or SCI, or neither. Note that a bit can be routed to either an SMI# or an SCI, but not both.

# 5.16 IDE Controller (D31:F1)

The ICH5 IDE controller features two sets of interface signals (Primary and Secondary) that can be independently enabled, tri-stated or driven low. In addition, the ICH5 IDE controller supports both legacy mode and native mode IDE interface. In native mode, the IDE controller is a fully PCI compliant software interface and does not use any legacy I/O or interrupt resources.

The IDE interfaces of the ICH5 can support several types of data transfers:

- **Programmed I/O (PIO):** Processor is in control of the data transfer.
- **8237 style DMA:** DMA protocol that resembles the DMA on the ISA bus, although it does not use the 8237 in the ICH5. This protocol off loads the processor from moving data. This allows higher transfer rate of up to 16 MB/s.
- **Ultra ATA/33:** DMA protocol that redefines signals on the IDE cable to allow both host and target throttling of data and transfer rates of up to 33 MB/s.
- **Ultra ATA/66:** DMA protocol that redefines signals on the IDE cable to allow both host and target throttling of data and transfer rates of up to 66 MB/s.
- **Ultra ATA/100:** DMA protocol that redefines signals on the IDE cable to allow both host and target throttling of data and transfer rates of up to 100 MB/s.

## 5.16.1 PIO Transfers

The ICH5 IDE controller includes both compatible and fast timing modes. The fast timing modes can be enabled only for the IDE data ports. All other transactions to the IDE registers are run in single transaction mode with compatible timings.

Up to two IDE devices may be attached per IDE connector (drive 0 and drive 1). The IDETIM and SIDETIM Registers permit different timing modes to be programmed for drive 0 and drive 1 of the same connector.

The Ultra ATA/33/66/100 synchronous DMA timing modes can also be applied to each drive by programming the IDE I/O Configuration register and the Synchronous DMA Control and Timing registers. When a drive is enabled for synchronous DMA mode operation, the DMA transfers are executed with the synchronous DMA timings. The PIO transfers are executed using compatible timings or fast timings if also enabled.

### 5.16.1.1 IDE Port Decode

The Command and Control Block registers are accessed differently depending on the decode mode, which is selected by the Programming Interface configuration register (Offset 09h).

***Note:*** The primary and secondary channels are controlled by separate bits, allowing one to be in native mode and the other in legacy mode simultaneously.

### 5.16.1.2 IDE Legacy Mode and Native Mode

The ICH5 IDE controller supports both legacy mode and PCI native mode. In legacy mode, the Command and Control Block registers are accessible at fixed I/O addresses. While in legacy mode, the ICH5 does not decode any of the native mode ranges. Likewise, in native mode the ICH5 does not decode any of the legacy mode ranges.

The IDE I/O ports involved in PIO transfers are decoded by the ICH5 to the IDE interface when D31:F1 I/O space is enabled and IDE decode is enabled through the IDE_TIMx registers. The IDE registers are implemented in the drive itself. An access to the IDE registers results in the assertion of the appropriate IDE chip select for the register, and the IDE command strobes (PDIOR#/ SDIOR#, PDIOW#/SDIOW#).

There are two I/O ranges for each IDE cable: the Command Block, which corresponds to the PCS1#/SCS1# chip select, and the Control Block, which corresponds to the PCS3#/SCS3# chip select. The Command Block is an 8-byte range, while the control block is a 4-byte range.

— **Command Block Offset:** 01F0h for Primary, 0170h for Secondary

— **Control Block Offset:** 03F4h for Primary, 0374h for Secondary

Table 75 specifies the registers as they affect the ICH5 hardware definition.

***Note:*** The Data Register (I/O Offset 00h) should be accessed using 16-bit or 32-bit I/O instructions. All other registers should be accessed using 8-bit I/O instructions.

**Table 75. IDE Legacy I/O Ports: Command Block Registers (CS1x# Chip Select)**

| I/O Offset | Register Function (Read) | Register Function (Write) |
|---|---|---|
| 00h | Data | Data |
| 01h | Error | Features |
| 02h | Sector Count | Sector Count |
| 03h | Sector Number | Sector Number |
| 04h | Cylinder Low | Cylinder Low |
| 05h | Cylinder High | Cylinder High |
| 06h | Drive | Head |
| 07h | Status | Command |

**NOTE:** For accesses to the Alt Status register in the Control Block, the ICH5 must always force the upper address bit (PDA2 or SDA2) to 1 in order to guarantee proper native mode decode by the IDE device. Unlike the legacy mode fixed address location, the native mode address for this register may contain a 0 in address bit 2 when it is received by the ICH5.

In native mode, the ICH5 does not decode the legacy ranges. The same offsets are used as in Table 75. However, the base addresses are selected using the PCI BARs, rather than fixed I/O locations.

### 5.16.1.3 PIO IDE Timing Modes

IDE data port transaction latency consists of startup latency, cycle latency, and shutdown latency. Startup latency is incurred when a PCI master cycle targeting the IDE data port is decoded and the DA[2:0] and CSxx# lines are not set up. Startup latency provides the setup time for the DA[2:0] and CSxx# lines prior to assertion of the read and write strobes (DIOR# and DIOW#).

Cycle latency consists of the I/O command strobe assertion length and recovery time. Recovery time is provided so that transactions may occur back-to-back on the IDE interface (without incurring startup and shutdown latency) without violating minimum cycle periods for the IDE interface. The command strobe assertion width for the enhanced timing mode is selected by the IDE_TIM Register and may be set to 2, 3, 4, or 5 PCI clocks. The recovery time is selected by the IDE_TIM Register and may be set to 1, 2, 3, or 4 PCI clocks.

If IORDY is asserted when the initial sample point is reached, no wait-states are added to the command strobe assertion length. If IORDY is negated when the initial sample point is reached, additional wait-states are added. Since the rising edge of IORDY must be synchronized, at least two additional PCI clocks are added.

Shutdown latency is incurred after outstanding scheduled IDE data port transactions (either a non-empty write post buffer or an outstanding read prefetch cycles) have completed and before other transactions can proceed. It provides hold time on the DA[2:0] and CSxx# lines with respect to the read and write strobes (DIOR# and DIOW#). Shutdown latency is two PCI clocks in duration.

The IDE timings for various transaction types are shown in Table 76. Note that bit 2 (16-bit I/O recovery enable) of the ISA I/O Recovery Timer Register does not add wait-states to IDE data port read accesses when any of the fast timing modes are enabled.

**Table 76. IDE Transaction Timings (PCI Clocks)**

| IDE Transaction Type | Startup Latency | IORDY Sample Point (ISP) | Recovery Time (RCT) | Shutdown Latency |
|---|---|---|---|---|
| Non-Data Port Compatible | 4 | 11 | 22 | 2 |
| Data Port Compatible | 3 | 6 | 14 | 2 |
| Fast Timing Mode | 2 | 2–5 | 1–4 | 2 |

### 5.16.1.4 IORDY Masking

The IORDY signal can be ignored and assumed asserted at the first IORDY Sample Point (ISP) on a drive by drive basis via the IDETIM Register.

### 5.16.1.5 PIO 32-Bit IDE Data Port Accesses

A 32-bit PCI transaction run to the IDE data address (01F0h primary, 0170h secondary) results in two back to back 16-bit transactions to the IDE data port. The 32-bit data port feature is enabled for all timings, not just enhanced timing. For compatible timings, a shutdown and startup latency is incurred between the two, 16-bit halves of the IDE transaction. This guarantees that the chip selects are deasserted for at least two PCI clocks between the two cycles.

#### 5.16.1.6 PIO IDE Data Port Prefetching and Posting

The ICH5 can be programmed via the IDETIM registers to allow data to be posted to and prefetched from the IDE data ports.

Data prefetching is initiated when a data port read occurs. The read prefetch eliminates latency to the IDE data ports and allows them to be performed back to back for the highest possible PIO data transfer rates. The first data port read of a sector is called the demand read. Subsequent data port reads from the sector are called prefetch reads. The demand read and all prefetch reads much be of the same size (16 or 32 bits).

Data posting is performed for writes to the IDE data ports. The transaction is completed on the PCI bus after the data is received by the ICH5. The ICH5 then runs the IDE cycle to transfer the data to the drive. If the ICH5 write buffer is non-empty and an unrelated (non-data or opposite channel) IDE transaction occurs, that transaction will be stalled until all current data in the write buffer is transferred to the drive.

### 5.16.2 Bus Master Function

The ICH5 can act as a PCI Bus master on behalf of an IDE slave device. Two PCI Bus master channels are provided, one channel for each IDE connector (primary and secondary). By performing the IDE data transfer as a PCI Bus master, the ICH5 off-loads the processor and improves system performance in multitasking environments. Both devices attached to a connector can be programmed for bus master transfers, but only one device per connector can be active at a time.

#### 5.16.2.1 Physical Region Descriptor Format

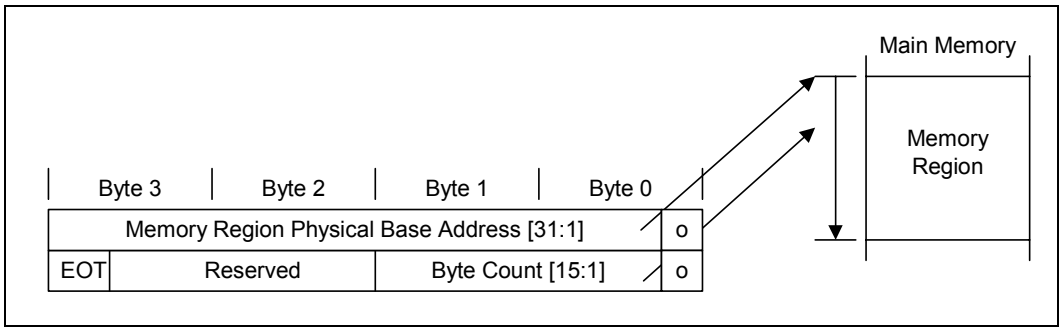
The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The PRDs are stored sequentially in a Descriptor Table in memory. The data transfer proceeds until all regions described by the PRDs in the table have been transferred. Note that the ICH5 bus master IDE function does not support memory regions or Descriptor tables located on ISA.

Descriptor Tables must not cross a 64-KB boundary. Each PRD entry in the table is 8 bytes in length. The first 4 bytes specify the byte address of a physical memory region. This memory region must be DWord-aligned and must not cross a 64-Kbyte boundary. The next two bytes specify the size or transfer count of the region in bytes (64-Kbyte limit per region). A value of 0 in these two bytes indicates 64 Kbytes (thus the minimum transfer count is 1). If bit 7 (EOT) of the last byte is a 1, it indicates that this is the final PRD in the Descriptor table. Bus master operation terminates when the last descriptor has been retired.

When the Bus Master IDE controller is reading data from the memory regions, bit 1 of the Base Address is masked and byte enables are asserted for all read transfers. When writing data, bit 1 of the Base Address is not masked and if set, will cause the lower Word byte enables to be deasserted for the first DWord transfer. The write to PCI typically consists of a 32-byte cache line. If valid data ends prior to end of the cache line, the byte enables will be deasserted for invalid data.

The total sum of the byte counts in every PRD of the descriptor table must be equal to or greater than the size of the disk transfer request. If greater than the disk transfer request, the driver must terminate the bus master transaction (by setting bit 0 in the Bus Master IDE Command Register to 0) when the drive issues an interrupt to signal transfer completion.

Figure 18. Physical Region Descriptor Table Entry




### 5.16.2.2 Line Buffer

A single line buffer exists for the ICH5 Bus master IDE interface. This buffer is not shared with any other function. The buffer is maintained in either the read state or the write state. Memory writes are typically 4-DWord bursts and invalid DWords have C/BE[3:0]#=0Fh. The line buffer allows burst data transfers to proceed at peak transfer rates.

The Bus Master IDE Active bit in Bus Master IDE Status register is reset automatically when the controller has transferred all data associated with a Descriptor Table (as determined by EOT bit in last PRD). The IDE Interrupt Status bit is set when the IDE device generates an interrupt. These events may occur prior to line buffer emptying for memory writes. If either of these conditions exist, all PCI Master non-Memory read accesses to ICH5 are retried until all data in the line buffers has been transferred to memory.

### 5.16.2.3 Bus Master IDE Timings

The timing modes used for Bus Master IDE transfers are identical to those for PIO transfers. The DMA Timing Enable Only bits in IDE Timing register can be used to program fast timing mode for DMA transactions only. This is useful for IDE devices whose DMA transfer timings are faster that its PIO transfer timings. The IDE device DMA request signal is sampled on the same PCI clock that DIOR# or DIOW# is deasserted. If inactive, the DMA Acknowledge signal is deasserted on the next PCI clock and no more transfers take place until DMA request is asserted again.

### 5.16.2.4 Interrupts

#### Legacy Mode

The ICH5 is connected to IRQ14 for the primary interrupt and IRQ15 for the secondary interrupt. This connection is done from the ISA pin, before any mask registers. This implies the following:

- Bus Master IDE devices are connected directly off of ICH5. IDE interrupts cannot be communicated through PCI devices or the serial stream.

***Warning:*** In this mode, the ICH5 does not drive the PCI Interrupt associated with this function. That is only used in native mode.

### Native Mode

In this case both the Primary and Secondary channels share an interrupt. It is internally connected to PIRQC# (IRQ18 in APIC mode). The interrupt is active-low and shared.

Behavioral notes in native mode

- The IRQ14 and IRQ15 pins do not affect the internal IRQ14 and IRQ15 inputs to the interrupt controllers. The IDE logic forces these signals inactive in such a way that the Serial IRQ source may be used.
- The IRQ14 and IRQ15 inputs (not external IRQ[14:15] pins) to the interrupt controller can come from other sources (Serial IRQ, PIRQx).
- The IRQ14 and IRQ15 pins are inverted from active-high to the active-low PIRQ.
- When switching the IDE controller to native mode, the IDE Interrupt Pin Register (see Section 10.1.18) is masked. If an interrupt occurs while the masking is in place and the interrupt is still active when the masking ends, the interrupt is allowed to be asserted.

## 5.16.2.5 Bus Master IDE Operation

To initiate a bus master transfer between memory and an IDE device, the following steps are required:

1. Software prepares a PRD table in system memory. The PRD table must be DWord aligned and must not cross a 64-KB boundary.
2. Software provides the starting address of the PRD Table by loading the PRD Table Pointer Register. The direction of the data transfer is specified by setting the Read/Write Control bit. The interrupt bit and Error bit in the Status register are cleared.
3. Software issues the appropriate DMA transfer command to the disk device.
4. The bus master function is engaged by software writing a 1 to the Start bit in the Command Register. The first entry in the PRD table is fetched and loaded into two registers which are not visible by software, the Current Base and Current Count registers. These registers hold the current value of the address and byte count loaded from the PRD table. The value in these registers is only valid when there is an active command to an IDE device.
5. Once the PRD is loaded internally, the IDE device will receive a DMA acknowledge.
6. The controller transfers data to/from memory responding to DMA requests from the IDE device. The IDE device and the host controller may or may not throttle the transfer several times. When the last data transfer for a region has been completed on the IDE interface, the next descriptor is fetched from the table. The descriptor contents are loaded into the Current Base and Current Count registers.
7. At the end of the transfer, the IDE device signals an interrupt.
8. In response to the interrupt, software resets the Start/Stop bit in the command register. It then reads the controller status followed by the drive status to determine if the transfer completed successfully.

The last PRD in a table has the End of List (EOL) bit set. The PCI bus master data transfers terminate when the physical region described by the last PRD in the table has been completely transferred. The active bit in the Status Register is reset and the DDRQ signal is masked.

The buffer is flushed (when in the write state) or invalidated (when in the read state) when a terminal count condition exists; that is, the current region descriptor has the EOL bit set and that region has been exhausted. The buffer is also flushed (write state) or invalidated (read state) when the Interrupt bit in the Bus Master IDE Status register is set. Software that reads the status register and finds the Error bit reset, and either the Active bit reset or the Interrupt bit set, can be assured that all data destined for system memory has been transferred and that data is valid in system memory. Table 77 describes how to interpret the Interrupt and Active bits in the Status Register after a DMA transfer has started.

During concurrent DMA or Ultra ATA transfers, the ICH5 IDE interface arbitrate between the primary and secondary IDE cables when a PRD expires.

**Table 77. Interrupt/Active Bit Interaction Definition**

| Interrupt | Active | Description |
|---|---|---|
| 0 | 1 | DMA transfer is in progress. No interrupt has been generated by the IDE device. |
| 1 | 0 | The IDE device generated an interrupt. The controller exhausted the Physical Region Descriptors. This is the normal completion case where the size of the physical memory regions was equal to the IDE device transfer size. |
| 1 | 1 | The IDE device generated an interrupt. The controller has not reached the end of the physical memory regions. This is a valid completion case where the size of the physical memory regions was larger than the IDE device transfer size. |
| 0 | 0 | This bit combination signals an error condition. If the Error bit in the status register is set, then the controller has some problem transferring data to/from memory. Specifics of the error have to be determined using bus-specific information. If the Error bit is not set, then the PRD's specified a smaller size than the IDE transfer size. |

### 5.16.2.6 Error Conditions

IDE devices are sector based mass storage devices. The drivers handle errors on a sector basis; either a sector is transferred successfully or it is not. A sector is 512 bytes.

If the IDE device does not complete the transfer due to a hardware or software error, the command will eventually be stopped by the driver setting Command Start bit to 0 when the driver times out the disk transaction. Information in the IDE device registers help isolate the cause of the problem.

If the controller encounters an error while doing the bus master transfers it will stop the transfer (i.e., reset the Active bit in the Command register) and set the Error bit in the Bus Master IDE Status register. The controller does not generate an interrupt when this happens. The device driver can use device specific information (PCI Configuration Space Status register and IDE Drive Register) to determine what caused the error.

Whenever a requested transfer does not complete properly, information in the IDE device registers (Sector Count) can be used to determine how much of the transfer was completed and to construct a new PRD table to complete the requested operation. In most cases the existing PRD table can be used to complete the operation.

### 5.16.2.7 8237-Like Protocol

The 8237 mode DMA is similar in form to DMA used on the ISA bus. This mode uses pins familiar to the ISA bus, namely a DMA Request, a DMA Acknowledge, and I/O read/write strobes. These pins have similar characteristics to their ISA counterparts in terms of when data is valid relative to strobe edges, and the polarity of the strobes, however the ICH5 does not use the 8237 for this mode.

## 5.16.3 Ultra ATA/33 Protocol

Ultra ATA/33 is enabled through configuration register 48h in Device 31:Function 1 for each IDE device. The IDE signal protocols are significantly different under this mode than for the 8237 mode.

Ultra ATA/33 is a physical protocol used to transfer data between a Ultra ATA/33 capable IDE controller (e.g., the ICH5) and one or more Ultra ATA/33 capable IDE devices. It utilizes the standard Bus Master IDE functionality and interface to initiate and control the transfer. Ultra ATA/33 utilizes a "source synchronous" signaling protocol to transfer data at rates up to 33 MB/s. The Ultra ATA/33 definition also incorporates a Cyclic Redundancy Checking (CRC-16) error checking protocol.

### 5.16.3.1 Signal Descriptions

The Ultra ATA/33 protocol requires no extra signal pins on the IDE connector. It does redefine a number of the standard IDE control signals when in Ultra ATA/33 mode. These redefinitions are shown in Table 78. Read cycles are defined as transferring data from the IDE device to the ICH5. Write cycles are defined as transferring data from ICH5 to IDE device.

**Table 78. UltraATA/33 Control Signal Redefinitions**

| Standard IDE Signal Definition | Ultra ATA/33 Read Cycle Definition | Ultra ATA/33 Write Cycle Definition | Intel® ICH5 Primary Channel Signal | Intel® ICH5 Secondary Channel Signal |
|---|---|---|---|---|
| DIOW# | STOP | STOP | PDIOW# | SDIOW# |
| DIOR# | DMARDY# | STROBE | PDIOR# | SDIOR# |
| IORDY | STROBE | DMARDY# | PIORDY | SIORDY |

The DIOW# signal is redefined as STOP for both read and write transfers. This is always driven by the ICH5 and is used to request that a transfer be stopped or as an acknowledgment to stop a request from the IDE device.

The DIOR# signal is redefined as DMARDY# for transferring data from the IDE device to the ICH5 (read). It is used by the ICH5 to signal when it is ready to transfer data and to add wait-states to the current transaction. The DIOR# signal is redefined as STROBE for transferring data from the ICH5 to the IDE device (write). It is the data strobe signal driven by the ICH5 on which data is transferred during each rising and falling edge transition.

The IORDY signal is redefined as STROBE for transferring data from the IDE device to the ICH5 (read). It is the data strobe signal driven by the IDE device on which data is transferred during each rising and falling edge transition. The IORDY signal is redefined as DMARDY# for transferring data from the ICH5 to the IDE device (write). It is used by the IDE device to signal when it is ready to transfer data and to add wait-states to the current transaction.

All other signals on the IDE connector retain their functional definitions during Ultra ATA/33 operation.

### 5.16.3.2 Operation

Initial setup programming consists of enabling and performing the proper configuration of the ICH5 and the IDE device for Ultra ATA/33 operation. For the ICH5, this consists of enabling synchronous DMA mode and setting up appropriate Synchronous DMA timings.

When ready to transfer data to or from an IDE device, the Bus Master IDE programming model is followed. Once programmed, the drive and ICH5 control the transfer of data via the Ultra ATA/33 protocol. The actual data transfer consists of three phases, a start-up phase, a data transfer phase, and a burst termination phase.

The IDE device begins the start-up phase by asserting DMARQ signal. When ready to begin the transfer, the ICH5 asserts DMACK# signal. When DMACK# signal is asserted, the host controller drives CS0# and CS1# inactive, DA0–DA2 low. For write cycles, the ICH5 deasserts STOP, waits for the IDE device to assert DMARDY#, and then drives the first data word and STROBE signal. For read cycles, the ICH5 tri-states the DD lines, deasserts STOP, and asserts DMARDY#. The IDE device then sends the first data word and STROBE.

The data transfer phase continues the burst transfers with the data transmitter (ICH5 – writes, IDE device – reads) providing data and toggling STROBE. Data is transferred (latched by receiver) on each rising and falling edge of STROBE. The transmitter can pause the burst by holding STROBE high or low, resuming the burst by again toggling STROBE. The receiver can pause the burst by deasserting DMARDY# and resumes the transfers by asserting DMARDY#. The ICH5 pauses a burst transaction to prevent an internal line buffer over or under flow condition, resuming once the condition has cleared. It may also pause a transaction if the current PRD byte count has expired, resuming once it has fetched the next PRD.

The current burst can be terminated by either the transmitter or receiver. A burst termination consists of a Stop Request, Stop Acknowledge and transfer of CRC data. The ICH5 can stop a burst by asserting STOP, with the IDE device acknowledging by deasserting DMARQ. The IDE device stops a burst by deasserting DMARQ and the ICH5 acknowledges by asserting STOP. The transmitter then drives the STROBE signal to a high level. The ICH5 then drives the CRC value onto the DD lines and deassert DMACK#. The IDE device latches the CRC value on rising edge of DMACK#. The ICH5 terminates a burst transfer if it needs to service the opposite IDE channel, if a Programmed I/O (PIO) cycle is executed to the IDE channel currently running the burst, or upon transferring the last data from the final PRD.

### 5.16.3.3 CRC Calculation

Cyclic Redundancy Checking (CRC-16) is used for error checking on Ultra ATA/33 transfers. The CRC value is calculated for all data by both the ICH5 and the IDE device over the duration of the Ultra ATA/33 burst transfer segment. This segment is defined as all data transferred with a valid STROBE edge from DDACK# assertion to DDACK# deassertion. At the end of the transfer burst segment, the ICH5 drives the CRC value onto the DD[15:0] signals. It is then latched by the IDE device on deassertion of DDACK#. The IDE device compares the ICH5 CRC value to its own and reports an error if there is a mismatch.

## 5.16.4 Ultra ATA/66 Protocol

In addition to Ultra ATA/33, the ICH5 supports the Ultra ATA/66 protocol. The Ultra ATA/66 protocol is enabled via configuration bits 3:0 at offset 54h. The two protocols are similar, and are intended to be device driver compatible. The Ultra ATA/66 logic can achieve transfer rates of up to 66 MB/s.

To achieve the higher data rate, the timings are shortened and the quality of the cable is improved to reduce reflections, noise, and inductive coupling. Note that the improved cable is required and still plugs into the standard IDE connector.

The Ultra ATA/66 protocol also supports a 44 MB/s mode.

## 5.16.5 Ultra ATA/100 Protocol

When the ATA_FAST bit is set for any of the four IDE devices, then the timings for the transfers to and from the corresponding device run at a higher rate. The ICH5 Ultra ATA/100 logic can achieve read transfer rates up to 100 MB/s, and write transfer rates up to 88.9 MB/s.

The cable improvements required for Ultra ATA/66 are sufficient for Ultra ATA/100, so no further cable improvements are required when implementing Ultra ATA/100.

## 5.16.6 Ultra ATA/33/66/100 Timing

The timings for Ultra ATA/33/66/100 modes are programmed via the Synchronous DMA Timing register and the IDE Configuration register. Different timings can be programmed for each drive in the system. The Base Clock frequency for each drive is selected in the IDE Configuration register. The Cycle Time (CT) and Ready to Pause (RP) time (defined as multiples of the Base Clock) are programmed in the Synchronous DMA Timing Register. The Cycle Time represents the minimum pulse width of the data strobe (STROBE) signal. The Ready to Pause time represents the number of Base Clock periods that the ICH5 waits from deassertion of DMARDY# to the assertion of STOP when it desires to stop a burst read transaction.

***Note:*** The internal Base Clock for Ultra ATA/100 (Mode 5) runs at 133 MHz, and the Cycle Time (CT) must be set for three Base Clocks. The ICH5 thus toggles the write strobe signal every 22.5 ns, transferring two bytes of data on each strobe edge. This means that the ICH5 performs Mode 5 write transfers at a maximum rate of 88.9 MB/s. For read transfers, the read strobe is driven by the ATA/100 device, and the ICH5 supports reads at the maximum rate of 100 MB/s.

## 5.16.7 IDE Swap Bay

To support a swap bay, the ICH5 allows the IDE output signals to be tri-stated and input buffers to be turned off. This should be done prior to the removal of the drive. The output signals can also be driven low. This can be used to remove charge built up on the signals. Configuration bits are included in the IDE I/O Configuration register, offset 54h in the IDE PCI configuration space.

In an IDE Hot Swap Operation, an IDE device is removed and a new one inserted while the IDE interface is powered down and the rest of the system is in a fully powered-on state (SO). During an IDE Hot Swap, if the operating system executes cycles to the IDE interface after it has been powered down it will cause the ICH5 to hang the system that is waiting for IORDY to be asserted from the drive.

To correct this issue, the following BIOS procedures are required for performing an IDE hot swap:

1. Program IDE SIG_MODE (Configuration register at offset 54h) to 10b (drive low mode).
2. Clear IORDY Sample Point Enable (bits 1 or 5 of IDE Timing reg.). This prevents the ICH5 from waiting for IORDY assertion when the operating system accesses the IDE device after the IDE drive powers down, and ensures that 0s are always be returned for read cycles that occur during hot swap operation.

***Warning:*** Software should **not** attempt to control the outputs (either tri-state or driving low), while an IDE transfer is in progress. Unpredictable results could occur, including a system lockup.

## 5.16.8 SMI Trapping (APM)

Offset 48h, bits 3:0 in the power management I/O space (see Section 9.10.14) 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 IDE 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 IDE 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 9.10.13) are updated indicating that a trap occurred. To block accesses to the native IDE ranges, software must use the generic power management control registers described in Section 9.8.1.

# 5.17 SATA Host Controller (D31:F2)

The SATA function in the ICH5 has dual modes of operation to support different operating system conditions. In the case of Native IDE enabled operating systems, the ICH5 has separate PCI functions for serial and parallel ATA. To support legacy operating systems, there is only one PCI function for both the serial and parallel ATA ports.

The MAP register, Section 11.1.32, 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. The SATA Capability Pointer Register (offset 34h) will change to indicate that MSI is not supported in combined mode.

The ICH5 SATA controller features two sets of interface signals that can be independently enabled or disabled (they cannot be tri-stated or driven low). Each interface is supported by an independent DMA controller.

The ICH5 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.

***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.

## 5.17.1 Theory of Operation

### 5.17.1.1 Standard ATA Emulation

The ICH5 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.

### 5.17.1.2 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.

## 5.17.2 Hot Swap Operation

Dynamic hot swap (e.g., surprise removal) is not supported by the SATA host controller. However, using the SPC register configuration bits, and power management flows, a device can be powered down by software, and the port can then be powered off, allowing removal and insertion of a new device.

***Note:*** This hot swap operation requires BIOS and OS support.

## 5.17.3 Intel® RAID Technology Configuration (Intel® 82801ER ICH5R Only)

The Intel® RAID Technology solution, available with the 82801ER ICH5 R (ICH5R), offers data stripping for higher performance (RAID Level 0), alleviating disk bottlenecks by taking advantage of the dual independent SATA controllers integrated in the ICH5R. There is no loss of PCI resources
(request/grant pair) or add-in card slot.

Intel RAID Technology functionality requires the following items:

1. ICH5R
2. Intel RAID Technology Option ROM must be on the platform
3. Intel® Application Accelerator RAID Edition drivers, most recent revision.
4. Two SATA hard disk drives.

Intel RAID Technology is not available in the following configurations:

1. The SATA controller in compatible mode.
2. Intel RAID Technology has been disabled - D31:F0:AE bits [7:6] have been cleared

### 5.17.3.1 Intel® RAID Technology Option ROM

The Intel RAID Technology for SATA Option ROM provides a pre-OS user interface for the Intel RAID Technology implementation and provides the ability for a Intel RAID Technology volume to be used as a boot disk as well as to detect any faults in the Intel RAID Technology volume(s) attached to the Intel RAID controller.

## 5.17.4 Power Management Operation

Power management of the ICH5 SATA controller and ports will cover operations of the host controller and the SATA wire.

### 5.17.4.1 Power State Mappings

The D0 PCI power management state for device is supported by the ICH5 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.

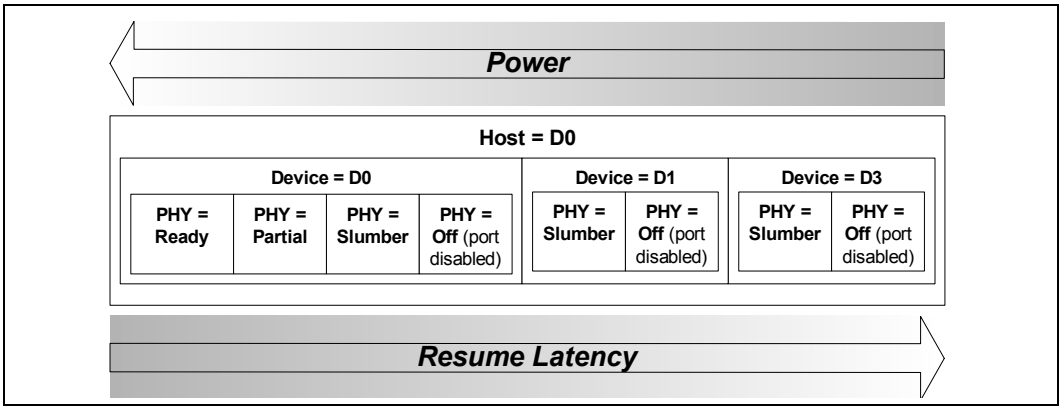
Each of these device states are subsets of the host controller's D0 state.

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 19. SATA Power States**




## 5.17.4.2 Power State Transitions

### 5.17.4.2.1 Partial and Slumber State Entry/Exit

The partial and slumber states save interface power when the interface is idle. 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.

#### 5.17.4.2.2 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.

### 5.17.4.3 SMI Trapping (APM)

Offset 48h, bits 3:0 in the power management I/O space (see Section 9.10.14) 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 9.10.13) are updated indicating that a trap occurred.

To block accesses to the native IDE ranges, software must use the generic power management control registers described in Section 9.8.9.

## 5.17.5 SATA Interrupts

Table 79 summarizes interrupt behavior for MSI and wire-modes. In the table “bits” refers to the four possible interrupt bits in I/O space, which are: PSTS.PRDIS (offset 02h, bit 7), PSTS.I (offset 02h, bit 2), SSTS.PRDIS (offset 0Ah, bit 7), and SSTS.I (offset 0Ah, bit 2).

**Table 79. SATA MSI vs. PCI IRQ Actions**

| Interrupt Register | Wire-Mode Action | MSI Action |
|---|---|---|
| All bits are 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 is set simultaneously | Wire Active | Send Message |

# 5.18 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.

ICH5 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 6.4). It is not expected that the operating system will move the location of these timers once it is set by the BIOS.

## 5.18.1 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.

## 5.18.2 Interrupt Mapping

### Mapping Option #1 (Legacy Option)

In this case, the Legacy Rout bit (LEG_RT_CNF) is set. This forces the mapping found in Table 80.

**Table 80. Legacy 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 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.

## 5.18.3 Periodic vs. Non-Periodic Modes

### Non-Periodic Mode

When a timer is set up for non-periodic mode, it generates a value in the main counter that matches the value in the timer's comparator register. If the timer is set up for 32-bit mode, then it generates another interrupt when the main counter wraps around and matches this same value again. Timer 0 is configurable to 32 (default) or 64-bit mode, whereas Timers 1 and 2 only support 32-bit mode (See Section 17.5).

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

***Warning:*** Software must be careful when programming the comparator registers. If the value written to the register is not sufficiently far in the future, then the counter may pass the value before it reaches the register and the interrupt will be missed.

All three timers support non-periodic mode.

### Periodic Mode

Timer 0 is the only timer that supports periodic mode. When Timer 0 is set up for periodic mode, the software writes a value into the timer's comparator value register. When the main counter value matches the value in the timer's comparator value register, an interrupt can be generated. The hardware then automatically increases the value in the comparator value register by the last value written to that register.

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

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

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

If the software resets the main counter, the value in the comparator's value register needs to reset as well. This can be done by setting the TIMER0_VAL_SET_CNF bit. Again, to avoid race conditions, this should be done with the main counter halted. The following usage model is expected:

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

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

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

## 5.18.4 Enabling the Timers

The BIOS or OS 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

## 5.18.5 Interrupt Levels

Interrupts directed to the internal 8259s are active high. SeeSection 5.9 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 OS 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.

## 5.18.6 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.

## 5.18.7 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.

# 5.19 USB UHCI Host Controllers (D29:F0, F1, F2, and F3)

The ICH5 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 8 USB ports.

- Overcurrent detection on all eight USB ports is supported. The overcurrent inputs are 5 V tolerant, and can be used as GPIs if not needed.
- The ICH5'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 high speed signaling rates, instead of USB I/O buffers.

## 5.19.1 Data Structures in Main Memory

This section describes the details of the data structures used to communicate control, status, and data between software and the ICH5: Frame Lists, Transfer Descriptors, and Queue Heads. Frame Lists are aligned on 4-KB boundaries. Transfer Descriptors and Queue Heads are aligned on 16-byte boundaries.

### 5.19.1.1 Frame List Pointer

The frame list pointer contains a link pointer to the first data object to be processed in the frame, as well as the control bits defined in Table 81.

**Table 81. Frame List Pointer Bit Description**

| Bit | Description |
|---|---|
| 31:4 | **Frame List Pointer (FLP).** This field contains the address of the first data object to be processed in the frame and corresponds to memory address signals [31:4], respectively. |
| 3:2 | Reserved. These bits must be written as 0. |
| 1 | **QH/TD Select (Q).** This bit indicates to the hardware whether the item referenced by the link pointer is a TD (Transfer Descriptor) or a QH (Queue Head). This allows the Intel® ICH5 to perform the proper type of processing on the item after it is fetched.<br>0 = TD<br>1 = QH |
| 0 | **Terminate (T).** This bit indicates to the ICH5 whether the schedule for this frame has valid entries in it.<br>0 = Pointer is valid (points to a QH or TD).<br>1 = Empty Frame (pointer is invalid). |

### 5.19.1.2 Transfer Descriptor (TD)

Transfer Descriptors (TDs) express the characteristics of the transaction requested on USB by a client. TDs are always aligned on 16-byte boundaries, and the elements of the TD are shown in Figure 20. The four, different USB transfer types are supported by a small number of control bits in the descriptor that the ICH5 interprets during operation. All TDs have the same, basic, 32-byte structure. During operation, the ICH5 hardware performs consistency checks on some fields of the TD. If a consistency check fails, the ICH5 halts immediately and issues an interrupt to the system. This interrupt cannot be masked within the ICH5.

**Figure 20. Transfer Descriptor**

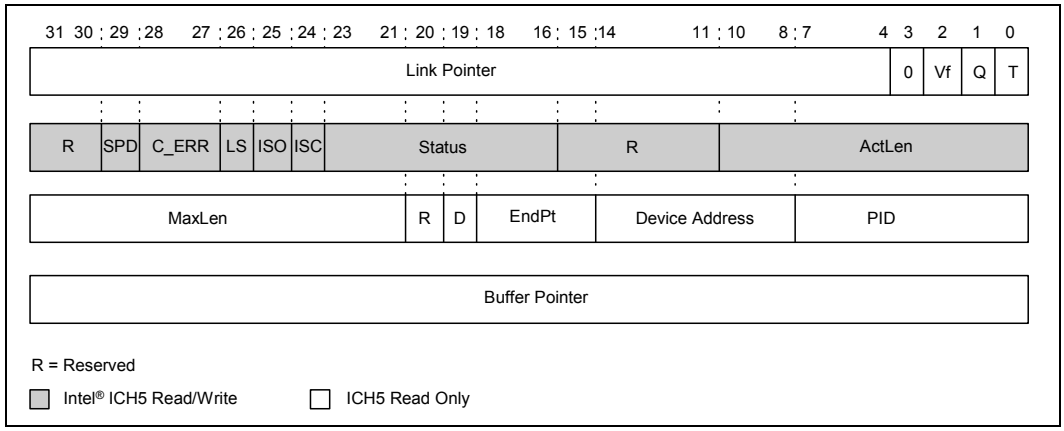



**Table 82. TD Link Pointer**

| Bits | Description |
|---|---|
| 31:4 | **Link Pointer (LP).** Bits [31:4] correspond to memory address signals [31:4], respectively. This field points to another TD or QH. |
| 3 | Reserved. Must be 0 when writing this field. |
| 2 | **Depth/Breadth Select (VF).** This bit is only valid for queued TDs and indicates to the hardware whether it should process in a depth first or breadth first fashion. When set to depth first, it informs the ICH5 to process the next transaction in the queue rather than starting a new queue.<br>0 = Breadth first<br>1 = Depth first |
| 1 | **QH/TD Select (Q).** This bit informs the Intel® ICH5 whether the item referenced by the link pointer is another TD or a QH. This allows the ICH5 to perform the proper type of processing on the item after it is fetched.<br>0 = TD<br>1 = QH |
| 0 | **Terminate (T).** This bit informs the ICH5 that the link pointer in this TD does not point to another valid entry. When encountered in a queue context, this bit indicates to the ICH5 that there are no more valid entries in the queue. A TD encountered outside of a queue context with the T bit set informs the ICH5 that this is the last TD in the frame.<br>0 = Link Pointer field is valid.<br>1 = Link Pointer field not valid. |

### Table 83. TD Control and Status (Sheet 1 of 2)

| Bit | Description |
|---|---|
| 31:30 | Reserved. |
| 29 | **Short Packet Detect (SPD).** When a packet has this bit set to 1 and the packet is an input packet, is in a queue; and successfully completes with an actual length less than the maximum length then the TD is marked inactive, the Queue Header is not updated and the USBINT status bit (Status Register) is set at the end of the frame. In addition, if the interrupt is enabled, the interrupt is sent at the end of the frame.<br>Note that any error (e.g., babble or FIFO error) prevents the short packet from being reported. The behavior is undefined when this bit is set with output packets or packets outside of queues.<br>0 = Disable<br>1 = Enable |
| 28:27 | **Error Counter (C_ERR).** This field is a 2-bit down counter that keeps track of the number of Errors detected while executing this TD. If this field is programmed with a non-zero value during setup, the Intel® ICH5 decrements the count and writes it back to the TD if the transaction fails. If the counter counts from 1 to 0, the ICH5 marks the TD inactive, sets the "STALLED" and error status bit for the error that caused the transition to 0 in the TD. An interrupt is generated to Host Controller Driver (HCD) if the decrement to 0 was caused by Data Buffer error, Bit stuff error, or if enabled, a CRC or Timeout error. If HCD programs this field to 0 during setup, the ICH5 will not count errors for this TD and there will be no limit on the retries of this TD.<br>**Bits[28:27] — Interrupt After**<br>00 — No Error Limit<br>01 — 1 Error<br>10 — 2 Errors<br>11 — 3 Errors<br>**Error — Decrement Counter**<br>CRC Error — Yes<br>Timeout Error — Yes<br>NAK Received — No<br>Babble Detected — No[1]<br>**Error — Decrement Counter**<br>Data Buffer Error — Yes<br>Stalled — No[1]<br>Bit stuff Error — Yes<br>**NOTE:**<br>1. Detection of Babble or Stall automatically deactivates the TD. Thus, count is not decremented. |
| 26 | **Low Speed Device (LS).** This bit indicates that the target device (USB data source or sink) is a low speed device, running at 1.5 Mb/s, instead of at full speed (12 Mb/sec). There are special restrictions on schedule placement for low speed TDs. If an ICH5 root hub port is connected to a full speed device and this bit is set to a 1 for a low speed transaction, the ICH5 sends out a low speed preamble on that port before sending the PID. No preamble is sent if a ICH5 root hub port is connected to a low speed device.<br>0 = Full Speed Device<br>1 = Low Speed Device |
| 25 | **Isochronous Select (IOS).** The field specifies the type of the data structure. If this bit is set to a 1, then the TD is an isochronous transfer. Isochronous TDs are always marked inactive by the hardware after execution, regardless of the results of the transaction.<br>0 = Non-isochronous Transfer Descriptor<br>1 = Isochronous Transfer Descriptor |
| 24 | **Interrupt on Complete (IOC).** This specifies that the ICH5 should issue an interrupt on completion of the frame in which this Transfer Descriptor is executed. Even if the Active bit in the TD is already cleared when the TD is fetched (no transaction will occur on USB), an IOC interrupt is generated at the end of the frame.<br>1 = Issue IOC |
| 23 | **Active.** For ICH5 schedule execution operations, see Section 5.19.2, Data Transfers to/from Main Memory.<br>0 = When the transaction associated with this descriptor is completed, the ICH5 sets this bit to 0 indicating that the descriptor should not be executed when it is next encountered in the schedule. The Active bit is also set to 0 if a stall handshake is received from the endpoint.<br>1 = Set to 1 by software to enable the execution of a message transaction by the ICH5. |

### Table 83. TD Control and Status (Sheet 2 of 2)

| Bit | Description |
|---|---|
| 22 | **Stalled.**<br>1 = Set to a 1 by the ICH5 during status updates to indicate that a serious error has occurred at the device/endpoint addressed by this TD. This can be caused by babble, the error counter counting down to 0, or reception of the STALL handshake from the device during the transaction. Any time that a transaction results in the Stalled bit being set, the Active bit is also cleared (set to 0). If a STALL handshake is received from a SETUP transaction, a Time Out Error will also be reported. |
| 21 | Data Buffer Error (DBE).<br>1 = Set to a 1 by the ICH5 during status update to indicate that the ICH5 is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (underrun). When this occurs, the actual length and Max Length field of the TD does not match. In the case of an underrun, the ICH5 transmits an incorrect CRC (thus, invalidating the data at the endpoint) and leaves the TD active (unless error count reached 0). If a overrun condition occurs, the ICH5 forces a timeout condition on the USB, invalidating the transaction at the source. |
| 20 | Babble Detected (BABD).<br>1 = Set to a 1 by the ICH5 during status update when “babble” is detected during the transaction generated by this descriptor. Babble is unexpected bus activity for more than a preset amount of time. In addition to setting this bit, the ICH5 also sets the” STALLED” bit (bit 22) to a 1. Since “babble” is considered a fatal error for that transfer, setting the” STALLED” bit to a 1 insures that no more transactions occur as a result of this descriptor. Detection of babble causes immediate termination of the current frame. No further TDs in the frame are executed. Execution resumes with the next frame list index. |
| 19 | Negative Acknowledgment (NAK) Received (NAKR).<br>1 = Set to a 1 by the ICH5 during status update when the ICH5 receives a “NAK” packet during the transaction generated by this descriptor. If a NAK handshake is received from a SETUP transaction, a Time Out Error will also be reported. |
| 18 | **CRC/Time Out Error (CRC_TOUT).**<br>1 = Set to a 1 by the ICH5 as follows:<br>• During a status update in the case that no response is received from the target device/endpoint within the time specified by the protocol chapter of the *Universal Serial Bus Revision 2.0 Specification*.<br>• During a status update when a Cycli Redundancy Check (CRC) error is detected during the transaction associated with this transfer descriptor.<br>In the transmit case (OUT or SETUP Command), this is in response to the ICH5 detecting a timeout from the target device/endpoint.<br>In the receive case (IN Command), this is in response to the ICH5’s CRC checker circuitry detecting an error on the data received from the device/endpoint or a NAK or STALL handshake being received in response to a SETUP transaction. |
| 17 | Bit stuff Error (BSE).<br>1 = This bit is set to a 1 by the ICH5 during status update to indicate that the receive data stream contained a sequence of more than six 1s in a row. |
| 16 | Bus Turn Around Time-out (BTTO).<br>1 = This bit is set to a 1 by the ICH5 during status updates to indicate that a bus time-out condition was detected for this USB transaction. This time-out is specially defined as not detecting an IDLE-to ‘K’ state Start of Packet (SOP) transition from 16 to 18 bit times after the SE0-to IDE transition of previous End of Packet (EOP). |
| 15:11 | Reserved |
| 10:0 | Actual Length (ACTLEN). The Actual Length field is written by the ICH5 at the conclusion of a USB transaction to indicate the actual number of bytes that were transferred. It can be used by the software to maintain data integrity. The value programmed in this register is encoded as n-1 (see Maximum Length field description in the TD Token). |

### Table 84. TD Token

| Bit | Description |
|---|---|
| 31:21 | **Maximum Length (MAXLEN).** The Maximum Length field specifies the maximum number of data bytes allowed for the transfer. The Maximum Length value does not include protocol bytes, such as Packet ID (PID) and CRC. The maximum data packet is 1280 bytes. The 1280 packet length is the longest packet theoretically guaranteed to fit into a frame. Actual packet maximum lengths are set by HCD according to the type and speed of the transfer. Note that the maximum length allowed by the *Universal Serial Bus Revision 2.0 Specification* is 1023 bytes. The valid encodings for this field are:<br>0x000 = 1 byte<br>0x001 = 2 bytes<br>....<br>0x3FE = 1023 bytes<br>0x3FF = 1024 bytes<br>....<br>0x4FF = 1280 bytes<br>0x7FF = 0 bytes (null data packet)<br>Note that values from 500h to 7FEh are illegal and cause a consistency check failure.<br>In the transmit case, the Intel® ICH5 uses this value as a terminal count for the number of bytes it fetches from host memory. In most cases, this is the number of bytes it will actually transmit. In rare cases, the ICH5 may be unable to access memory (e.g., due to excessive latency) in time to avoid underrunning the transmitter. In this instance the ICH5 would transmit fewer bytes than specified in the Maximum Length field. |
| 20 | Reserved. |
| 19 | **Data Toggle (D).** This bit is used to synchronize data transfers between a USB endpoint and the host. This bit determines which data PID is sent or expected (0=DATA0 and 1=DATA1). The Data Toggle bit provides a 1-bit sequence number to check whether the previous packet completed. This bit must always be 0 for Isochronous TDs. |
| 18:15 | **Endpoint (ENDPT).** This 4-bit field extends the addressing internal to a particular device by providing 16 endpoints. This permits more flexible addressing of devices in which more than one sub-channel is required. |
| 14:8 | **Device Address.** This field identifies the specific device serving as the data source or sink. |
| 7:0 | **Packet Identification (PID).** This field contains the Packet ID to be used for this transaction. Only the IN (69h), OUT (E1h), and SETUP (2Dh) tokens are allowed. Any other value in this field causes a consistency check failure resulting in an immediate halt of the ICH5. Bits [3:0] are complements of bits [7:4]. |

### Table 85. TD Buffer Pointer

| Bit | Description |
|---|---|
| 31:0 | **Buffer Pointer (BUFF_PNT).** Bits [31:0] corresponds to memory address [31:0], respectively. It points to the beginning of the buffer that will be used during this transaction. This buffer must be at least as long as the value in the Maximum Length field described int the TD token. The data buffer may be byte-aligned. |

### 5.19.1.3 Queue Head (QH)

Queue heads are special structures used to support the requirements of Control, Bulk, and Interrupt transfers. Since these TDs are not automatically retired after each use, their maintenance requirements can be reduced by putting them into a queue. Queue Heads must be aligned on a 16-byte boundary, and the elements are shown in Table 86.

**Table 86. Queue Head Block**

| Bytes | Description | Attributes |
|---|---|---|
| 00–03 | Queue Head Link Pointer | RO |
| 04–07 | Queue Element Link Pointer | R/W |

**Table 87. Queue Head Link Pointer**

| Bit | Description |
|---|---|
| 31:4 | **Queue Head Link Pointer (QHLP).** This field contains the address of the next data object to be processed in the horizontal list and corresponds to memory address signals [31:4], respectively. |
| 3:2 | Reserved. These bits must be written as 0s. |
| 1 | **QH/TD Select (Q).** This bit indicates to the hardware whether the item referenced by the link pointer is another TD or a QH.<br>0 = TD<br>1 = QH |
| 0 | **Terminate (T).** This bit indicates to the Intel® ICH5 that this is the last QH in the schedule. If there are active TDs in this queue, they are the last to be executed in this frame.<br>0 = Pointer is valid (points to a QH or TD).<br>1 = Last QH (pointer is invalid). |

**Table 88. Queue Element Link Pointer**

| Bit | Description |
|---|---|
| 31:4 | **Queue Element Link Pointer (QELP).** This field contains the address of the next TD or QH to be processed in this queue and corresponds to memory address signals [31:4], respectively. |
| 3:2 | Reserved. |
| 1 | **QH/TD Select (Q).** This bit indicates to the hardware whether the item referenced by the link pointer is another TD or a QH. For entries in a queue, this bit is typically set to 0.<br>0 = TD<br>1 = QH |
| 0 | **Terminate (T).** This bit indicates to the Intel® ICH5 that there are no valid TDs in this queue. When HCD has new queue entries it overwrites this value with a new TD pointer to the queue entry.<br>0 = Pointer is valid.<br>1 = Terminate (No valid queue entries). |

## 5.19.2 Data Transfers to/from Main Memory

The following sections describe the details on how HCD and the ICH5 communicate via the Schedule data structures. The discussion is organized in a top-down manner, beginning with the basics of walking the Frame List, followed by a description of generic processing steps common to all transfer descriptors, and finally a discussion on Transfer Queuing.

### 5.19.2.1 Executing the Schedule

Software programs the ICH5 with the starting address of the Frame List and the Frame List index, then causes the ICH5 to execute the schedule by setting the Run/Stop bit in the Control register to Run. The ICH5 processes the schedule one entry at a time: the next element in the frame list is not fetched until the current element in the frame list is retired.

Schedule execution proceeds in the following fashion:

- The ICH5 first fetches an entry from the Frame List. This entry has three fields. Bit 0 indicates whether the address pointer field is valid. Bit 1 indicates whether the address points to a Transfer Descriptor or to a queue head. The third field is the pointer itself.
- If isochronous traffic is to be moved in a given frame, the Frame List entry points to a Transfer Descriptor. If no isochronous data is to be moved in that frame, the entry points to a queue head or the entry is marked invalid and no transfers are initiated in that frame.
- If the Frame List entry indicates that it points to a Transfer Descriptor, the ICH5 fetches the entry and begins the operations necessary to initiate a transaction on USB. Each TD contains a link field that points to the next entry, as well as indicating whether it is a TD or a QH.
- If the Frame List entry contains a pointer to a QH, the ICH5 processes the information from the QH to determine the address of the next data object that it should process.
- The TD/QH process continues until the millisecond allotted to the current frame expires. At this point, the ICH5 fetches the next entry from the Frame List. If the ICH5 is not able to process all of the transfer descriptors during a given frame, those descriptors are retired by software without having been executed.

### 5.19.2.2 Processing Transfer Descriptors

The ICH5 executes a TD using the following generalized algorithm. These basic steps are common across all modes of TDs. Subsequent sections present processing steps unique to each TD mode.

1. ICH5 fetches TD or QH from the current Link Pointer.
2. If a QH, go to 1 to fetch from the Queue Element Link Pointer. If inactive, go to 12
3. Build token, actual bits are in TD token.
4. If (Host-to-Function) then
   [*Memory Access*] issue request for data, (referenced through TD.BufferPointer)
   wait for first chunk data arrival
   end if
5. [*Begin USB Transaction*] Issue token (from token built in 2, above) and begin data transfer.
   if (Host-to-Function) then Go to 6
   else Go to 7
   end if
6. Fetch data from memory (via TD BufferPointer) and transfer over USB until TD Max-Length bytes have been read and transferred. [*Concurrent system memory and USB Accesses*].
   Go to 8.
7. Wait for data to arrive (from USB). Write incoming bytes into memory beginning at TD BufferPointer. Internal HC buffer should signal end of data packet. Number of bytes received must be (TD Max-Length; The length of the memory area referenced by TD BufferPointer must be (TD Max-Length. [*Concurrent system memory and USB Accesses*].
8. Issue handshake based on status of data received (Ack or Time-out). Go to 10.
9. Wait for handshake, if required [*End of USB Transaction*].
10. Update Status [*Memory Access*] (TD.Status and TD.ActualLength).
    If the TD was an isochronous TD, mark the TD inactive. Go to 12.
    If not an isochronous TD, and the TD completed successfully, mark the TD inactive. Go to 11.
    If not successful, and the error count has not been reached, leave the TD active. If the error count has been reached, mark the TD inactive. Go to 12.
11. Write the link pointer from the current TD into the element pointer field of the QH structure. If the Vf bit is set in the TD link pointer, go to 2.
12. Proceed to next entry.

### 5.19.2.3 Command Register, Status Register, and TD Status Bit Interaction

**Table 89. Command Register, Status Register and TD Status Bit Interaction**

| Condition | Intel® ICH5 USB Status Register Actions | TD Status Register Actions |
|---|---|---|
| CRC/Time Out Error | Set USB Error Int bit[1], Clear HC Halted bit | Clear Active bit[1] and set Stall bit[1] |
| Illegal PID, PID Error, Max Length (illegal) | Clear Run/Stop bit in command register<br>Set HC Process Error and HC Halted bits | |
| PCI Master/Target Abort | Clear Run/Stop bit in command register<br>Set Host System Error and HC Halted bits | |
| Suspend Mode | Clear Run/Stop bit in command register[2]<br>Set HC Halted bit | |
| Resume Received and Suspend Mode = 1 | Set Resume received bit | |
| Run/Stop = 0 | Clear Run/Stop bit in command register<br>Set HC Halted bit | |
| Config Flag Set | Set Config Flag in command register | |
| HC Reset/Global Reset | Clear Run/Stop and Config Flag in command register<br>Clear USB Int, USB Error Int, Resume received, Host System Error, HC Process Error, and HC Halted bits | |
| IOC = 1 in TD Status | Set USB Int bit | |
| Stall | Set USB Error Int bit | Clear Active bit[1] and set Stall bit |
| Bit Stuff/Data Buffer Error | Set USB Error Int bit[1] | Clear Active bit[1] and set Stall bit[1] |
| Short Packet Detect | Set USB Int bit | Clear Active bit |

**NOTES:**
1. Only If error counter counted down from 1 to 0
2. Suspend mode can be entered only when Run/Stop bit is 0

Note that if a NAK or STALL response is received from a SETUP transaction, a Time Out Error is reported. This causes the Error counter to decrement and the CRC/Time-out Error status bit to be set within the TD Control and Status DWord during write back. If the Error counter changes from 1 to 0, the Active bit will be reset to 0 and Stalled bit to 1 as normal.

### 5.19.2.4 Transfer Queuing

Transfer Queues are used to implement a guaranteed data delivery stream to a USB Endpoint. Transfer Queues are composed of two parts: a Queue Header (QH) and a linked list. The linked list of TDs and QHs has an indeterminate length (0 to n).

The QH contains two link pointers and is organized as two contiguous DWords. The first DWord is a horizontal pointer (Queue Head Link Pointer), used to link a single transfer queue with either another transfer queue, or a TD (target data structure depends on Q bit). If the T bit is set, this QH

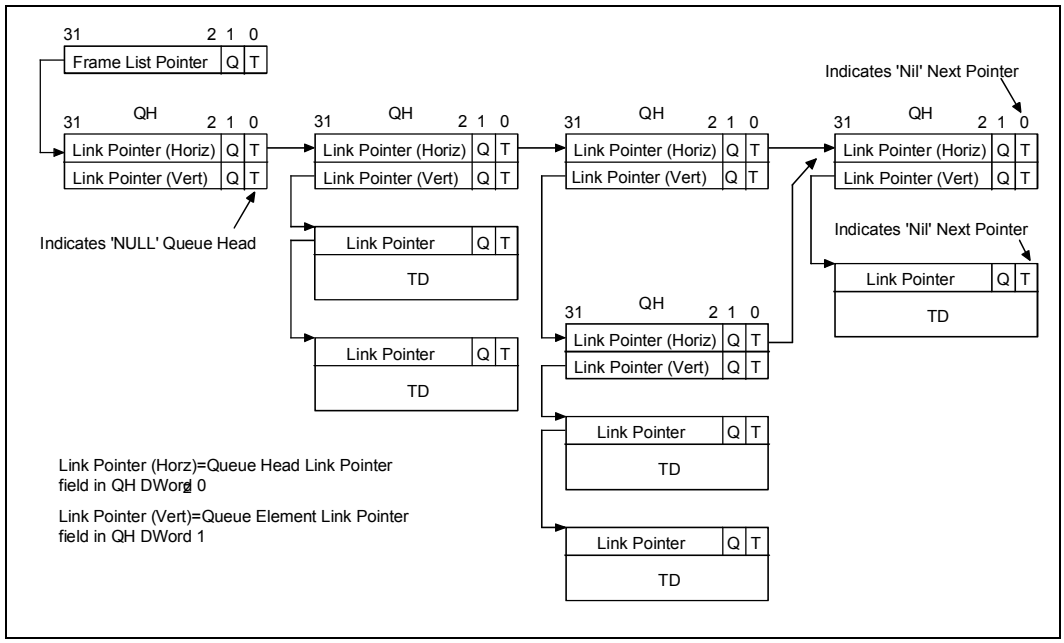
represents the last data structure in the current Frame. The T bit informs the ICH5 that no further processing is required until the beginning of the next frame. The second DWord is a vertical pointer (Queue Element Link Pointer) to the first data structure (TD or QH) being managed by this QH. If the T bit is set, the queue is empty. This pointer may reference a TD or another QH.

Figure 21 illustrates four example queue conditions. The first QH (on far left) is an example of an "empty" queue; the termination bit (T Bit), in the vertical link pointer field, is set to 1. The horizontal link pointer references another QH. The next queue is the expected typical configuration. The horizontal link pointer references another QH, and the vertical link pointer references a valid TD.

Typically, the vertical pointer in a QH points to a TD. However, as shown in Figure 21 (third example from left side of figure) the vertical pointer could point to another QH. When this occurs, a new Q Context is entered and the Q Context just exited is NULL (ICH5 will not update the vertical pointer field).

The far right QH is an example of a frame "termination" node. Since its horizontal link pointer has its termination bit set, the ICH5 assumes there is no more work to complete for the current frame.

**Figure 21. Example Queue Conditions**




Transfer Queues are based on the following characteristics:

- A QH's vertical link pointer (Queue Element Link Pointer) references the "Top" queue member. A QH's horizontal link pointer (Queue Head Link Pointer) references the "next" work element in the Frame.
- Each queue member's link pointer references the next element within the queue.

In the simplest model, the ICH5 follows vertical link point to a queue element, then executes the element. If the completion status of the TD satisfies the advance criteria as shown in Table 90, the ICH5 advances the queue by writing the just-executed TD's link pointer back into the QH's Queue Element link pointer. The next time the queue head is traversed, the next queue element will be the Top element.

The traversal has two options: Breadth first, or Depth first. A flag bit in each TD (Vf — Vertical Traversal Flag) controls whether traversal is Breadth or Depth first. The default mode of traversal is Breadth-First. For Breadth-First, the ICH5 only executes the top element from each queue. The execution path is shown below:

1. QH (Queue Element Link Pointer)
2. TD
3. Write-Back to QH (Queue Element Link Pointer)
4. QH (Queue Head Link pointer).

Breadth-First is also performed for every transaction execution that fails the advance criteria. This means that if a queued TD fails, the queue does not advance, and the ICH5 traverses the QH's Queue Head Link Pointer.

In a depth-first traversal, the top queue element must complete successfully to satisfy the *advance criteria* for the queue. If the ICH5 is currently processing a queue, and the advance criteria are met, and the Vf bit is set, the ICH5 follows the TD's link pointer to the next schedule work item.

Note that regardless of traversal model, when the advance criteria are met, the successful TD's link pointer is written back to the QH's Queue Element link pointer. When the ICH5 encounters a QH, it caches the QH internally, and sets internal state to indicate it is in a Q-context. It needs this state to update the correct QH (for auto advancement) and also to make the correct decisions on how to traverse the frame list.

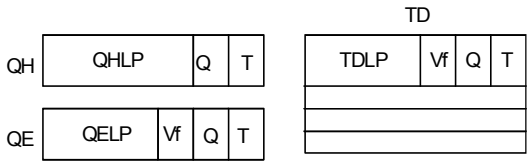
Restricting the advancement of queues to advancement criteria implements a guaranteed data delivery stream. A queue is **never** advanced on an error completion status (even in the event the error count was exhausted).

Table 90 lists the general queue advance criteria, which are based on the execution status of the TD at the "Top" of a currently "active" queue.

### Table 90. Queue Advance Criteria

| Function-to-Host (IN) | | | Host-to-Function (OUT) | | |
|---|---|---|---|---|---|
| **Non-NULL** | **NULL** | **Error/NAK** | **Non-NULL** | **NULL** | **Error/NAK** |
| Advance Q | Advance Q | Retry Q Element | Advance Q | Advance Q | Retry Q Element |

Table 91 is a decision table illustrating the valid combinations of link pointer bits and the valid actions taken when advancement criteria for a queued transfer descriptor are met. The column headings for the link pointer fields are encoded, based on the following list:




Legends:

QH.LP = Queue Head Link Pointer (or Horizontal Link Pointer)
QE.LP = Queue Element Link Pointer (or Vertical Link Pointer)
TD.LP = TD Link Pointer
QH.Q = Q bit in QH
QH.T = T bit in QH

QE.Q = Q bit in QE
QE.T = T bit in QE
TD. Vf = Vf bit in TD
TD.Q = Q bit in TD
TD. T = T bit in TD

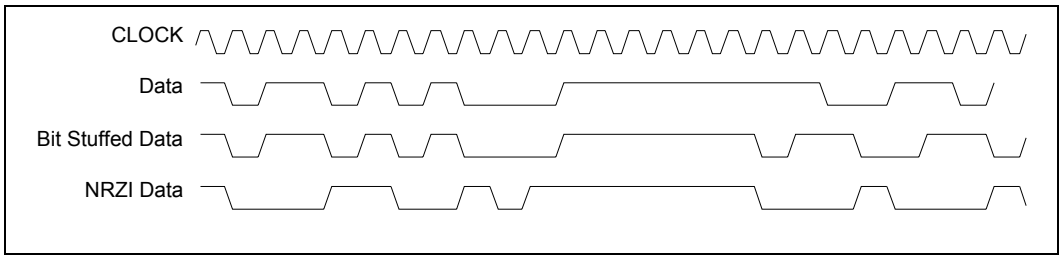
**Table 91. USB Schedule List Traversal Decision Table**

| Q Context | QH.Q | QH.T | QE.Q | QE.T | TD.Vf | TD.Q | TD.T | Description |
|---|---|---|---|---|---|---|---|---|
| 0 | – | – | – | – | x | 0 | 0 | • Not in Queue — execute TD.<br>• Use TD.LP to get next TD |
| 0 | – | – | – | – | x | x | 1 | • Not in Queue — execute TD. End of Frame |
| 0 | – | – | – | – | x | 1 | 0 | • Not in Queue — execute TD.<br>• Use TD.LP to get next (QH+QE).<br>• Set Q Context to 1. |
| 1 | 0 | 0 | 0 | 0 | 0 | x | x | • In Queue. Use QE.LP to get TD.<br>• Execute TD. Update QE.LP with TD.LP.<br>• Use QH.LP to get next TD. |
| 1 | x | x | 0 | 0 | 1 | 0 | 0 | • In Queue. Use QE.LP to get TD.<br>• Execute TD. Update QE.LP with TD.LP.<br>• Use TD.LP to get next TD. |
| 1 | x | x | 0 | 0 | 1 | 1 | 0 | • In Queue. Use QE.LP to get TD.<br>• Execute TD. Update QE.LP with TD.LP.<br>• Use TD.LP to get next (QH+QE). |
| 1 | 0 | 0 | x | 1 | x | x | x | • In Queue. Empty queue.<br>• Use QH.LP to get next TD |
| 1 | x | x | 1 | 0 | – | – | – | • In Queue. Use QE.LP to get (QH+QE) |
| 1 | x | 1 | 0 | 0 | 0 | x | x | • In Queue. Use QE.LP to get TD.<br>• Execute TD. Update QE.LP with TD.LP.<br>• End of Frame |
| 1 | x | 1 | x | 1 | x | x | x | • In Queue. Empty queue. End of Frame |
| 1 | 1 | 0 | 0 | 0 | 0 | x | x | • In Queue. Use QE.LP to get TD.<br>• Execute TD. Update QE.LP with TD.LP.<br>• Use QH.LP to get next (QH+QE). |
| 1 | 1 | 0 | x | 1 | x | x | x | • In Queue. Empty queue.<br>• Use QH.LP to get next (QH+QE) |

## 5.19.3 Data Encoding and Bit Stuffing

The USB employs NRZI data encoding (Non-Return to Zero Inverted) when transmitting packets. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. A string of 0s causes the NRZI data to toggle each bit time. A string of 1s causes long periods with no transitions in the data. In order to ensure adequate signal transitions, bit stuffing is employed by the transmitting device when sending a packet on the USB. A 0 is inserted after every six, consecutive, 1s in the data stream before the data is NRZI encoded to force a transition in the NRZI data stream. This gives the receiver logic a data transition at least once every seven bit times to guarantee the data and clock lock. A waveform of the data encoding is shown in Figure 22.

**Figure 22. USB Data Encoding**




Bit stuffing is enabled beginning with the Sync Pattern and throughout the entire transmission. The data 1 that ends the Sync Pattern is counted as the first one in a sequence. Bit stuffing is always enforced, without exception. If required by the bit stuffing rules, a 0 bit is inserted even if it is the last bit before the end-of-packet (EOP) signal.

## 5.19.4 Bus Protocol

### 5.19.4.1 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.

### 5.19.4.2 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.

### 5.19.4.3 Packet Field Formats

Field formats for the token, data, and handshake packets are described in the following section. The effects of NRZI coding and bit stuffing have been removed for the sake of clarity. All packets have distinct start and end of packet delimiters.

**Table 92. PID Format**

| Bit | Data Sent | Bit | Data Sent |
|---|---|---|---|
| 0 | PID 0 | 4 | NOT(PID 0) |
| 1 | PID 1 | 5 | NOT(PID 1) |
| 2 | PID 2 | 6 | NOT(PID 2) |
| 3 | PID 3 | 7 | NOT(PID 3) |

#### Packet Identifier Field

A packet identifier (PID) immediately follows the SYNC field of every USB packet. A PID consists of a four bit packet type field followed by a four-bit check field as shown in Table 92. The PID indicates the type of packet and, by inference, the format of the packet and the type of error detection applied to the packet. The four-bit check field of the PID insures reliable decoding of the PID so that the remainder of the packet is interpreted correctly. The PID check field is generated by performing a 1s complement of the packet type field.

Any PID received with a failed check field or which decodes to a non-defined value is assumed to be corrupted and the remainder of the packet is assumed to be corrupted and is ignored by the receiver. PID types, codes, and descriptions are listed in Table 93.

**Table 93. PID Types**

| PID Type | PID Name | PID[3:0] | Description |
|---|---|---|---|
| Token | OUT | b0001 | Address + endpoint number in host -> function transaction |
| | IN | b1001 | Address + endpoint number in function -> host transaction |
| | SOF | b0101 | Start of frame marker and frame number |
| | SETUP | b1101 | Address + endpoint number in host -> function transaction for setup to a control endpoint |
| Data | DATA0 | b0011 | Data packet PID even |
| | DATA1 | b1011 | Data packet PID odd |
| Handshake | ACK | b0010 | Receiver accepts error free data packet |
| | NAK | b1010 | Rx device cannot accept data or Tx device cannot send data |
| | STALL | b1110 | Endpoint is stalled |
| Special | PRE | b1100 | Host-issued preamble. Enables downstream bus traffic to low speed devices. |

PIDs are divided into four coding groups: token, data, handshake, and special, with the first two transmitted PID bits (PID[1:0]) indicating which group. This accounts for the distribution of PID codes.

### 5.19.4.4 Address Fields

Function endpoints are addressed using the function address field and the endpoint field.

**Table 94. Address Field**

| Bit | Data Sent |
|---|---|
| 0 | ADDR 0 |
| 1 | ADDR 1 |
| 2 | ADDR 2 |
| 3 | ADDR 3 |

| Bit | Data Sent |
|---|---|
| 4 | ADDR 4 |
| 5 | ADDR 5 |
| 6 | ADDR 6 |
| | |

#### Address Field

The function address (ADDR) field specifies the function, via its address, that is either the source or destination of a data packet, depending on the value of the token PID. As shown in Table 94, a total of 128 addresses are specified as ADDR[6:0]. The ADDR field is specified for IN, SETUP, and OUT tokens.

#### Endpoint Field

An additional four-bit endpoint (ENDP) field, shown in Table 95, permits more flexible addressing of functions in which more than one sub-channel is required. Endpoint numbers are function specific. The endpoint field is defined for IN, SETUP, and OUT token PIDs only.

**Table 95. Endpoint Field**

| Bit | Data Sent |
|---|---|
| 0 | ENDP 0 |
| 1 | ENDP 1 |
| 2 | ENDP 2 |
| 3 | ENDP 3 |

### 5.19.4.5 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 x7FF, and is sent only for SOF tokens at the start of each frame.

### 5.19.4.6 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.

### 5.19.4.7 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. The PID is not included in the CRC check of a packet containing CRC. All CRCs are generated over their respective fields in the transmitter before bit stuffing is performed. Similarly, CRCs are decoded in the receiver after stuffed bits have been removed. Token and data packet CRCs provide 100% coverage for all single and double bit errors. A failed CRC is considered to indicate that one or more of the protected fields is corrupted and causes the receiver to ignore those fields, and, in most cases, the entire packet.

## 5.19.5 Packet Formats

### 5.19.5.1 Token Packets

Table 96 shows the field formats for a token packet. A token consists of a PID, specifying either IN, OUT, or SETUP packet type, and ADDR and ENDP fields. For OUT and SETUP transactions, the address and endpoint fields uniquely identify the endpoint that will receive the subsequent data packet. For IN transactions, these fields uniquely identify which endpoint should transmit a data packet. Only the ICH5 can issue token packets. IN PIDs define a data transaction from a function to the ICH5. OUT and SETUP PIDs define data transactions from the ICH5 to a function.

Token packets have a five-bit CRC which covers the address and endpoint fields as shown above. The CRC does not cover the PID, which has its own check field. Token and SOF packets are delimited by an EOP after three bytes of packet field data. If a packet decodes as an otherwise valid token or SOF but does not terminate with an EOP after three bytes, it must be considered invalid and ignored by the receiver.

**Table 96. Token Format**

| Packet | Width |
|---|---|
| PID | 8 bits |
| ADDR | 7 bits |
| ENDP | 4 bits |
| CRC5 | 5 bits |

### 5.19.5.2 Start of Frame Packets

Table 97 shows a Start Of Frame (SOF) packet. SOF packets are issued by the host at a nominal rate of once every 1.00 ms 0.05. SOF packets consist of a PID indicating packet type followed by an 11-bit frame number field.

The SOF token comprises the token-only transaction that distributes a start of frame marker and accompanying frame number at precisely timed intervals corresponding to the start of each frame. All full speed functions, including hubs, must receive and decode the SOF packet. The SOF token does not cause any receiving function to generate a return packet; therefore, SOF delivery to any given function cannot be guaranteed. The SOF packet delivers two pieces of timing information. A function is informed that a start of frame has occurred when it detects the SOF PID. Frame timing sensitive functions, which do not need to keep track of frame number, need only decode the SOF PID; they can ignore the frame number and its CRC. If a function needs to track frame number, it must comprehend both the PID and the time stamp.

**Table 97. SOF Packet**

| Packet | Width |
|---|---|
| PID | 8 bits |
| Frame Number | 11 bits |
| CRC5 | 5 bits |

### 5.19.5.3 Data Packets

A data packet consists of a PID, a data field, and a CRC as shown in Table 98. There are two types of data packets, identified by differing PIDs: DATA0 and DATA1. Two data packet PIDs are defined to support data toggle synchronization.

Data must always be sent in integral numbers of bytes. The data CRC is computed over only the data field in the packet and does not include the PID, which has its own check field.

**Table 98. Data Packet Format**

| Packet | Width |
|---|---|
| PID | 8 bits |
| DATA | 0–1023 bytes |
| CRC16 | 16 bits |

### 5.19.5.4 Handshake Packets

Handshake packets consist of only a PID. Handshake packets are used to report the status of a data transaction and can return values indicating successful reception of data, flow control, and stall conditions. Only transaction types that support flow control can return handshakes. Handshakes are always returned in the handshake phase of a transaction and may be returned, instead of data, in the data phase. Handshake packets are delimited by an EOP after one byte of packet field. If a packet is decoded as an otherwise valid handshake but does not terminate with an EOP after one byte, it must be considered invalid and ignored by the receiver.

There are three types of handshake packets:

- **ACK** indicates that the data packet was received without bit stuff or CRC errors over the data field and that the data PID was received correctly. An ACK handshake is applicable only in transactions in which data has been transmitted and where a handshake is expected. ACK can be returned by the host for IN transactions and by a function for OUT transactions.
- **NAK** indicates that a function was unable to accept data from the host (OUT) or that a function has no data to transmit to the host (IN). NAK can only be returned by functions in the data phase of IN transactions or the handshake phase of OUT transactions. The host can never issue a NAK. NAK is used for flow control purposes to indicate that a function is temporarily unable to transmit or receive data, but will eventually be able to do so without need of host intervention. NAK is also used by interrupt endpoints to indicate that no interrupt is pending.
- **STALL** is returned by a function in response to an IN token or after the data phase of an OUT. STALL indicates that a function is unable to transmit or receive data, and that the condition requires host intervention to remove the stall. Once a function's endpoint is stalled, the function must continue returning STALL until the condition causing the stall has been cleared through host intervention. The host is not permitted to return a STALL under any condition.

#### 5.19.5.5 Handshake Responses

##### IN Transaction

A function may respond to an IN transaction with a STALL or NAK. If the token received was corrupted, the function issues no response. If the function can transmit data, it issues the data packet. The ICH5, as the USB host, can return only one type of handshake on an IN transaction, an ACK. If it receives a corrupted data, or cannot accept data due to a condition such as an internal buffer overrun, it discards the data and issues no response.

##### OUT Transaction

A function may respond to an OUT transaction with a STALL, ACK, or NAK. If the transaction contained corrupted data, it issues no response.

##### SETUP Transaction

Setup defines a special type of host to function data transaction which permits the host to initialize an endpoint's synchronization bits to those of the host. Upon receiving a Setup transaction, a function must accept the data. Setup transactions cannot be STALLed or NAKed and the receiving function must accept the Setup transfer's data. If a non-control endpoint receives a SETUP PID, it must ignore the transaction and return no response.

### 5.19.6 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 ICH5 operation error. All transaction-based sources can be masked by software through the ICH5's Interrupt Enable register. Additionally, individual transfer descriptors can be marked to generate an interrupt on completion.

When the ICH5 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.

#### 5.19.6.1 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 guarantees 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 the ICH5 to a USB device or a packet transmitted from a USB device to the ICH5 generates a CRC error. The ICH5 is informed of this event by a time-out from the USB device or by the ICH5'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 one. 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 the ICH5 (due to incorrect schedule for instance), the ICH5 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 the ICH5 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.

## 5.19.6.2 Non-Transaction Based Interrupts

If an ICH5 process error or system error occur, the ICH5 halts and immediately issues a hardware interrupt to the system.

### Resume Received

This event indicates that the ICH5 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.

### ICH5 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

The ICH5 sets this bit to 1 when a Parity error, Master Abort, or Target Abort occur. When this error occurs, the ICH5 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.

## 5.19.7 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 the ICH5 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 the ICH5 enters the S3, S4, or S5 states.

**Table 99. 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 the ICH5 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.

## 5.19.8 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. The ICH5 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.

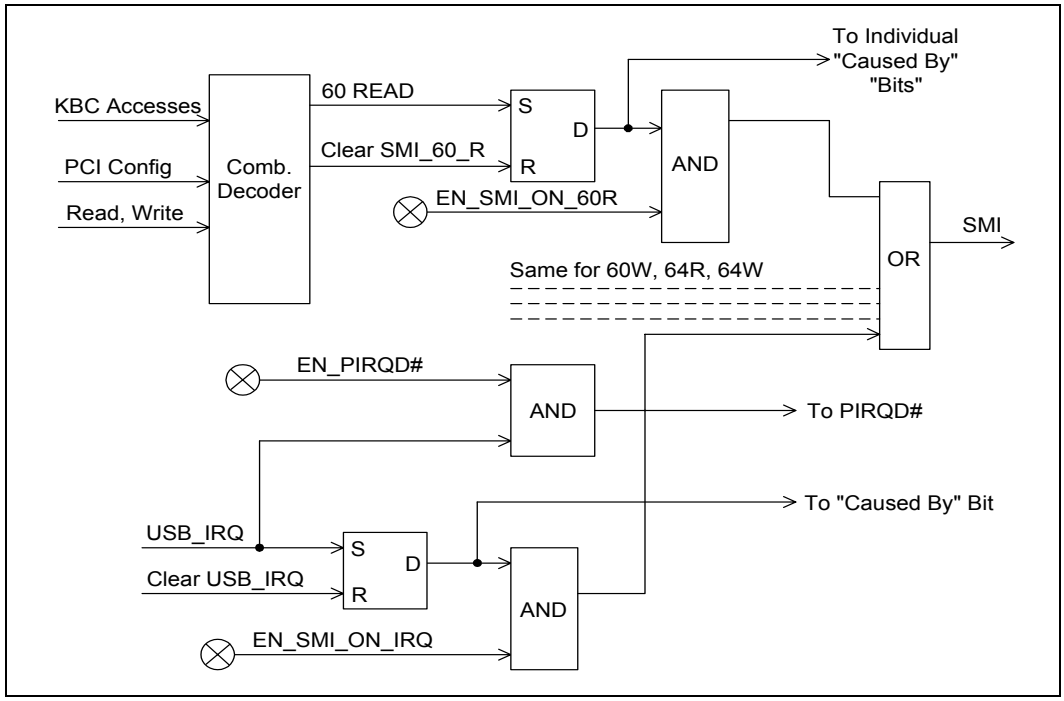
This legacy operation is performed through SMM space. Figure 23 shows the Enable and Status path. The latched SMI source (60R, 60W, 64R, 64W) is available in the Status Register. Because the enable is after the latch, it is possible to check for other events that didn't necessarily cause an SMI. It is 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 100.

**Figure 23. USB Legacy Keyboard Flow Diagram**



### Table 100. USB Legacy Keyboard State Transitions

| 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. |
| 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. |

# 5.20 USB EHCI Host Controller (D29:F7)

The ICH5 contains an Enhanced Host Controller Interface (EHCI) compliant host controller which supports up to eight USB 2.0 high speed compliant root ports. USB 2.0 allows data transfers up to 480 Mbps using the same pins as the eight USB full speed/low speed ports. The ICH5 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 the ICH5.

A summary of the key architectural differences between the USB UHCI host controllers and the EHCI host controller are shown in Table 101.

**Table 101. 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 |

## 5.20.1 EHC Initialization

The following descriptions step through the expected ICH5 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.

### 5.20.1.1 Power On

The suspend well is a “deeper” power plane than the core well, which means that the suspend well is always functional when the core well is functional but the core well may not be functional when the suspend well is. Therefore, the suspend well reset pin (RSMRST#) deasserts before the core well reset pin (PWROK) rises.

1. The suspend well reset deasserts, leaving all registers and logic in the suspend well in the default state. However, it is not possible to read any registers until after the core well reset deasserts. Note that normally the suspend well reset only occurs when a system is unplugged. In other words, suspend well resets are not easily achieved by software or the end-user. This step will typically not occur immediately before the remaining steps.
2. The core well reset deasserts, leaving all registers and logic in the core well in the default state. The EHC configuration space is accessible at this point. Note that the core well reset can (and typically does) occur without the suspend well reset asserting. This means that all of the Configure Flag and Port Status and Control bits (and any other suspend-well logic) may be in any valid state at this time.

### 5.20.1.2 BIOS Initialization

BIOS performs a number of platform customization steps after the core well has powered up. Contact your Intel Field Representative for additional ICH5 BIOS information.

#### 5.20.1.3 Driver Initialization

See Chapter 4 of the *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0.*

#### 5.20.1.4 EHC Resets

In addition to the standard ICH5 hardware resets, portions of the EHC are reset by the HCRESET bit and the transition from the D3hot 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 D3hot (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.

### 5.20.2 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.

### 5.20.3 USB 2.0 Enhanced Host Controller DMA

The ICH5 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 USB 2.0 Based Debug Port), 2) the Periodic DMA engine, and 3) the Asynchronous DMA engine. The ICH5 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.

The following subsections describe the policies of the periodic and asynchronous DMA engines.

#### 5.20.3.1 Periodic List Execution

The Periodic DMA engine contains buffering for two control structures (two transactions). By implementing two entries, the EHC is able to pipeline the memory accesses for the next transaction while executing the current transaction on the USB ports. Note that a multiple-packet, High-Bandwidth transaction occupies one of these buffer entries, which means that up to six, 1-KB data packets may be associated with the two buffered control structures.

#### 5.20.3.1.1 Read Policies for Periodic DMA

The Periodic DMA engine performs reads for the following structures.

| Memory Structure | Size (DWords) | Comments |
|---|---|---|
| Periodic Frame List entry | 1 | The EHC reads the entry for each microframe. The frame list is not internally cached across microframes. |
| iTD | 23 | Only the 64-bit addressing format is supported. |
| siTD | 9 | Only the 64-bit addressing format is supported. |
| qTD | 13 | Only the 64-bit addressing format is supported. |
| Queue Head | 17 | Only the 64-bit addressing format is supported. |
| Out Data | Up to 257 | The Intel® ICH5 breaks large read requests down into smaller aligned read requests based on the setting of the Read Request Max Length field. |
| Frame Span Transversal Node | 2 | |

The EHC Periodic DMA Engine (PDE) does not generate accesses to main memory unless all three of the following conditions are met.

- The HCHalted bit is 0 (memory space, offset 24h, bit 12). Software clears this bit indirectly by setting the RUN/STOP bit to 1.
- The Periodic Schedule Status bit is 1 (memory space, offset 24h, bit 14). Software sets this bit indirectly by setting the Periodic Schedule Enable Bit to 1.
- The Bus Master Enable bit is 1 (configuration space, offset 04h, bit 2).

***Note:*** Prefetching is limited to the current and next microframes only.

***Note:*** Once the PDE checks the length of a periodic packet against the remaining time in the microframe (late-start check) and decides that there is not enough time to run it on the wire, then the EHC switches over to run asynchronous traffic.

#### 5.20.3.1.2 Write Policies for Periodic DMA

The Periodic DMA engine performs writes for the following reasons:

| Memory Structure | Size (DWords) | Comments |
|---|---|---|
| iTD Status Write | 1 | Only the DWord that corresponds to the just-executed microframe's status is written. All bytes of the DWord are written. |
| siTD Status Write | 3 | DWords 0C:17h are written. IOC and Buffer Pointer fields are re-written with the original value. |
| Interrupt Queue Head Overlay | 14 | Only the 64-bit addressing format is supported. DWords 0C:43h are written. |
| Interrupt Queue Head Status Write | 54 | DWords 14:27h are written. |
| Interrupt qTD Status Write | 3 | DWords 04:0Fh are written. PID Code, IOC, Buffer Pointers, and Alt. Next qTD Pointers are re-written with the original value. |
| In Data | Up to 257 | The Intel® ICH5 breaks data writes down into 16 DWord aligned chunks. |

**NOTES:**
1. The Periodic DMA Engine (PDE) will only generate writes after a transaction is executed on USB.
2. Status writes are always performed after In Data writes for the same transaction.

### 5.20.3.2 Asynchronous List Execution

The Asynchronous DMA engine contains buffering for two control structures (two transactions). By implementing two entries, the EHC is able to pipeline the memory accesses for the next transaction while executing the current transaction on the USB ports.

#### 5.20.3.2.1 Read Policies for Asynchronous DMA

The Asynchronous DMA engine performs reads for the following structures.

| Memory Structure | Size (DW) | Comments |
|---|---|---|
| qTD | 13 | Only the 64-bit addressing format is supported. |
| Queue Head | 17 | Only the 64-bit addressing format is supported. |
| Out Data | Up to 129 | The Intel® ICH5 breaks large read requests down in to smaller aligned read requests based on the setting of the Read Request Max Length field. |

The EHC Asynchronous DMA Engine (ADE) does not generate accesses to main memory unless all four of the following conditions are met. (Note that the ADE may be active when the periodic schedule is actively executed, unlike the description in the *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0*; since the EHC contains independent DMA engines, the ADE may perform memory accesses interleaved with the PDE accesses.)

- The HCHalted bit is 0 (memory space, offset 24h, bit 12). Software clears this bit indirectly by setting the RUN/STOP bit to 1.
- The Asynchronous Schedule Status bit is 1 (memory space, offset 24h, bit 14). Software sets this bit indirectly by setting the Asynchronous Schedule Enable Bit to 1.
- The Bus Master Enable bit is 1 (configuration space, offset 04h, bit 2).
- The ADE is not sleeping due to the detection of an empty schedule. There is not one single bit that indicates this state. However, the sleeping state is entered when the Queue Head with the H bit set is encountered when the Reclamation bit in the USB 2.0 Status register is 0.

*Note:* The ADE does not fetch data when a QH is encountered in the Ping state. An Ack handshake in response to the Ping results in the ADE writing the QH to the Out state, which results in the fetching and delivery of the Out Data on the next iteration through the asynchronous list.

*Note:* Once the ADE checks the length of an asynchronous packet against the remaining time in the microframe (late-start check) and decides that there is not enough time to run it on the wire, then the EHC stops all activity on the USB ports for the remainder of that microframe.

*Note:* Once the ADE detects an "empty" asynchronous schedule as described in Section 4 of the *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0*, it implements a waking mechanism like the one in the example. The amount of time that the ADE "sleeps" is 10 µs ± 30 ns.

#### 5.20.3.2.2 Write Policies for Asynchronous DMA

The Asynchronous DMA engine performs writes for the following reasons:

| Memory Structure | Size (DWords) | Comments |
|---|---|---|
| Asynchronous Queue Head Overlay | 14 | Only the 64-bit addressing format is supported. DWords 0C:43h are written. |
| Asynchronous Queue Head Status Write | 34 | DWords 14:1Fh are written. |
| Asynchronous qTD Status Write | 3 | DWords 04:0Fh are written. PID Code, IOC, Buffer Pointer (Page 0), and Alt. Next qTD Pointers are re-written with the original value. |
| In Data | Up to 1297 | The Intel® ICH5 breaks data writes down into 16-DWord aligned chunks. |

**NOTES:**
1. The Asynchronous DMA Engine (ADE) will only generate writes after a transaction is executed on USB.
2. Status writes are always performed after In Data writes for the same transaction.

## 5.20.4 Data Encoding and Bit Stuffing

See Chapter 8 of the *Universal Serial Bus Specification, Revision 2.0.*

## 5.20.5 Packet Formats

See Chapter 8 of the *Universal Serial Bus Specification, Revision 2.0.*

## 5.20.6 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 ICH5-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 never occur on the ICH5.
- 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.
- The ICH5 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 the hub interface before the interrupt is asserted.
- Since the ICH5 supports the 1024-element Frame List size, the Frame List Rollover interrupt occurs every 1024 milliseconds.
- The ICH5 delivers interrupts using PIRQH#.
- The ICH5 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.

### 5.20.6.1 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.

## 5.20.7 USB 2.0 Power Management

### 5.20.7.1 Pause Feature

This feature allows platforms (especially mobile 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 the ICH5. 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).

### 5.20.7.2 Suspend Feature

The *Enhanced Host Controller Interface (EHCI) For Universal Serial Bus Specification*, Section 4.3 describes the details of Port Suspend and Resume.

### 5.20.7.3 ACPI Device States

The USB 2.0 function only supports the D0 and D3 PCI Power Management states. Notes regarding the ICH5 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 never 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.

### 5.20.7.4 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 5.20.7.1) enables dynamic processor low-power states to be entered.

— The PLL in the EHC is disabled when entering the S3/S4/S5 states (core power turns off).

— All core well logic is reset in the S3/S4/S5 states (core power turns off).

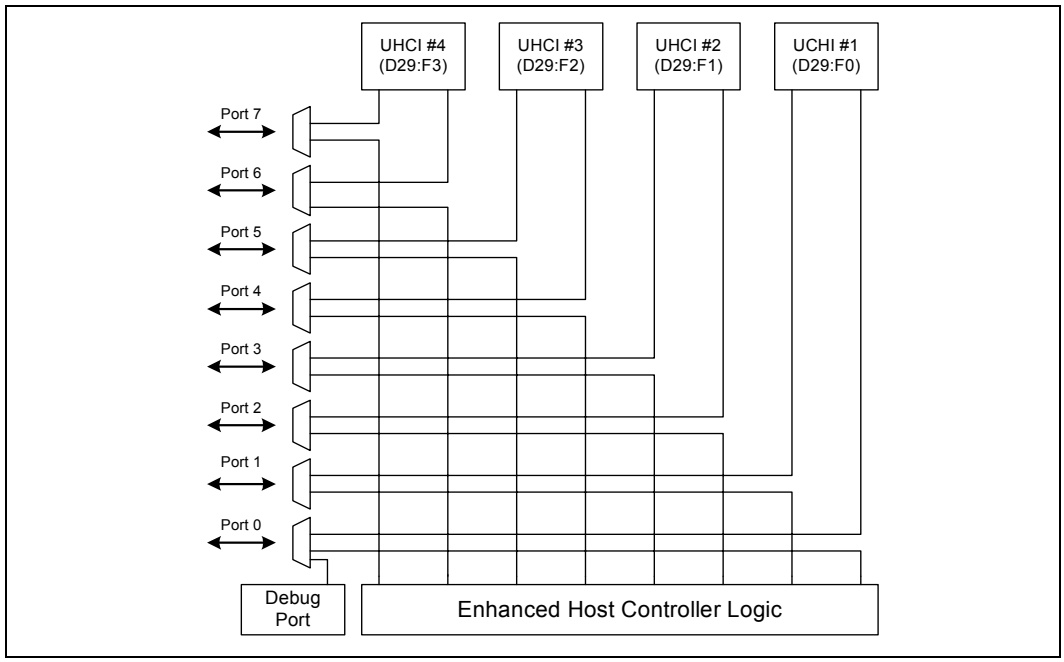
## 5.20.8 Interaction with UHCI Host Controllers

The Enhanced Host controller shares the eight USB ports with four UHCI Host controllers in the ICH5. 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 24 shows the USB Port Connections at a conceptual level.

### 5.20.8.1 Port-Routing Logic

Integrated into the EHC functionality is port-routing logic, which performs the muxing between the UHCI and EHCI host controllers. The ICH5 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 24. Intel® ICH5-USB Port Connections**




Note that the port-routing logic is the only block of logic within the ICH5 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 muxed/demuxed 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.

The ICH5 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.

### 5.20.8.2 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) never sees 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) never sees 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.

### 5.20.8.3 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) never sees 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 never sees a device attached.

### 5.20.8.4 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) |

## 5.20.9 USB 2.0 Legacy Keyboard Operation

The ICH5 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 5.19.8).

The EHC provides the basic ability to generate SMIs on an interrupt event, along with more sophisticated control of the generation of SMIs.

## 5.20.10 USB 2.0 Based Debug Port

The ICH5 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:

- Must be operational before USB 2.0 drivers are loaded.
- Must work even when the port is disabled.
- Must work even though non-configured port is default-routed to the classic controller. 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.
- Must allow 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 to a high-speed port on ICH5 systems.
- Debug Port FIFO must always make 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 OS 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 OS 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

### 5.20.10.1 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 102 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 102. Debug Port Behavior**

| 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 |
| 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 | Illegal. Host controller driver should never put 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. |

#### 5.20.10.1.1 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

#### 5.20.10.1.2 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.

#### 5.20.10.1.3 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., 0000=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 guarantee 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.

## 5.21 SMBus Controller (D31:F3)

The ICH5 provides an SMBus 2.0 compliant 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). The ICH5 is also capable of operating in a mode in which it can communicate with $I^2C$ compatible devices.

The ICH5 can perform SMBus messages with either packet error checking (PEC) enabled or disabled. The actual PEC calculation and checking is performed in hardware by the ICH5.

The Slave Interface allows an external master to read from or write to the ICH5. Write cycles can be used to cause certain events or pass messages, and the read cycles can be used to determine the state of various status bits. The ICH5's internal Host controller cannot access the ICH5's internal Slave Interface.

The ICH5 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. The ICH5 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.

### 5.21.1 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 generate 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 the ICH5's SMB slave port is supported. The ICH5 is fully compliant with 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.

### 5.21.1.1 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 never appended to the Quick Protocol. Software should force the PEC_EN bit to 0 when performing the Quick Command. Software must force the I2C_EN bit to 0 when running this command.The format of the protocol is shown in Table 103.

**Table 103. Quick Protocol**

| Bit | Description |
|---|---|
| 1 | Start Condition |
| 8:2 | Slave Address — 7 bits |
| 9 | Read / Write Direction |
| 10 | Acknowledge from slave |
| 11 | Stop |

#### 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. The format of the protocol is shown in Table 104. and Table 105.

**Table 104. Send / Receive Byte Protocol without PEC**

| Send Byte Protocol | | Receive Byte Protocol | |
|---|---|---|---|
| **Bit** | **Description** | **Bit** | **Description** |
| 1 | Start | 1 | Start |
| 8:2 | Slave Address — 7 bits | 8:2 | Slave Address — 7 bits |
| 9 | Write | 9 | Read |
| 10 | Acknowledge from slave | 10 | Acknowledge from slave |
| 18:11 | Command code — 8 bits | 18:11 | Data byte from slave |
| 19 | Acknowledge from slave | 19 | NOT Acknowledge |
| 20 | Stop | 20 | Stop |

### Table 105. Send/Receive Byte Protocol with PEC

| Send Byte Protocol | | Receive Byte Protocol | |
|---|---|---|---|
| **Bit** | **Description** | **Bit** | **Description** |
| 1 | Start | 1 | Start |
| 8:2 | Slave Address — 7 bits | 8:2 | Slave Address — 7 bits |
| 9 | Write | 9 | Read |
| 10 | Acknowledge from slave | 10 | Acknowledge from slave |
| 18:11 | Command code — 8 bits | 18:11 | Data byte from slave |
| 19 | Acknowledge from slave | 19 | Acknowledge |
| 27:20 | PEC | 27:20 | PEC from slave |
| 28 | Acknowledge from slave | 28 | Not Acknowledge |
| 29 | Stop | 29 | Stop |

## 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. The format of the protocol is shown in Table 106 and Table 107.

### Table 106. Write Byte/Word Protocol without PEC

| Write Byte Protocol | | Write Word Protocol | |
|---|---|---|---|
| **Bit** | **Description** | **Bit** | **Description** |
| 1 | Start | 1 | Start |
| 8:2 | Slave Address — 7 bits | 8:2 | Slave Address — 7 bits |
| 9 | Write | 9 | Write |
| 10 | Acknowledge from slave | 10 | Acknowledge from slave |
| 18:11 | Command code — 8 bits | 18:11 | Command code — 8 bits |
| 19 | Acknowledge from slave | 19 | Acknowledge from slave |
| 27:20 | Data Byte — 8 bits | 27:20 | Data Byte Low — 8 bits |
| 28 | Acknowledge from Slave | 28 | Acknowledge from Slave |
| 29 | Stop | 36:29 | Data Byte High — 8 bits |
| | | 37 | Acknowledge from slave |
| | | 38 | Stop |

### Table 107. Write Byte/Word Protocol with PEC

| Write Byte Protocol | | Write Word Protocol | |
|---|---|---|---|
| **Bit** | **Description** | **Bit** | **Description** |
| 1 | Start | 1 | Start |
| 8:2 | Slave Address — 7 bits | 8:2 | Slave Address — 7 bits |
| 9 | Write | 9 | Write |
| 10 | Acknowledge from slave | 10 | Acknowledge from slave |
| 18:11 | Command code — 8 bits | 18:11 | Command code — 8 bits |
| 19 | Acknowledge from slave | 19 | Acknowledge from slave |
| 27:20 | Data Byte — 8 bits | 27:20 | Data Byte Low — 8 bits |
| 28 | Acknowledge from Slave | 28 | Acknowledge from Slave |
| 36:29 | PEC | 36:29 | Data Byte High — 8 bits |
| 37 | Acknowledge from Slave | 37 | Acknowledge from slave |
| 38 | Stop | 45:38 | PEC |
| | | 46 | Acknowledge from slave |
| | | 47 | Stop |

## Read Byte/Word

Reading data is slightly more complicated than writing data. First the ICH5 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. The format of the protocol is shown in Table 108 and Table 109.

### Table 108. Read Byte/Word Protocol without PEC

| Read Byte Protocol | | Read Word Protocol | |
|---|---|---|---|
| **Bit** | **Description** | **Bit** | **Description** |
| 1 | Start | 1 | Start |
| 8:2 | Slave Address — 7 bits | 82 | Slave Address — 7 bits |
| 9 | Write | 9 | Write |
| 10 | Acknowledge from slave | 10 | Acknowledge from slave |
| 18:11 | Command code — 8 bits | 18:11 | Command code — 8 bits |
| 19 | Acknowledge from slave | 19 | Acknowledge from slave |
| 20 | Repeated Start | 20 | Repeated Start |
| 27:21 | Slave Address — 7 bits | 27:21 | Slave Address — 7 bits |
| 28 | Read | 28 | Read |
| 29 | Acknowledge from slave | 29 | Acknowledge from slave |
| 37:30 | Data from slave — 8 bits | 37:30 | Data Byte Low from slave — 8 bits |
| 38 | NOT acknowledge | 38 | Acknowledge |
| 39 | Stop | 46:39 | Data Byte High from slave — 8 bits |
| | | 47 | NOT acknowledge |
| | | 48 | Stop |

### Table 109. Read Byte/Word Protocol with PEC

| Read Byte Protocol | | Read Word Protocol | |
|---|---|---|---|
| **Bit** | **Description** | **Bit** | **Description** |
| 1 | Start | 1 | Start |
| 8:2 | Slave Address — 7 bits | 8:2 | Slave Address — 7 bits |
| 9 | Write | 9 | Write |
| 10 | Acknowledge from slave | 10 | Acknowledge from slave |
| 18:11 | Command code — 8 bits | 18:11 | Command code — 8 bits |
| 19 | Acknowledge from slave | 19 | Acknowledge from slave |
| 20 | Repeated Start | 20 | Repeated Start |
| 27:21 | Slave Address — 7 bits | 27:21 | Slave Address — 7 bits |
| 28 | Read | 28 | Read |
| 29 | Acknowledge from slave | 29 | Acknowledge from slave |
| 37:30 | Data from slave — 8 bits | 37:30 | Data Byte Low from slave — 8 bits |
| 38 | Acknowledge | 38 | Acknowledge |
| 46:39 | PEC from slave | 46:39 | Data Byte High from slave — 8 bits |
| 47 | NOT Acknowledge | 47 | Acknowledge |
| 48 | Stop | 55:48 | PEC from slave |
| | | 56 | NOT acknowledge |
| | | 57 | Stop |

### Process Call

The process call is so named because a command sends data and waits for the slave to return a value dependent on that data. The protocol is simply a Write Word followed by a Read Word, but without a second command or stop condition.

When programmed for the Process Call command, the ICH5 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. The format of the protocol is shown in Table 110 and Table 111.

***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.

**Table 110. Process Call Protocol without PEC**

| Bit | Description |
|---|---|
| 1 | Start |
| 8:2 | Slave Address — 7 bits |
| 9 | Write |
| 10 | Acknowledge from Slave |
| 18:11 | Command code — 8 bits (Skip this step if I2C_EN bit is set) |
| 19 | Acknowledge from slave (Skip this step if I2C_EN bit is set) |
| 27:20 | Data byte Low — 8 bits |
| 28 | Acknowledge from slave |
| 36:29 | Data Byte High — 8 bits |
| 37 | Acknowledge from slave |
| 38 | Repeated Start |
| 45:39 | Slave Address — 7 bits |
| 46 | Read |
| 47 | Acknowledge from slave |
| 55:48 | Data Byte Low from slave — 8 bits |
| 56 | Acknowledge |
| 64:57 | Data Byte High from slave — 8 bits |
| 65 | NOT acknowledge |
| 66 | Stop |

**Table 111. Process Call Protocol with PEC**

| Bit | Description |
|---|---|
| 1 | Start |
| 8:2 | Slave Address — 7 bits |
| 9 | Write |
| 10 | Acknowledge from Slave |
| 18:11 | Command code — 8 bits |
| 19 | Acknowledge from slave |
| 27:20 | Data byte Low — 8 bits |
| 28 | Acknowledge from slave |
| 36:29 | Data Byte High — 8 bits |
| 37 | Acknowledge from slave |
| 38 | Repeated Start |
| 45:39 | Slave Address — 7 bits |
| 46 | Read |
| 47 | Acknowledge from slave |
| 55:48 | Data Byte Low from slave — 8 bits |
| 56 | Acknowledge |
| 64:57 | Data Byte High from slave — 8 bits |
| 65 | Acknowledge |
| 73:66 | PEC from slave |
| 74 | NOT acknowledge |
| 75 | Stop |

### Block Read/Write

The ICH5 contains a 32-byte buffer for read and write data which can be enabled by setting bit 1 of the Auxiliary Control register at offset 0Dh in I/O space, as opposed to a single byte of buffering. This 32-byte buffer is filled with write data before transmission, and filled with read data on reception. In the ICH5, the interrupt is generated only after a transmission or reception of 32 bytes, or when the entire byte count has been transmitted/received.

This requires the ICH5 to check the byte count field. Currently, the byte count field is transmitted but ignored by the hardware 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.

SMBus mode: The block write begins with a slave address and a write condition. After the command code the ICH5 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. The format of the Block Read/Write protocol is shown in Table 112 and Table 113.

***Note:*** For Block Write, if the $I^2C$_EN bit is set, the format of the command changes slightly. The ICH5 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.

**Table 112. Block Read/Write Protocol without PEC**

| Block Write Protocol | | Block Read Protocol | |
|---|---|---|---|
| **Bit** | **Description** | **Bit** | **Description** |
| 1 | Start | 1 | Start |
| 8:2 | Slave Address — 7 bits | 8:2 | Slave Address — 7 bits |
| 9 | Write | 9 | Write |
| 10 | Acknowledge from slave | 10 | Acknowledge from slave |
| 18:11 | Command code — 8 bits | 18:11 | Command code — 8 bits |
| 19 | Acknowledge from slave | 19 | Acknowledge from slave |
| 27:20 | Byte Count — 8 bits<br>(Skip this step if $I^2C$_En bit set) | 20 | Repeated Start |
| 28 | Acknowledge from Slave<br>(Skip this step if I2C_EN bit set) | 27:21 | Slave Address — 7 bits |
| 36:29 | Data Byte 1 — 8 bits | 28 | Read |
| 37 | Acknowledge from Slave | 29 | Acknowledge from slave |
| 45:3 | Data Byte 2 — 8 bits | 37:30 | Byte Count from slave — 8 bits |
| 46 | Acknowledge from slave | 38 | Acknowledge |
| ... | Data Bytes / Slave<br>Acknowledges... | 46:39 | Data Byte 1 from slave — 8 bits |
| ... | Data Byte N — 8 bits | 47 | Acknowledge |
| ... | Acknowledge from Slave | 55:48 | Data Byte 2 from slave — 8 bits |
| ... | Stop | 56 | Acknowledge |
| | | ... | Data Bytes from slave/Acknowledge |
| | | ... | Data Byte N from slave — 8 bits |
| | | ... | NOT Acknowledge |
| | | ... | Stop |

### Table 113. Block Read/Write Protocol with PEC

| Block Write Protocol | | Block Read Protocol | |
|---|---|---|---|
| **Bit** | **Description** | **Bit** | **Description** |
| 1 | Start | 1 | Start |
| 8:2 | Slave Address — 7 bits | 8:2 | Slave Address — 7 bits |
| 9 | Write | 9 | Write |
| 10 | Acknowledge from slave | 10 | Acknowledge from slave |
| 18:11 | Command code — 8 bits | 18:11 | Command code — 8 bits |
| 19 | Acknowledge from slave | 19 | Acknowledge from slave |
| 27:20 | Byte Count — 8 bits | 20 | Repeated Start |
| 28 | Acknowledge from Slave | 27:21 | Slave Address — 7 bits |
| 36:29 | Data Byte 1 — 8 bits | 28 | Read |
| 37 | Acknowledge from Slave | 29 | Acknowledge from slave |
| 45:38 | Data Byte 2 — 8 bits | 37:30 | Byte Count from slave — 8 bits |
| 46 | Acknowledge from slave | 38 | Acknowledge |
| ... | Data Bytes / Slave Acknowledges... | 46:39 | Data Byte 1 from slave — 8 bits |
| ... | Data Byte N — 8 bits | 47 | Acknowledge |
| ... | Acknowledge from Slave | 55:48 | Data Byte 2 from slave — 8 bits |
| ... | PEC — 8 bits | 56 | Acknowledge |
| ... | Acknowledge from Slave | ... | Data Bytes from slave/Acknowledge |
| ... | Stop | ... | Data Byte N from slave — 8 bits |
| | | ... | Acknowledge |
| | | ... | PEC from slave — 8 bits |
| | | ... | NOT Acknowledge |
| | | ... | Stop |

### I²C Read

This command allows the ICH5 to perform block reads to certain I²C devices, such as serial E²PROMs. The SMBus Block Read supports the 7-bit addressing mode only.

However, this does not allow access to devices using the I²C "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 I²C 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 I²C 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 new command is shown in Table 114.

**Table 114. I²C 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 |

The ICH5 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 32byte 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.

**Table 115. Block Write–Block Read Process Call Protocol with/without PEC (Sheet 1 of 2)**

| Bit | Description |
|---|---|
| 1 | Start |
| 8:2 | Slave Address — 7 bits |
| 9 | Write |
| 10 | Acknowledge from slave |
| 18:11 | Command code — 8 bits |
| 19 | Acknowledge from slave |
| 27:20 | Data Byte Count (M) — 8 bits |
| 28 | Acknowledge from slave |
| 36:29 | Data Byte (1) — 8 bits |
| 37 | Acknowledge from slave |
| 45:38 | Data Byte (2) — 8 bits |
| 46 | Acknowledge from slave |
| – | – |
| | Data Byte (M) — 8 bits |
| | Acknowledge from slave |

**Table 115. Block Write–Block Read Process Call Protocol with/without PEC (Sheet 2 of 2)**

| Bit | Description |
|---|---|
| | Repeated Start |
| | Slave Address — 7 bits |
| | Read |
| | Acknowledge from master |
| | Data Byte Count (N) from master — 8 bits |
| | Acknowledge from slave |
| | Data Byte (1) from master — 8 bits |
| | Acknowledge from slave |
| | Data Byte (2) from master — 8 bits |
| | Acknowledge from slave |
| – | – |
| | Data Byte Count (N) from master — 8 bits |
| | Acknowledge from slave |
| | Data Byte High from slave - 8 bits |
| | Acknowledge from slave (Skip if no PEC) |
| | PEC from master (Skip if no PEC) |
| | NOT acknowledge |
| | Stop |

### 5.21.1.2 $I^2C$ Behavior

When the $I^2C$_EN bit is set, the ICH5 SMBus logic will instead be set to communicate with $I^2C$ devices. This forces the following changes:

- The Process Call command will skip the Command code (and its associated acknowledge)
- The Block Write command will skip sending the Byte Count (DATA0)

In addition, the ICH5 will support the new $I^2C$ Read command. This is independent of the $I^2C$_EN bit.

***Note:*** When operating in $I^2C$ mode the ICH5 will not use the 32-byte buffer for block commands.

## 5.21.2 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. The ICH5 must continuously monitor the SMBDATA line. When the ICH5 is attempting to drive the bus to a 1 by letting go of the SMBDATA line, and it samples SMBDATA low, then some other master is driving the bus and the ICH5 must stop transferring data.

If the ICH5 sees that it has lost arbitration, the condition is called a collision. The ICH5 will set the BUS_ERR bit in the Host Status Register, and if enabled, generate an interrupt or SMI#. The processor is responsible for restarting the transaction.

When the ICH5 is a SMBus master, it drives the clock. When the ICH5 is sending address or command as an SMBus master, or data bytes as a master on writes, it drives data relative to the clock it is also driving. It will not start toggling the clock until the start or stop condition meets proper setup and hold time. The ICH5 will also guarantee minimum time between SMBus transactions as a master.

***Note:*** The ICH5 supports the same arbitration protocol for both the SMBus and the System Management (SMLINK) interfaces.

## 5.21.3 Bus Timing

### 5.21.3.1 Clock Stretching

Some devices may not be able to handle their clock toggling at the rate that the ICH5 as an SMBus master would like. They have the capability of stretching the low time of the clock. When the ICH5 attempts to release the clock (allowing the clock to go high), the clock will remain low for an extended period of time.

The ICH5 must monitor 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.

### 5.21.3.2 Bus Time Out (Intel® ICH5 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. The ICH5 will discard the cycle, and set the DEV_ERR bit. The time out minimum is 25 ms. The time-out counter inside the ICH5 will start after the last bit of data is transferred by the ICH5 and it is waiting for a response. The 25 ms will be a count of 800 RTC clocks.

## 5.21.4 Interrupts / SMI#

The ICH5 SMBus controller uses PIRQB# as its interrupt pin. However, the system can alternatively be set up to generate SMI# instead of an interrupt, by setting the SMBUS_SMI_EN bit.

Table 117 and Table 118 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 116. 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 117. 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. 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 118. 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) |

### 5.21.5 SMBALERT#

SMBALERT# is multiplexed with GPIO11. When enable and the signal is asserted, The ICH5 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 GPIO or not), causes the event message to be sent in heartbeat mode.

### 5.21.6 SMBus CRC Generation and Checking

If the AAC bit is set in the Auxiliary Control register, the ICH5 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.

### 5.21.7 SMBus Slave Interface

The ICH5'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 the ICH5 to decode cycles, and allows an external microcontroller to perform specific actions. Key features and capabilities include:

- Supports decode of three types of messages: Byte Write, Byte Read, and Host Notify.
- Receive Slave Address register: This is the address that the ICH5 decodes. A default value is provided so that the slave interface can be used without the processor having to program this register.
- Receive Slave Data register in the SMBus I/O space that includes the data written by the external microcontroller.
- Registers that the external microcontroller can read to get the state of the ICH5. See Table 123.
- 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 9.10.9) for all others

If a master leaves the clock and data bits of the SMBus interface at 1 for 50 µs or more in the middle of a cycle, the ICH5 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 micro controller accesses the SMBus Slave Interface over the SMBus a translation in the address is needed to accommodate the least significant bit used for read/write control. For example, if the ICH5 slave address (RCV_SLVA) is left at 44h (default), the external micro controller would use an address of 88h/89h (write/read).

### 5.21.7.1 Format of Slave Write Cycle

The external master performs Byte Write commands to the ICH5 SMBus Slave I/F. The "Command" field (bits 11:18) indicate which register is being accessed. The Data field (bits 20:27) indicate the value that should be written to that register.

The Write Cycle format is shown in Table 119. Table 120 has the values associated with the registers.

**Table 119. Slave Write Cycle Format**

| Bits | Description | Driven By | Comment |
|---|---|---|---|
| 1 | Start Condition | 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 | ICH5 | |
| 18:11 | Command | External Microcontroller | This field indicates which register will be accessed. See Table 120 for the register definitions |
| 19 | ACK | ICH5 | |
| 27:20 | Register Data | External Microcontroller | See Table 120 for the register definitions |
| 28 | ACK | ICH5 | |
| 29 | Stop | External Microcontroller | |

**Table 120. Slave Write Registers**

| Register | Function |
|---|---|
| 0 | Command Register. See Table 121 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 | Frequency Straps will be written on bits 3:0. Bits 7:4 should be 0, but will be ignored. |
| 9–FFh | Reserved |

**NOTE:** The external microcontroller is responsible to make sure that it does not update the contents of the data byte registers until they have been read by the system processor. The ICH5 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. ICH5 will not attempt to cover this race condition (i.e., unpredictable results in this case).

### Table 121. 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 the Intel® ICH5 from sending Heartbeat and Event messages (as described in Section 5.14.2). 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 ICH5 detects this command type while in the S0 state, it sets the SMLINK_SLV_SMI_STS bit (see Section 9.11.7). This command should only be used if the system is in an S0 state. If the message is received during S1–S5 states, the ICH5 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 |

### 5.21.7.2 Format of Read Command

The external master performs Byte Read commands to the ICH5 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 122 shows the Read Cycle format. Table 123 shows the register mapping for the data byte.

**Table 122. 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 | Intel® ICH5 | |
| 18:11 | Command code – 8 bits | External Microcontroller | Indicates which register is being accessed. See Table 123. |
| 19 | ACK | ICH5 | |
| 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 | ICH5 | |
| 37:30 | Data Byte | ICH5 | Value depends on register being accessed. See Table 123. |
| 38 | NOT ACK | External Microcontroller | |
| 39 | Stop | External Microcontroller | |

### Table 123. 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 |
| 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 the Intel® ICH5'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, which indicates that the flash BIOS is probably blank. |
| 5 | 1 | Reserved |
| 5 | 2 | **CPU Power Failure Status.** 1 if the CPUPWR_FLR bit in the GEN_PMCON_2 register is set. |
| 5 | 7:3 | Reserved |
| 6 | 7:0 | Contents of the Message 1 register. See Section 9.11.9. |
| 7 | 7:0 | Contents of the Message 2 register. See Section 9.11.9. |
| 8 | 7:0 | Contents of the WDSTATUS register. See Section 9.11.10. |
| 9–FFh | 7:0 | Reserved |

#### 5.21.7.2.1 Behavioral Notes

According to SMBus protocol, Read and Write messages always begin with a Start bit – Address–Write bit sequence. When the ICH5 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 (See Table 119 and Table 122). In other words, if a Start –Address–Read occurs (which is illegal for SMBus Read or Write protocol), and the address matches the ICH5's Slave Address, the ICH5 will still grab the cycle.

Also according to SMBus protocol, a Read cycle contains a Repeated Start–Address–Read sequence beginning at bit 20 (See Table 122). Once again, if the Address matches the ICH5's Receive Slave Address, it will assume that the protocol is followed, ignore bit 28, and proceed with the Slave Read cycle.

***Note:*** An external microcontroller must not attempt to access the ICH5's SMBus Slave logic until at least 1 second after both RTCRST# and RSMRST# are deasserted (high).

### 5.21.7.3 Format of Host Notify Command

The ICH5 tracks and responds to the standard Host Notify command as specified in the *System Management Bus (SMBus) Specification, Version 2.0*. The host address for this command is fixed to 0001000b. If the ICH5 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 124 shows the Host Notify format.

**Table 124. Host Notify Format**

| Bit | Description | Driven By | Comment |
|---|---|---|---|
| 1 | Start | External Master | |
| 8:2 | SMB Host Address — 7 bits | External Master | Always 0001_000 |
| 9 | Write | External Master | Always 0 |
| 10 | ACK (or NACK) | Intel® ICH5 | ICH5 NACKs if HOST_NOTIFY_STS is 1 |
| 17:11 | Device Address – 7 bits | External Master | Indicates the address of the master; loaded into the Notify Device Address Register |
| 18 | Unused — Always 0 | External Master | 7-bit-only address; this bit is inserted to complete the byte |
| 19 | ACK | ICH5 | |
| 27:20 | Data Byte Low — 8 bits | External Master | Loaded into the Notify Data Low Byte Register |
| 28 | ACK | ICH5 | |
| 36:29 | Data Byte High — 8 bits | External Master | Loaded into the Notify Data High Byte Register |
| 37 | ACK | ICH5 | |
| 38 | Stop | External Master | |

## 5.22 AC '97 Controller (Audio D31:F5, Modem D31:F6)

***Note:*** All references to AC '97 in this document refer to the *AC '97 Specification*, Version 2.3. For further information on the operation of the AC-link protocol, see the *AC '97 Specification*, *Version 2.3*.

The ICH5 AC '97 controller features include:

- Independent PCI functions for audio and modem.
- Independent bus master logic for dual Microphone input, dual PCM Audio input (2-channel stereo per input), PCM audio output (2-, 4- or 6-channel audio), Modem input, Modem output and S/PDIF output.
- 20-bit sample resolution
- Multiple sample rates up to 48 kHz
- 16 GPIOs
- Single modem line
- Configure up to three codecs with three AC_SDIN pins

Table 125 shows a detailed list of features supported by the ICH5 AC '97 digital controller

**Table 125. Features Supported by Intel® ICH5 (Sheet 1 of 2)**

| Feature | Description |
|---|---|
| System Interface | • Isochronous low latency bus master memory interface<br>• Scatter/gather support for word-aligned buffers in memory (all mono or stereo 20-bit and 16-bit data types are supported, no 8-bit data types are supported)<br>• Data buffer size in system memory from 3 to 65535 samples per input<br>• Data buffer size in system memory from 0 to 65535 samples per output<br>• Independent PCI audio and modem functions with configuration and I/O spaces<br>• AC '97 codec registers are shadowed in system memory via driver<br>• AC '97 codec register accesses are serialized via semaphore bit in PCI I/O space (new accesses are not allowed while a prior access is still in progress) |
| Power Management | • Power management via PCI Power Management |
| PCI Audio Function | • Read/write access to audio codec registers 00h–3Ah and vendor registers 5Ah–7Eh<br>• 20-bit stereo PCM output, up to 48 kHz (L,R, Center, Sub-woofer, L-rear and R-rear channels on slots 3,4,6,7,8,9,10,11)<br>• 16-bit stereo PCM input, up to 48 kHz (L,R channels on slots 3,4)<br>• 16-bit mono mic in w/ or w/o mono mix, up to 48 kHz (L,R channel, slots 3,4) (mono mix supports mono hardware AEC reference for speakerphone)<br>• 16-bit mono PCM input, up to 48 kHz from dedicated mic ADC (slot 6) (supports speech recognition or stereo hardware AEC ref for speakerphone)<br>• During cold reset AC_RST# is held low until after POST and software deassertion of AC_RST# (supports passive PC_BEEP to speaker connection during POST) |

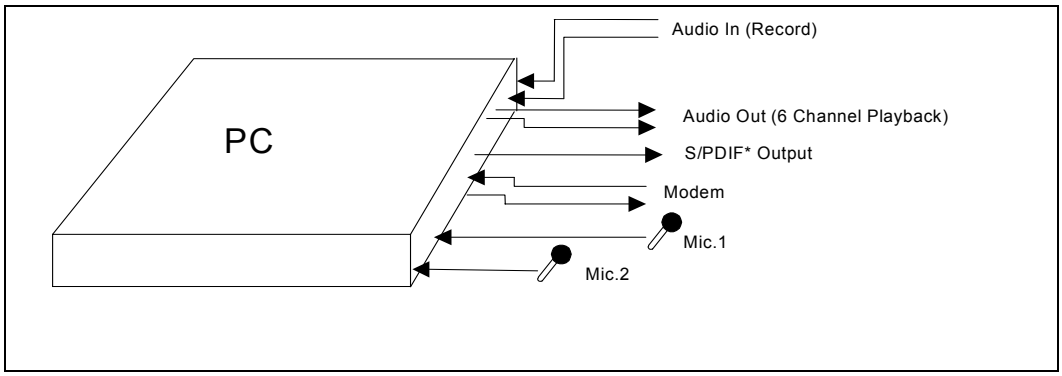
### Table 125. Features Supported by Intel® ICH5 (Sheet 2 of 2)

| Feature | Description |
|---|---|
| PCI Modem function | • Read/write access to modem codec registers 3Ch–58h and vendor registers 5Ah–7Eh<br>• 16-bit mono modem line1 output and input, up to 48 kHz (slot 5)<br>• Low latency GPIO[15:0] via hardwired update between slot 12 and PCI I/O register<br>• Programmable PCI interrupt on modem GPIO input changes via slot 12 GPIO_INT<br>• SCI event generation on AC_SDIN[2:0] wake-up signal |
| AC-link | • AC '97 2.3 AC-link interface<br>• Variable sample rate output support via AC '97 SLOTREQ protocol (slots 3,4,5,6,7,8,9,10,11)<br>• Variable sample rate input support via monitoring of slot valid tag bits (slots 3,4,5,6)<br>• 3.3 V digital operation meets AC '97 2.3 DC switching levels<br>• AC-link I/O driver capability meets AC '97 2.3 triple codec specifications<br>• Codec register status reads must be returned with data in the next AC-link frame, per *AC '97 v2.3 Specification*. |
| Multiple Codec | • Triple codec addressing: All AC '97 Audio codec register accesses are addressable to codec ID 00 (primary), codec ID 01 (secondary), or codec ID 10 (tertiary).<br>• Modem codec addressing: All AC '97 Modem codec register accesses are addressable to codec ID 00 (primary) or codec ID 01 (secondary).<br>• Triple codec receive capability via AC_SDIN[2:0] pins (AC_SDIN[2:0] frames are internally validated, synchronized, and OR'd depending on the Steer Enable bit status in the SDM register)<br>• AC_SDIN mapping to DMA engine mapping capability allows for simultaneous input from two different audio codecs.<br>**NOTES:**<br>1. Audio Codec IDs are remappable and not limited to 00,01,10.<br>2. Modem Codec IDs are remappable and limited to 00, 01.<br>3. When using multiple codecs, the Modem Codec must be ID 01. |

***Note:*** Throughout this document, references to D31:F5 indicate that the audio function exists in PCI Device 31, Function 5. References to D31:F6 indicate that the modem function exists in PCI Device 31, Function 6.

***Note:*** Throughout this document references to tertiary, third, or triple codecs refer to the third codec in the system connected to the AC_SDIN2 pin. The *AC '97 v2.3 Specification* refers to non-primary codecs as multiple secondary codecs. To avoid confusion and excess verbiage, this datasheet refers to it as the third or tertiary codec.

### Figure 25. Intel® ICH5-Based Audio Codec '97 Specification, Version 2.3



## 5.22.1 PCI Power Management

This Power Management section applies for all AC ’97 controller functions. After a power management event is detected, the AC ’97 controller wakes the host system. The following sections describe these events and the AC ’97 controller power states.

### Device Power States

The AC ’97 controller supports D0 and D3 PCI Power Management states. The following are notes regarding the AC ’97 controller implementation of the Device States:

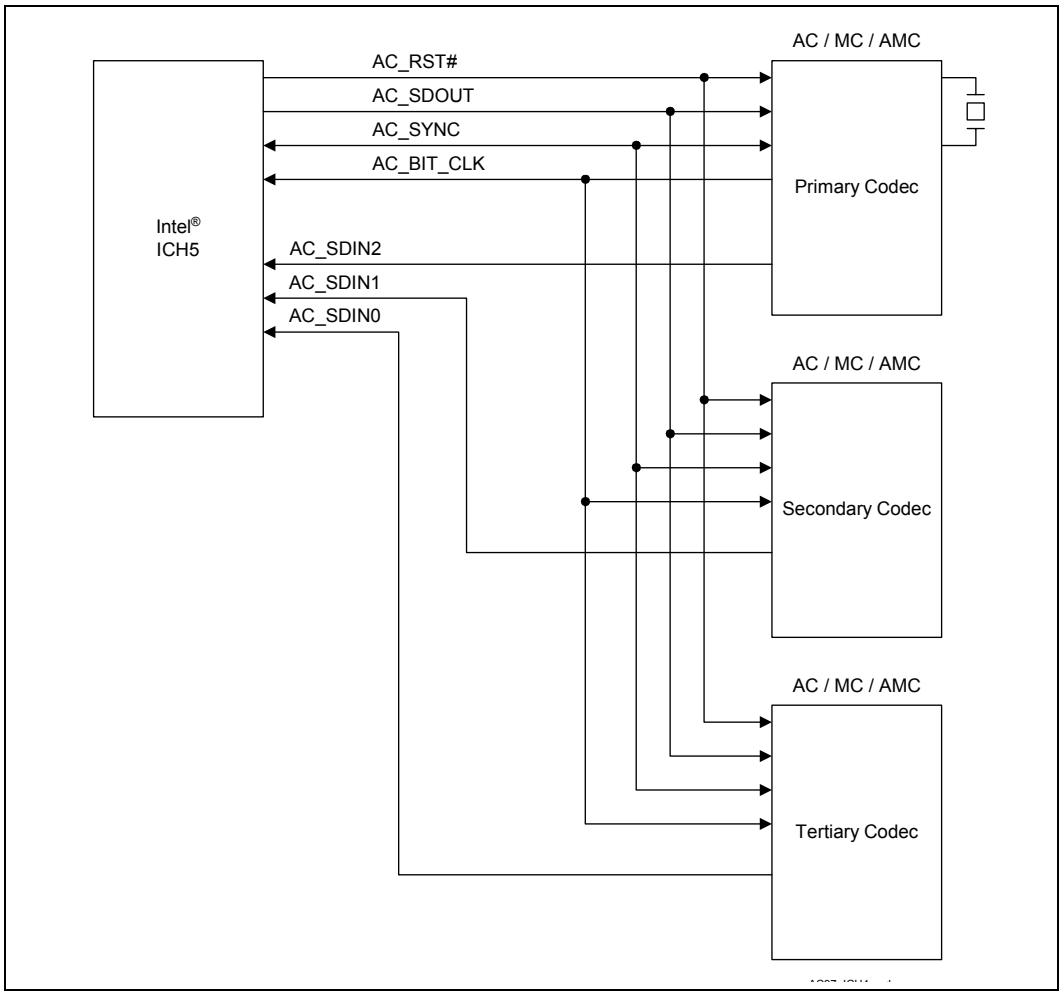
1. The AC ’97 controller hardware does not inherently consume any more power when it is in the D0 state than it does in D3 state. However, software can halt the DMA engine prior to entering these low power states such that the maximum power consumption is reduced.
2. In the D0 state, all implemented AC ’97 controller features are enabled.
3. In D3 state, accesses to the AC ’97 controller memory-mapped or I/O range results in master abort.
4. In D3 state, the AC ’97 controller interrupt must never assert for any reason. The internal PME# signal is used to signal wake events, etc.
5. When the Device Power State field is written from $D3_{HOT}$ to D0, an internal reset is generated. See Section 16.1 for general rules on the effects of this reset.
6. AC97 STS bit is set only when the audio or modem resume events were detected and their respective PME enable bits were set.
7. GPIO Status change interrupt no longer has a direct path to the AC97 STS bit. This causes a wake up event only if the modem controller was in D3
8. Resume events on AC_SDIN[2:0] cause resume interrupt status bits to be set only if their respective controllers are not in D3.
9. Edge detect logic prevents the interrupts from being asserted in case the AC97 controller is switched from D3 to D0 after a wake event.
10. Once the interrupt status bits are set, they will cause PIRQB# if their respective enable bits were set. One of the audio or the modem drivers will handle the interrupt.

## 5.22.2 AC-Link Overview

The ICH5 is an AC ’97 2.3 controller that communicates with companion codecs via a digital serial link called the AC-link. All digital audio/modem streams and command/status information is communicated over the AC-link.

The AC-link is a bi-directional, serial PCM digital stream. It handles multiple input and output data streams, as well as control register accesses, employing a time division multiplexed (TDM) scheme. The AC-link architecture provides for data transfer through individual frames transmitted in a serial fashion. Each frame is divided into 12 outgoing and 12 incoming data streams, or slots. The architecture of the ICH5 AC-link allows a maximum of three codecs to be connected. Figure 24 shows a three codec topology of the AC-link for the ICH5.

### Figure 26. AC '97 2.3 Controller-Codec Connection




The AC-link consists of a five signal interface between the controller and codec. Table 129 indicates the AC-link signal pins on the ICH5 and their associated power wells.

### Table 126. AC '97 Signals

| Signal Name | Type | Power Well | Description |
|---|---|---|---|
| AC_RST# | Output | Resume | Master hardware reset |
| AC_SYNC | Output | Core | 48 kHz fixed rate sample sync |
| AC_BIT_CLK | Input | Core | 12.288 MHz Serial data clock |
| AC_SDOUT | Output | Core | Serial output data |
| AC_SDIN0 | Input | Resume | Serial input data |
| AC_SDIN1 | Input | Resume | Serial input data |
| AC_SDIN2 | Input | Resume | Serial input data |

**NOTE:** Power well voltage levels are 3.3 V

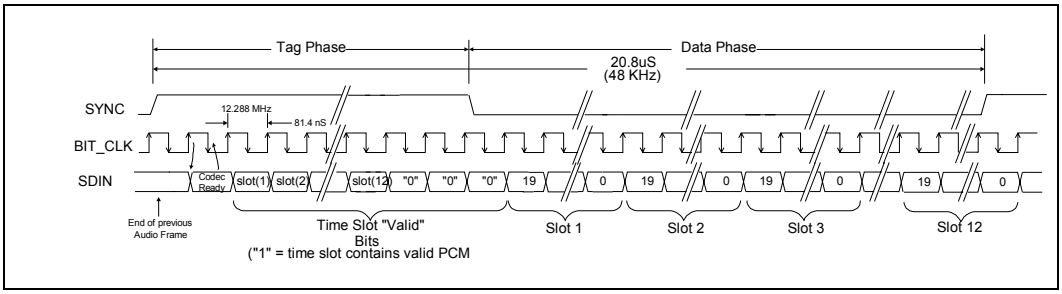
ICH5 core well outputs may be used as strapping options for the ICH5, sampled during system reset. These signals may have weak pullups/pulldowns; however, this will not interfere with link operation. ICH5 inputs integrate weak pulldowns to prevent floating traces when a secondary and/or tertiary codec is not attached. When the Shut Off bit in the control register is set, all buffers will be turned off and the pins will be held in a steady state, based on these pullups/pulldowns.

AC_BIT_CLK is fixed at 12.288 MHz and is sourced by the primary codec. It provides the necessary clocking to support the twelve 20-bit time slots. AC-link serial data is transitioned on each rising edge of AC_BIT_CLK. The receiver of AC-link data samples each serial bit on the falling edge of AC_BIT_CLK.

If AC_BIT_CLK makes no transitions for four consecutive PCI clocks, the ICH5 assumes the primary codec is not present or not working. It sets bit 28 of the Global Status Register (I/O offset 30h). All accesses to codec registers with this bit set will return data of FFh to prevent system hangs.

Synchronization of all AC-link data transactions is signaled by the AC '97 controller via the AC_SYNC signal, as shown in Figure 25. The primary codec drives the serial bit clock onto the AC-link, which the AC '97 controller then qualifies with the AC_SYNC signal to construct data frames. AC_SYNC, fixed at 48 kHz, is derived by dividing down AC_BIT_CLK. AC_SYNC remains high for a total duration of 16 AC_BIT_CLKs at the beginning of each frame. The portion of the frame where AC_SYNC is high is defined as the tag phase. The remainder of the frame where AC_SYNC is low is defined as the data phase. Each data bit is sampled on the falling edge of AC_BIT_CLK.

**Figure 27. AC-Link Protocol**




The ICH5 has three AC_SDIN pins allowing a single, dual, or triple codec configuration. When multiple codecs are connected, the primary, secondary, and tertiary codecs can be connected to any AC_SDIN line. The ICH5 does not distinguish between codecs on its AC_SDIN[2:0] pins, however the registers do distinguish between AC_SDIN0, AC_SDIN1, and AC_SDIN2 for wake events, etc. If using a Modem Codec it is recommended to connect it to AC_SDIN1.

See your Platform Design Guide for a matrix of valid codec configurations. The ICH5 does not support optional test modes as outlined in the *AC '97 Specification, Version 2.3*.

#### 5.22.2.1 AC-Link Output Frame (SDOUT)

A new output frame begins with a low to high transition of AC_SYNC. AC_SYNC is synchronous to the rising edge of AC_BIT_CLK. On the immediately following falling edge of AC_BIT_CLK, the codec samples the assertion of AC_SYNC. This falling edge marks the time when both sides of AC-link are aware of the start of a new frame. On the next rising edge of AC_BIT_CLK, the ICH5 transitions AC_SDOUT into the first bit position of slot 0, or the valid frame bit. Each new bit position is presented to the AC-link on a rising edge of AC_BIT_CLK, and subsequently sampled by the codec on the following falling edge of AC_BIT_CLK. This sequence ensures that data transitions and subsequent sample points for both incoming and outgoing data streams are time aligned.

The output frame data phase corresponds to the multiplexed bundles of all digital output data targeting codec DAC inputs and control registers. Each output frame supports up to twelve outgoing data time slots. The ICH5 generates 16 or 20 bits and stuffs remaining bits with 0s.

The output data stream is sent with the most significant bit first, and all invalid slots are stuffed with 0s. When mono audio sample streams are output from the ICH5, software must ensure both left and right sample stream time slots are filled with the same data.

#### 5.22.2.2 Output Slot 0: Tag Phase

Slot 0 is considered the tag phase. The tag phase is a special,16-bit time slot wherein each bit conveys a valid tag for its corresponding time slot within the current frame. A 1 in a given bit position of slot 0, indicates that the corresponding time slot within the current frame has been assigned to a data stream and contains valid data. If a slot is tagged invalid with a 0 in the corresponding bit position of slot 0, the ICH5 stuffs the corresponding slot with 0s during that slot's active time.

Within slot 0, the first bit is a valid frame bit (slot 0, bit 15) which flags the validity of the entire frame. If the valid frame bit is set to 1, this indicates that the current frame contains at least one slot with valid data. When there is no transaction in progress, the ICH5 deasserts the frame valid bit. Note that after a write to slot 12, that slot will always stay valid, and therefore the frame valid bit remains set.

The next 12 bit positions of slot 0 (bits [14:3]) indicate which of the corresponding twelve time slots contain valid data. Bits [1:0] of slot 0 are used as codec ID bits to distinguish between separate codecs on the link.

Using the valid bits in the tag phase allows data streams of differing sample rates to be transmitted across the link at its fixed 48 kHz frame rate. The codec can control the output sample rate of the ICH5 using the SLOTREQ bits as described in the *AC '97 v2.3 Specification*

#### 5.22.2.3 Output Slot 1: Command Address Port

The command port is used to control features and monitor status of AC '97 functions including, but not limited to, mixer settings and power management. The control interface architecture supports up to 64, 16-bit read/write registers, addressable on even byte boundaries. Only the even registers (00h, 02h, etc.) are valid. Output frame slot 1 communicates control register address, and write/ read command information.

In the case of the multiple codec implementation, accesses to the codecs are differentiated by the driver using address offsets 00h–7Fh for the primary codec, address offsets 80h–FEh for the secondary codec, and address offsets 100h–17Fh for the tertiary codec. The differentiation on the link, however, is done via the codec ID bits. See Section 6.20.2.23 for further details.

#### 5.22.2.4 Output Slot 2: Command Data Port

The command data port is used to deliver 16-bit control register write data in the event that the current command port operation is a write cycle as indicated in slot 1, bit 19. If the current command port operation is a read, the entire slot time stuffed with 0s by the ICH5. Bits [19:4] contain the write data. Bits [3:0] are reserved and are stuffed with 0s.

#### 5.22.2.5 Output Slot 3: PCM Playback Left Channel

Output frame slot 3 is the composite digital audio left playback stream. Typically, this slot is composed of standard PCM (.wav) output samples digitally mixed by the host processor. The ICH5 transmits sample streams of 16 bits or 20 bits and stuffs remaining bits with 0s.

Data in output slots 3 and 4 from the ICH5 should be duplicated by software if there is only a single channel out.

#### 5.22.2.6 Output Slot 4: PCM Playback Right Channel

Output frame slot 4 is the composite digital audio right playback stream. Typically, this slot is composed of standard PCM (.wav) output samples digitally mixed by the host processor. The ICH5 transmits sample streams of 16 or 20 bits and stuffs remaining bits with 0s.

Data in output slots 3 and 4 from the ICH5 should be duplicated by software if there is only a single channel out.

#### 5.22.2.7 Output Slot 5: Modem Codec

Output frame slot 5 contains modem DAC data. The modem DAC output supports 16-bit resolution. At boot time, if the modem codec is supported, the AC ’97 controller driver determines the DAC resolution. During normal runtime operation the ICH5 stuffs trailing bit positions within this time slot with 0s.

#### 5.22.2.8 Output Slot 6: PCM Playback Center Front Channel

When set up for 6-channel mode, this slot is used for the front center channel. The format is the same as Slots 3 and 4. If not set up for 6-channel mode, this channel is always stuffed with 0s by ICH5.

#### 5.22.2.9 Output Slots 7–8: PCM Playback Left and Right Rear Channels

When set up for 4 or 6 channel modes, slots 7 and 8 are used for the rear Left and Right channels. The format for these two channels are the same as Slots 3 and 4.

### 5.22.2.10 Output Slot 9: Playback Sub Woofer Channel

When set for 6-channel mode, this slot is used for the Sub Woofer. The format is the same as Slot 3. If not set up for 6-channel mode, this channel is always stuffed with 0s by ICH5.

### 5.22.2.11 Output Slots 10–11: Reserved

Output frame slots 10–11 are reserved and are always stuffed with 0s by the ICH5 AC ’97 controller.

### 5.22.2.12 Output Slot 12: I/O Control

The 16 bits of DAA and GPIO control (output) and status (input) have been directly assigned to bits on slot 12 to minimize latency of access to changing conditions.

The value of the bits in this slot are the values written to the GPIO control register at offset 54h and D4h (in the case of a secondary codec) in the modem codec I/O space. The following rules govern the usage of slot 12.

1. Slot 12 is marked invalid by default on coming out of AC-link reset, and remains invalid until a register write to 54h/D4h.
2. A write to offset 54h/D4h in codec I/O space causes the write data to be transmitted on slot 12 in the next frame, with slot 12 marked valid, and the address/data information to also be transmitted on slots 1 and 2.
3. After the first write to offset 54h/D4h, slot 12 remains valid for all following frames. The data transmitted on slot 12 is the data last written to offset 54h/D4h. Any subsequent write to the register causes the new data to be sent out on the next frame.
4. Slot 12 gets invalidated after the following events: PCI reset, AC ’97 cold reset, warm reset, and hence a wake from S3, S4, or S5. Slot 12 remains invalid until the next write to offset 54h/D4h.

### 5.22.2.13 AC-Link Input Frame (SDIN)

There are three AC_SDIN lines on the ICH5 for use with up to three codecs. Each AC_SDIN pin can have a codec attached. The input frame data streams correspond to the multiplexed bundles of all digital input data targeting the AC ’97 controller. As in the case for the output frame, each AC-link input frame consists of twelve time slots.

A new audio input frame begins with a low to high transition of AC_SYNC. AC_SYNC is synchronous to the rising edge of AC_BIT_CLK. On the immediately following falling edge of AC_BIT_CLK, the receiver samples the assertion of AC_SYNC. This falling edge marks the time when both sides of AC-link are aware of the start of a new audio frame. On the next rising edge of AC_BIT_CLK, the codec transitions AC_SDIN into the first bit position of slot 0 (codec ready bit). Each new bit position is presented to AC-link on a rising edge of AC_BIT_CLK, and subsequently sampled by the ICH5 on the following falling edge of AC_BIT_CLK. This sequence ensures that data transitions and subsequent sample points for both incoming and outgoing data streams are time aligned.

AC_SDIN data stream must follow the *AC ’97 v2.3 Specification* and be MSB justified with all non-valid bit positions (for assigned and/or unassigned time slots) stuffed with 0s. AC_SDIN data is sampled by the ICH5 on the falling edge of AC_BIT_CLK.

### 5.22.2.14 Input Slot 0: Tag Phase

Input slot 0 consists of a codec ready bit (bit 15), and slot valid bits for each subsequent slot in the frame (bits [14:3]).

The codec ready bit within slot 0 (bit 15) indicates whether the codec on the AC-link is ready for register access (digital domain). If the codec ready bit in slot 0 is a 0, the codec is not ready for register access. When the AC-link codec ready bit is a 1, it indicates that the AC-link and codec control and status registers are in a fully operational state. The codec ready bits are visible through the Global Status register of the ICH5. Software must further probe the Powerdown Control/Status register in the codec to determine exactly which subsections, if any, are ready.

Bits [14:3] in slot 0 indicate which slots of the input stream to the ICH5 contain valid data, just as in the output frame. The remaining bits in this slot are stuffed with 0s.

### 5.22.2.15 Input Slot 1: Status Address Port / Slot Request Bits

The status port is used to monitor status of codec functions including, but not limited to, mixer settings and power management.

Slot 1 must echo the control register index, for historical reference, for the data to be returned in slot 2, assuming that slots 1 and 2 had been tagged valid by the codec in slot 0.

For variable sample rate output, the codec examines its sample rate control registers, the state of its FIFOs, and the incoming SDOUT tag bits at the beginning of each audio output frame to determine which SLOTREQ bits to set active (low). SLOTREQ bits asserted during the current audio input frame signal which output slots require data from the controller in the next audio output frame. For fixed 48 kHz operation the SLOTREQ bits are always set active (low) and a sample is transferred each frame.

For variable sample rate input, the tag bit for each input slot indicates whether valid data is present or not.

**Table 127. Input Slot 1 Bit Definitions**

| Bit | Description |
|---|---|
| 19 | Reserved (Set to 0) |
| 18:12 | Control Register Index (Stuffed with 0s if tagged invalid) |
| 11 | Slot 3 Request: PCM Left Channel(1) |
| 10 | Slot 4 Request: PCM Right Channel(1) |
| 9 | Slot 5 Request: Modem Line 1 |
| 8 | Slot 6 Request: PCM Center Channel(1) |
| 7 | Slot 7 Request: PCM Left Surround(1) |
| 6 | Slot 8 Request: PCM Right Surround(1) |
| 5 | Slot 9 Request: PCM LFE Channel(1) |
| 4:2 | Slot Request 10–12: Not Implemented |
| 1:0 | Reserved (Stuffed with 0s) |

**NOTES:**
1. Slot 3 Request and Slot 4 Request bits must be the same value, i.e. set or cleared in tandem. This is also true for the Slot 7 and Slot 8 Request bits, as well as the Slot 6 and Slot 9 Request bits.

As shown in Table 127, slot 1 delivers codec control register read address and multiple sample rate slot request flags for all output slots of the controller. When a slot request bit is set by the codec, the controller returns data in that slot in the next output frame. Slot request bits for slots 3 and 4 are always set or cleared in tandem (i.e., both are set or cleared).

When set, the input slot 1 tag bit only pertains to Status Address Port data from a previous read. SLOTREQ bits are always valid independent of the slot 1 tag bit.

### 5.22.2.16 Input Slot 2: Status Data Port

The status data port receives 16-bit control register read data.

Bit [19:4]: Control Register Read Data

Bit [3:0]: Reserved.

### 5.22.2.17 Input Slot 3: PCM Record Left Channel

Input slot 3 is the left channel input of the codec. The ICH5 supports 16-bit sample resolution. Samples transmitted to the ICH5 must be in left/right channel order.

### 5.22.2.18 Input Slot 4: PCM Record Right Channel

Input slot 4 is the right channel input of the codec. The ICH5 supports 16-bit sample resolution. Samples transmitted to the ICH5 must be in left/right channel order.

### 5.22.2.19 Input Slot 5: Modem Line

Input slot 5 contains MSB justified modem data. The ICH5 supports 16-bit sample resolution.

#### 5.22.2.20 Input Slot 6: Optional Dedicated Microphone Record Data

Input slot 6 is a third PCM system input channel available for dedicated use by a microphone. This input channel supplements a true stereo output which enables more precise echo cancellation algorithm for speakerphone applications. The ICH5 supports 16-bit resolution for slot 6 input.

#### 5.22.2.21 Input Slots 7–11: Reserved

Input frame slots 7–11 are reserved for future use and should be stuffed with 0s by the codec, per the *AC '97 Specification, Version 2.3*.

#### 5.22.2.22 Input Slot 12: I/O Status

The status of the GPIOs configured as inputs are to be returned on this slot in every frame. The data returned on the latest frame is accessible to software by reading the register at offset 54h/D4h in the codec I/O space. Only the 16 MSBs are used to return GPI status. In order for GPI events to cause an interrupt, both the 'sticky' and 'interrupt' bits must be set for that particular GPIO pin in regs 50h and 52h. Therefore, the interrupt will be signalled until it has been cleared by the controller, which can be much longer than one frame.

Reads from 54h/D4h are not transmitted across the link in slot 1 and 2. The data from the most recent slot 12 is returned on reads from offset 54h/D4h.

#### 5.22.2.23 Register Access

In the ICH5 implementation of the AC-link, up to three codecs can be connected to the SDOUT pin. The following mechanism is used to address the primary, secondary, and tertiary codecs individually.

The primary device uses bit 19 of slot 1 as the direction bit to specify read or write. Bits [18:12] of slot 1 are used for the register index. For I/O writes to the primary codec, the valid bits [14:13] for slots 1 and 2 must be set in slot 0, as shown in Table 128. Slot 1 is used to transmit the register address, and slot 2 is used to transmit data. For I/O reads to the primary codec, only slot 1 should be valid since only an address is transmitted. For I/O reads only slot 1 valid bit is set, while for I/O writes both slots 1 and 2 valid bits are set.

The secondary and tertiary codec registers are accessed using slots 1 and 2 as described above, however the slot valid bits for slots 1 and 2 are marked invalid in slot 0 and the codec ID bits [1:0] (bit 0 and bit 1 of slot 0) is set to a non-zero value. This allows the secondary or tertiary codec to monitor the slot valid bits of slots 1and 2, and bits [1:0] of slot 0 to determine if the access is directed to the secondary or tertiary codec. If the register access is targeted to the secondary or tertiary codec, slot 1 and 2 will contain the address and data for the register access. Since slots 1 and 2 are marked invalid, the primary codec will ignore these accesses.

**Table 128. Output Tag Slot 0**

| Bit | Primary Access Example | Secondary Access Example | Description |
|---|---|---|---|
| 15 | 1 | 1 | Frame Valid |
| 14 | 1 | 0 | Slot 1 Valid, Command Address bit (Primary codec only) |
| 13 | 1 | 0 | Slot 2 Valid, Command Data bit (Primary codec only) |
| 12:3 | X | X | Slot 3–12 Valid |
| 2 | 0 | 0 | Reserved |
| 1:0 | 00 | 01 | Codec ID (00 reserved for primary; 01 indicate secondary; 10 indicate tertiary) |

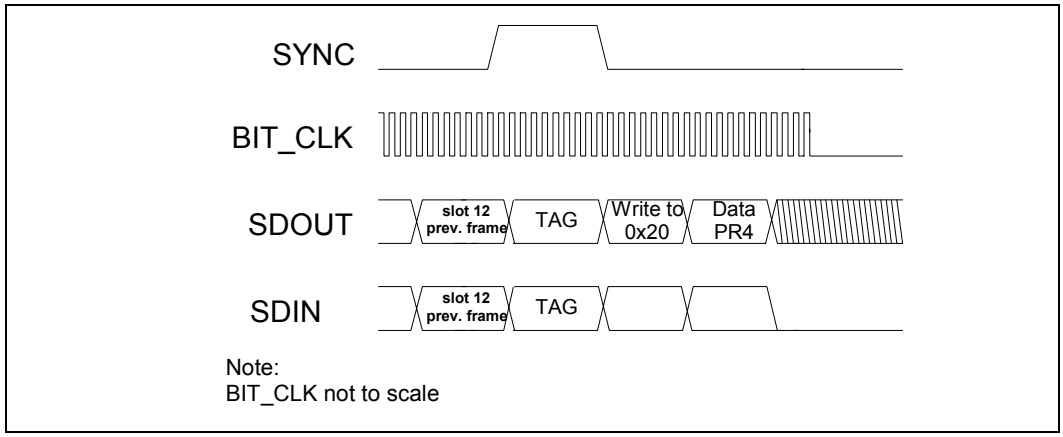
When accessing the codec registers, only one I/O cycle can be pending across the AC-link at any time. The ICH5 implements write posting on I/O writes across the AC-link (i.e., writes across the link are indicated as complete before they are actually sent across the link). In order to prevent a second I/O write from occurring before the first one is complete, software must monitor the CAS bit in the Codec Access Semaphore register which indicates that a codec access is pending. Once the CAS bit is cleared, then another codec access (read or write) can go through. The exception to this being reads to offset 54h/D4h/154h (slot 12) which are returned immediately with the most recently received slot 12 data. Writes to offset 54h, D4h, and 154h (primary, secondary and tertiary codecs), get transmitted across the AC-link in slots 1 and 2 as a normal register access. Slot 12 is also updated immediately to reflect the data being written.

The controller does not issue back to back reads. It must get a response to the first read before issuing a second. In addition, codec reads and writes are only executed once across the link, and are not repeated.

## 5.22.3 AC-Link Low Power Mode

The AC-link signals can be placed in a low-power mode. When the AC ’97 Powerdown register (26h), is programmed to the appropriate value, both AC_BIT_CLK and AC_SDIN will be brought to, and held at a logic low voltage level.

**Figure 28. AC-Link Powerdown Timing**



AC_BIT_CLK and AC_SDIN transition low immediately after a write to the Powerdown Register (26h) with PR4 enabled. When the AC ’97 controller driver is at the point where it is ready to program the AC-link into its low-power mode, slots 1 and 2 are assumed to be the only valid stream in the audio output frame.

The AC ’97 controller also drives AC_SYNC, and AC_SDOUT low after programming AC ’97 to this low power, halted mode

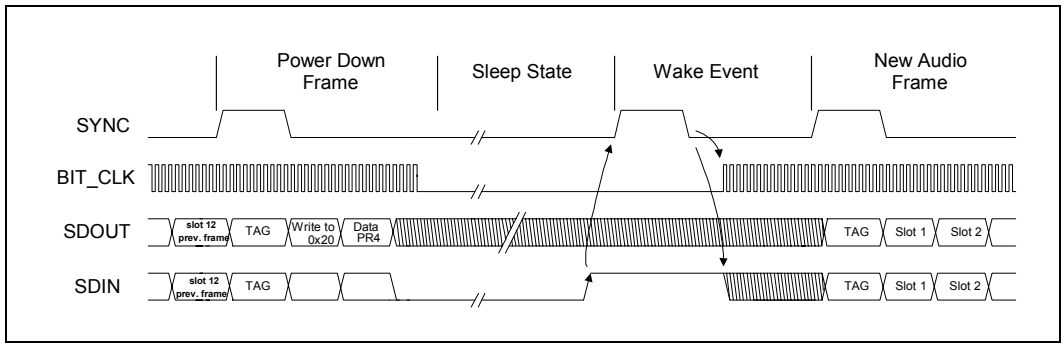
Once the codec has been instructed to halt, AC_BIT_CLK, a special wake up protocol must be used to bring the AC-link to the active mode since normal output and input frames can not be communicated in the absence of AC_BIT_CLK. Once in a low-power mode, the ICH5 provides three methods for waking up the AC-link; external wake event, cold reset and warm reset.

***Note:*** Before entering any low-power mode where the link interface to the codec is expected to be powered down while the rest of the system is awake, the software must set the “Shut Off” bit in the control register.

### 5.22.3.1 External Wake Event

Codecs can signal the controller to wake the AC-link, and wake the system using AC_SDIN.

**Figure 29. SDIN Wake Signaling**




The minimum AC_SDIN wake up pulse width is 1 us. The rising edge of AC_SDIN0, AC_SDIN1 or AC_SDIN2 causes the ICH5 to sequence through an AC-link warm reset and set the AC97_STS bit in the GPE0_STS register to wake the system. The primary codec must wait to sample AC_SYNC high and low before restarting AC_BIT_CLK as diagrammed in Figure 27. The codec that signaled the wake event must keep its AC_SDIN high until it has sampled AC_SYNC having gone high, and then low.

The AC-link protocol provides for a cold reset and a warm reset. The type of reset used depends on the system’s current power down state. Unless a cold or register reset (a write to the Reset register in the codec) is performed, wherein the AC ’97 codec registers are initialized to their default values, registers are required to keep state during all power down modes.

Once powered down, activation of the AC-link via re-assertion of the AC_SYNC signal must not occur for a minimum of four audio frame times following the frame in which the power down was triggered. When AC-link powers up, it indicates readiness via the codec ready bit.

## 5.22.4 AC '97 Cold Reset

A cold reset is achieved by asserting AC_RST# for 1 µs. By driving AC_RST# low, AC_BIT_CLK, and AC_SDOUT will be activated and all codec registers will be initialized to their default power on reset values. AC_RST# is an asynchronous AC '97 input to the codec.

## 5.22.5 AC '97 Warm Reset

A warm reset re-activates the AC-link without altering the current codec register values. A warm reset is signaled by driving AC_SYNC high for a minimum of 1 µs in the absence of AC_BIT_CLK.

Within normal frames, AC_SYNC is a synchronous AC '97 input to the codec. However, in the absence of AC_BIT_CLK, AC_SYNC is treated as an asynchronous input to the codec used in the generation of a warm reset.

The codec must not respond with the activation of AC_BIT_CLK until AC_SYNC has been sampled low again by the codec. This prevents the false detection of a new frame.

***Note:*** On receipt of wake up signalling from the codec, the digital controller issues an interrupt if enabled. Software then has to issue a warm or cold reset to the codec by setting the appropriate bit in the Global Control Register.

## 5.22.6 System Reset

Table 129 indicates the states of the link during various system reset and sleep conditions.

**Table 129. AC-link State during PCIRST#**

| Signal | Power Plane | I/O | During PCIRST#/ | After PCIRST#/ | S1 | S3 | S4/S5 |
|---|---|---|---|---|---|---|---|
| AC_RST# | Resume[3] | Output | Low | Low | Cold Reset Bit (Hi) | Low | Low |
| AC_SDOUT | Core[1] | Output | Low | Running | Low | Low | Low |
| AC_SYNC | Core | Output | Low | Running | Low | Low | Low |
| AC_BIT_CLK | Core | Input | Driven by codec | Running | Low[2,4] | Low[2,4] | Low[2,4] |
| AC_SDIN[2:0] | Resume | Input | Driven by codec | Running | Low[2,4] | Low[2,4] | Low[2,4] |

**NOTES:**
1. ICH5 core well outputs are used as strapping options for the ICH5, sampled during system reset. These signals may have weak pullups/pulldowns on them. The ICH5 outputs are driven to the appropriate level prior to AC_RST# being deasserted, preventing a codec from entering test mode. Straps are tied to the core well to prevent leakage during a suspend state.
2. The pull-down resistors on these signals are only enabled when the AC-Link Shut Off bit in the AC '97 Global Control Register is set to 1. All other times, the pull-down resistor is disabled.
3. AC_RST# are held low during S3–S5. It cannot be programmed high during a suspend state.
4. AC_BIT_CLK and AC_SDIN[2:0] are driven low by the codecs during normal states. If the codec is powered during suspend states it holds these signals low. However, if the codec is not present, or not powered in suspend, external pull-down resistors are required.

The transition of AC_RST# to the deasserted state only occurs under driver control. In the S1sleep state, the state of the AC_RST# signal is controlled by the AC '97 Cold Reset# bit (bit 1) in the Global Control register. AC_RST# is asserted (low) by the ICH5 under the following conditions:

- RSMRST# (system reset, including the a reset of the resume well and PCIRST#)
- Mechanical power up (causes PCIRST#)
- Write to CF9h hard reset (causes PCIRST#)
- Transition to S3/S4/S5 sleep states (causes PCIRST#)
- Write to AC '97 Cold Reset# bit in the Global Control Register.

Hardware never deasserts AC_RST# (i.e., never deasserts the Cold Reset# bit) automatically. Only software can deassert the Cold Reset# bit and, hence, the AC_RST# signal. This bit, while it resides in the core well, remains cleared upon return from S3/S4/S5 sleep states. The AC_RST# pin remains actively driven from the resume well as indicated.

## 5.22.7 Hardware Assist to Determine AC_SDIN Used Per Codec

Software first performs a read to one of the audio codecs. The read request goes out on AC_SDOUT. Since under our micro-architecture only one read can be performed at a time on the link, eventually the read data will come back in on one of the AC_SDIN[2:0] lines.

The codec does this by indicating that status data is valid in its TAG, then echoes the read address in slot 1 followed by the read data in slot 2.

The new function of the ICH5 hardware is to notice which AC_SDIN line contains the read return data, and to set new bits in the new register indicating which AC_SDIN line the register read data returned on. If it returned on AC_SDIN0, bits [1:0] contain the value 00. If it returned on AC_SDIN1, the bits contain the value 01, etc.

ICH5 hardware can set these bits every time register read data is returned from a function 5 read. No special command is necessary to cause the bits to be set. The new driver/BIOS software reads the bits from this register when it cares to, and can ignore it otherwise. When software is attempting to establish the codec-to-AC_SDIN mapping, it will single feed the read request and not pipeline to ensure it gets the right mapping, we cannot ensure the serialization of the access.

## 5.22.8 Software Mapping of AC_SDIN to DMA Engine

Once software has performed the register read to determine codec-to-AC_SDIN mapping, it will then either set bits [5:4] or [7:6] in the SDATA_IN MAP register to map this codec to the DMA engine. After it maps the audio codecs, it sets the "SE" (steer enable) bit, which now lets the hardware know to no longer OR the AC_SDIN lines, and to use the mappings in the register to steer the appropriate AC_SDIN line to the correct DMA engines.

This page is intentionally left blank.

# Register and Memory Mapping 6

The ICH5 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 ICH5 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. |
| **Default** | When ICH5 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 ICH5 registers accordingly. |
| **Bold** | Register bits that are highlighted in bold text indicate that the bit is implemented in the ICH5. Register bits that are not implemented or are hardwired will remain in plain text. |

# 6.1 PCI Devices and Functions

The ICH5 incorporates a variety of PCI functions as shown in Table 130. These functions are divided into four logical devices (B0:D30, B0:D31, B0:D29 and B1:D8). D30 is the hub interface-to-PCI bridge, D31 contains the PCI-to-LPC Bridge, IDE controller, SATA controller, SMBus controller and the AC ’97 Audio and Modem controller functions and D29 contains the four USB UHCI controllers and one USB EHCI controller. B1:D8 is the integrated LAN controller.

***Note:*** From a software perspective, the integrated LAN controller resides on the ICH5’s external PCI bus (See Section 5.1.2). 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 Device 31’s Functions 1–6, Device 29’s functions, or Device 8, they can individually be disabled. The integrated LAN controller will be disabled if no Platform LAN Connect component is detected (See Section 5.2). 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. This is intended to prevent software from thinking that a function is present (and reporting it to the end-user).

**Table 130. PCI Devices and Functions**

| Bus:Device:Function | Function Description |
|---|---|
| Bus 0:Device 30:Function 0 | Hub Interface to PCI Bridge |
| Bus 0:Device 31:Function 0 | PCI to LPC Bridge[1] |
| Bus 0:Device 31:Function 1 | IDE Controller |
| Bus 0:Device 31:Function 2 | New: SATA Controller |
| Bus 0:Device 31:Function 3 | SMBus Controller |
| Bus 0:Device 31:Function 5 | AC’97 Audio Controller |
| Bus 0:Device 31:Function 6 | AC’97 Modem 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 | New: USB UHCI Controller #4 |
| Bus 0:Device 29:Function 7 | USB 2.0 EHCI 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, DMA.

## 6.2 PCI Configuration Map

Each PCI function on the ICH5 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. Refer to Table 204 for a complete list of all PCI Configuration Registers.

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).

## 6.3 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.

### 6.3.1 Fixed I/O Address Ranges

Table 131 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. The hub interface cycles that go to target ranges that are marked as “Reserved” will not be decoded by the ICH5, and will be passed to PCI. If a PCI master targets one of the fixed I/O target ranges, it will be positively decoded by the ICH5 in Medium speed.

Refer to Table 205 for a complete list of all fixed I/O registers. Address ranges that are not listed or marked “Reserved” are **not** decoded by the ICH5 (unless assigned to one of the variable ranges).

### Table 131. Fixed I/O Ranges Decoded by Intel® ICH5 (Sheet 1 of 2)

| 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 |
| 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 |
| 63h | NMI Controller | NMI Controller | Processor I/F |
| 64h | Microcontroller | Microcontroller | Forwarded to LPC |
| 65h | NMI Controller | NMI Controller | Processor I/F |
| 66h | Microcontroller | Microcontroller | Forwarded to LPC |
| 67h | NMI Controller | NMI Controller | Processor I/F |
| 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 | DMA Controller and LPC or PCI | DMA |

## Table 131. Fixed I/O Ranges Decoded by Intel® ICH5 (Sheet 2 of 2)

| I/O Address | Read Target | Write Target | Internal Unit |
|---|---|---|---|
| 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 |
| 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 | See Note 3 | FERR#/IGNNE# / Interrupt Controller | Processor I/F |
| 170h–177h | IDE Controller[2] | IDE Controller[2] | Forwarded to IDE |
| 1F0h–1F7h | IDE Controller[1] | IDE Controller[1] | Forwarded to IDE |
| 376h | IDE Controller[2] | IDE Controller[2] | Forwarded to IDE |
| 3F6h | IDE Controller[1] | IDE Controller[1] | Forwarded IDE |
| 4D0h–4D1h | Interrupt Controller | Interrupt Controller | Interrupt |
| CF9h | Reset Generator | Reset Generator | Processor I/F |

**NOTES:**
1. Only if IDE Standard I/O space is enabled for Primary Channel and the IDE Controller is in legacy mode. Otherwise, the target is PCI.
2. Only if IDE Standard I/O space is enabled for Secondary Channel and the IDE Controller is in legacy mode. Otherwise, the target is PCI.
3. If POS_DEC_EN bit is enabled, reads from F0h will not be decoded by the ICH5. If POS_DEC_EN is not enabled, reads from F0h will forward to LPC.

## 6.3.2 Variable I/O Decode Ranges

Table 132 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.

When a cycle is detected on the hub interface, the ICH5 positively decodes the cycle. If the response is on the behalf of an LPC device, ICH5 forwards the cycle to the LPC interface.

Refer to Table 206 for a complete list of all variable I/O registers.

***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 ICH5 does not perform any checks for conflicts.

**Table 132. 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 |
| USB UHCI Controller #1 | Anywhere in 64 KB I/O Space | 32 | USB Unit 1 |
| SMBus | Anywhere in 64 KB I/O Space | 32 | SMB Unit |
| AC'97 Audio Mixer | Anywhere in 64 KB I/O Space | 256 | AC'97 Unit |
| AC'97 Audio Bus Master | Anywhere in 64 KB I/O Space | 64 | AC'97 Unit |
| AC'97 Modem Mixer | Anywhere in 64 KB I/O Space | 256 | AC'97 Unit |
| AC'97 Modem Bus Master | Anywhere in 64 KB I/O Space | 128 | AC'97 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 |
| MIDI | 4 Ranges in 64 KB I/O Space | 2 | LPC Peripheral |
| MSS | 4 Ranges in 64 KB I/O Space | 8 | LPC Peripheral |
| SoundBlaster | 2 Ranges in 64 KB I/O Space | 32 | LPC Peripheral |
| LAN | Anywhere in 64 KB I/O Space | 64 | LAN Unit |
| 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 |
| LPC Generic 1 | Anywhere in 64 KB I/O Space | 128 | LPC Peripheral |
| LPC Generic 2 | Anywhere in 64 KB I/O Space | 16 | LPC Peripheral |
| Monitors 4:7 | Anywhere in 64 KB I/O Space | 16 | LPC Peripheral or Trap on PCI |
| Native IDE Primary Command | Anywhere in 64 KB I/O Space | 8 | IDE Unit |
| Native IDE Primary Control | Anywhere in 64 KB I/O Space | 4 | IDE Unit |
| Native IDE Secondary Command | Anywhere in 64 KB I/O Space | 8 | IDE Unit |
| Native IDE Secondary Control | Anywhere in 64 KB I/O Space | 4 | IDE Unit |

## 6.4 Memory Map

Table 133 shows (from the processor perspective) the memory ranges that the ICH5 decodes. Cycles that arrive from the hub interface that are not directed to any of the internal memory targets that decode directly from hub interface will be driven out on PCI. The ICH5 may then claim the cycle for it to be forwarded to LPC or claimed by the internal APIC. If subtractive decode is enabled, the cycle can be forwarded to LPC.

PCI cycles generated by an external PCI master are positively decoded unless they falls in the PCI-to-PCI bridge forwarding range (those addresses are reserved for PCI peer-to-peer traffic). If the cycle is not in the I/O APIC or LPC ranges, it is forwarded up the hub interface to the host controller. PCI masters can not access the memory ranges for functions that decode directly from hub interface.

**Table 133. Memory Decode Ranges from Processor Perspective (Sheet 1 of 2)**

| Memory Range | Target | Dependency/Comments |
|---|---|---|
| 0000 0000h–000D FFFFh<br>0010 0000h–TOM (Top of Memory) | Main Memory | TOM registers in host controller |
| 000E 0000h–000F FFFFh | Flash BIOS | Bit 7 in flash BIOS decode enable register is set |
| FEC0 0000h–FEC0 0100h | I/O APIC inside ICH5 | |
| FFC0 0000h–FFC7 FFFFh<br>FF80 0000h–FF87 FFFFh | Flash BIOS | Bit 0 in flash BIOS decode enable register |
| FFC8 0000h–FFCF FFFFh<br>FF88 0000h–FF8F FFFFh | Flash BIOS | Bit 1 in flash BIOS decode enable register |
| FFD0 0000h–FFD7 FFFFh<br>FF90 0000h–FF97 FFFFh | Flash BIOS | Bit 2 in flash BIOS decode enable register is set |
| FFD8 0000h–FFDF FFFFh<br>FF98 0000h–FF9F FFFFh | Flash BIOS | Bit 3 in flash BIOS decode enable register is set |
| FFE0 000h–FFE7 FFFFh<br>FFA0 0000h–FFA7 FFFFh | Flash BIOS | Bit 4 in flash BIOS decode enable register is set |
| FFE8 0000h–FFEF FFFFh<br>FFA8 0000h–FFAF FFFFh | Flash BIOS | Bit 5 in flash BIOS decode enable register is set |
| FFF0 0000h–FFF7 FFFFh<br>FFB0 0000h–FFB7 FFFFh | Flash BIOS | Bit 6 in flash BIOS decode enable register is set. |
| FFF8 0000h–FFFF FFFFh<br>FFB8 0000h–FFBF FFFFh | Flash BIOS | 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 | Flash BIOS | Bit 3 in flash BIOS decode enable 2 register is set |
| FF60 0000h–FF6F FFFFh<br>FF20 0000h–FF2F FFFFh | Flash BIOS | Bit 2 in flash BIOS decode enable 2 register is set |
| FF50 0000h–FF5F FFFFh<br>FF10 0000h–FF1F FFFFh | Flash BIOS | Bit 1 in flash BIOS decode enable 2 register is set |
| FF40 0000h–FF4F FFFFh<br>FF00 0000h–FF0F FFFFh | Flash BIOS | Bit 0 in flash BIOS decode enable 2 register is set |
| 4 KB anywhere in 4 GB range | Integrated LAN Controller | Enable via BAR in Device 29:Function 0 (Integrated LAN Controller) |
| 1 KB anywhere in 4 GB range | IDE Expansion[2] | Enable via standard PCI mechanism and bits in IDE I/O Configuration Register (Device 31, Function 1) |

**Table 133. Memory Decode Ranges from Processor Perspective (Sheet 2 of 2)**

| Memory Range | Target | Dependency/Comments |
|---|---|---|
| 1 KB anywhere in 4 GB range | USB EHCI Controller[1,2] | Enable via standard PCI mechanism (Device 29, Function 7) |
| FED0 X000–FED0 X3FF | High-Precision Event Timers[1,2] | 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. These ranges are decoded directly from Hub Interface. The memory cycles will not be seen on PCI.
2. Software must not attempt locks to memory mapped I/O ranges for USB EHCI, High-Precision Event Timers, and IDE Expansion. If attempted, the lock is not honored, which means potential deadlock conditions may occur.

## 6.4.1 Boot-Block Update Scheme

The ICH5 supports a "top-block swap" mode that has the ICH5 swap the top block in the flash BIOS (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 ICH5 will invert A16 for cycles targeting flash BIOS space. When this bit is 0, the ICH5 will not invert A16. This bit is automatically set to 0 by RTCRST#, but not by PCIRST#.

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 flash BIOS. processor access to FFFF_0000h through FFFF_FFFFh will be directed to FFFE_0000h through FFFE_FFFFh in the flash BIOS, 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 2.21.1). 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 flash BIOS space, not feature space.

***Note:*** The top-block swap mode has no effect on accesses below FFFE_0000h.

# LAN Controller Registers (B1:D8:F0) 7

The ICH5 integrated LAN controller appears to reside at PCI Device 8, Function 0 on the secondary side of the ICH5's virtual PCI-to-PCI Bridge (See Section 5.1.2). 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.

## 7.1 PCI Configuration Registers (LAN Controller—B1:D8:F0)

***Note:*** Address locations that are not shown in Table 134 should be treated as Reserved (See Section 6.2 for details).

**Table 134. LAN Controller PCI Register Address Map (LAN Controller—B1:D8:F0)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086h | RO |
| 02–03h | DID | Device Identification | 1051h | RO |
| 04–05h | PCICMD | PCI Command | 0000h | RO, RW |
| 06–07h | 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 | 02h | RO |
| 0Ch | CLS | Cache Line Size | 00h | R/W |
| 0Dh | PMLT | Primary Master Latency Timer | 00h | R/W |
| 0Eh | HEADTYP | Header Type | 00h | RO |
| 10–13h | CSR_MEM_BASE | CSR Memory–Mapped Base Address | 00000008h | R/W, RO |
| 14–17h | CSR_IO_BASE | CSR I/O–Mapped Base Address | 00000001h | R/W, RO |
| 2C–2Dh | SVID | Subsystem Vendor Identification | 0000h | RO |
| 2E–2Fh | SID | Subsystem Identification | 0000h | RO |
| 34h | CAP_PTR | Capabilities Pointer | DCh | RO |
| 3Ch | INT_LN | Interrupt Line | 00h | R/W |
| 3D | INT_PN | Interrupt Pin | 01h | RO |
| 3E | MIN_GNT | Minimum Grant | 08h | RO |
| 3F | MAX_LAT | Maximum Latency | 38h | RO |
| DCh | CAP_ID | Capability ID | 01h | RO |
| DDh | NXT_PTR | Next Item Pointer | 00h | RO |
| DE–DFh | PM_CAP | Power Management Capabilities | FE21h | RO |
| E0–E1h | PMCSR | Power Management Control/Status | 0000h | R/W, RO, R/WC |
| E3 | PCIDATA | PCI Power Management Data | 00h | RO |

## 7.1.1 VID—Vendor Identification Register (LAN Controller—B1:D8:F0)

Offset Address: 00–01h
Default Value: 8086h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | Vendor ID — RO. This is a 16-bit value assigned to Intel. |

## 7.1.2 DID—Device Identification Register (LAN Controller—B1:D8:F0)

Offset Address: 02–03h
Default Value: 1051h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | Device ID — RO. This is a 16-bit value assigned to the Intel® ICH5 integrated LAN controller.<br>1. If the EEPROM is not present (or not properly programmed), reads to the Device ID return the default value of 1051h.<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 7.1.14 - SID, Subsystem ID of LAN controller for detail) |

## 7.1.3 PCICMD—PCI Command Register (LAN Controller—B1:D8:F0)

Offset Address: 04–05h
Default Value: 0000h
Attribute: RO, R/W
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 the hub interface. |
| 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 Intel® ICH5's integrated may function as a PCI bus master. |
| 1 | **Memory Space Enable (MSE)** — R/W.<br>0 = Disable<br>1 = Enable. The ICH5'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 ICH5's integrated LAN controller will respond to the I/O space accesses. |

## 7.1.4 PCISTS—PCI Status Register (LAN Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 06–07h | Attribute: | RO, R/WC |
| Default Value: | 0290h | 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 Intel® ICH5'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 ICH5'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 = this bit is cleared by writing a 1 to the bit location.<br>1 = The ICH5'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 ICH5'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 never 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. |
| 2:0 | Reserved |

## 7.1.5 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 Revision ID fields in the EEPROM.<br><br>**NOTE:** Refer to the latest Intel® ICH5 / ICH5R Specification Update for the value of the Revision Identification register. |

## 7.1.6 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. |

## 7.1.7 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. |

## 7.1.8 CLS—Cache Line Size Register (LAN Controller—B1:D8:F0)

Offset Address: 0Ch Attribute: RW
Default Value: 00h Size: 8 bits

| Bit | Description |
|---|---|
| 7:5 | Reserved |
| 4:3 | **Cache Line Size (CLS)** — RW.<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 |

## 7.1.9 PMLT—Primary Master Latency Timer Register (LAN Controller—B1:D8:F0)

Offset Address: 0Dh Attribute: R/W
Default Value: 00h 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 |

## 7.1.10 HEADTYP—Header Type Register (LAN Controller—B1:D8:F0)

Offset Address: 0Eh Attribute: RO
Default Value: 00h 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. |

## 7.1.11 CSR_MEM_BASE — CSR Memory-Mapped Base Address Register (LAN Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 10–13h | Attribute: | R/W, RO |
| Default Value: | 00000008h | Size: | 32 bits |

*Note:* The ICH5'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 CSR 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. |

## 7.1.12 CSR_IO_BASE — CSR I/O-Mapped Base Address Register (LAN Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 14–17h | Attribute: | R/W, RO |
| Default Value: | 00000001h | Size: | 32 bits |

*Note:* The ICH5'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 CSR. |
| 5:1 | Reserved |
| 0 | I/O Space Indicator (IO_SPACE) — RO. Hardwired to 1 to indicate that this base address maps to I/O space. |

## 7.1.13 SVID — Subsystem Vendor Identification Register (LAN Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 2C–2D | Attribute: | RO |
| Default Value: | 0000h | Size: | 16 bits |

| Bit | Description |
|---|---|
| 15:0 | Subsystem Vendor ID (SVID) — RO. See Section 7.1.14 for details. |

## 7.1.14 SID — Subsystem Identification Register (LAN Controller—B1:D8:F0)

Offset Address: 2E–2Fh
Default Value: 0000h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | Subsystem ID (SID) — RO. |

*Note:* The ICH5'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 135.

**Table 135. Configuration of Subsystem ID and Subsystem Vendor ID via EEPROM**

| Bits 15:14 | Bit 13 | Device ID | Vendor ID | Revision ID | 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 Revision ID is subject to change according to the silicon stepping.
2. The Device ID is loaded from Word 23h only if the value of Word 23h is not 0000h or FFFFh

## 7.1.15 CAP_PTR — Capabilities Pointer Register (LAN Controller—B1:D8:F0)

Offset Address: 34h
Default Value: DCh
Attribute: RO
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. |

## 7.1.16 INT_LN — Interrupt Line Register (LAN Controller—B1:D8:F0)

Offset Address: 3Ch
Default Value: 00h
Attribute: R/W
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. |

## 7.1.17 INT_PN — Interrupt Pin Register (LAN Controller—B1:D8:F0)

Offset Address: 3Dh
Default Value: 01h
Attribute: RO
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 Intel® ICH5 implementation, when the LAN controller interrupt is generated PIRQE# will go active, not PIRQA#. Note that if the PIRQE# signal is used as a GPIO, the external visibility will be lost (though PIRQE# will still go active internally). |

## 7.1.18 MIN_GNT — Minimum Grant Register (LAN Controller—B1:D8:F0)

Offset Address: 3Eh
Default Value: 08h
Attribute: RO
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. |

## 7.1.19 MAX_LAT — Maximum Latency Register (LAN Controller—B1:D8:F0)

Offset Address: 3Fh
Default Value: 38h
Attribute: RO
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. |

## 7.1.20 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 Intel® ICH5's integrated LAN controller supports PCI power management. |

## 7.1.21 NXT_PTR — Next Item Pointer Register (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. |

## 7.1.22 PM_CAP — Power Management Capabilities Register (LAN Controller—B1:D8:F0)

Offset Address: DE–DFh
Default Value: FE21h
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.* |

## 7.1.23 PMCSR — Power Management Control/Status Register (LAN Controller—B1:D8:F0)

Offset Address: E0–E1h
Default Value: 0000h
Attribute: RO, R/W, R/WC
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 eight 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 Intel® ICH5’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 |

## 7.1.24 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 ICH5'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 136.

**Table 136. 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 |

# 7.2 LAN Control / Status Registers (CSR) (LAN Controller—B1:D8:F0)

**Table 137. Intel® ICH5 Integrated LAN Controller CSR Space Register Address Map**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 01h–00h | SCB_STA | System Control Block Status Word | 0000h | R/WC, RO |
| 03h–02h | SCB_CMD | System Control Block Command Word | 0000h | R/W, WO |
| 07h–04h | SCB_GENPNT | System Control Block General Pointer | 0000 0000h | R/W |
| 0Bh–08h | Port | PORT Interface | 0000 0000h | R/W (special) |
| 0Dh–0Ch | — | Reserved | — | — |
| 0Eh | EEPROM_CNTL | EEPROM Control | 00 | R/W, RO, WO |
| 0Fh | — | Reserved | — | — |
| 13h–10h | MDT_CNTL | Management Data Interface Control | 0000 0000h | R/W (special) |
| 17h–14h | REC_DMA_BC | Receive DMA Byte Count | 0000 0000h | RO |
| 18h | | Early Receive Interrupt | 00h | R/W |
| 1A–19h | 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 | — | — |

## 7.2.1 SCB_STA—System Control Block Status Word Register (LAN Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 00–01h | Attribute: | R/WC, RO |
| Default Value: | 0000h | Size: | 16 bits |

The ICH5'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. |
| 13 | **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 two, 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. |
| 12 | **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. |
| 11 | **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). |
| 10 | **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. |
| 9 | **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. |
| 8 | **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. |
| 7:6 | **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 |

| Bit | Description |
|---|---|
| 5:2 | **Receive Unit Status (RUS)** — RO.<br>0000 = Idle<br>0001 = Suspended<br>0010 = No Resources<br>0011 = Reserved<br>0100 = Ready<br>0101 = Reserved<br>0110 = Reserved<br>0111 = Reserved<br>1000 = Reserved<br>1001 = Suspended with no more RBDs<br>1010 = No resources due to no more RBDs<br>1011 = Reserved<br>1100 = Ready with no RBDs present<br>1101 = Reserved<br>1110 = Reserved<br>1111 = Reserved |
| 1:0 | Reserved |

## 7.2.2 SCB_CMD—System Control Block Command Word Register (LAN Controller—B1:D8:F0)

Offset Address: 02–03h
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 (never 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 never started. |
| 3 | Reserved |
| 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 0s 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. |

## 7.2.3 SCB_GENPNT—System Control Block General Pointer Register (LAN Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 04–07h | 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. |

## 7.2.4 PORT—PORT Interface Register (LAN Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 08–0Bh | Attribute: | R/W (special) |
| Default Value: | 0000 0000h | Size: | 32 bits |

The PORT interface allows the processor to reset the ICH5'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 138. 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 138. 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 |

## 7.2.5 EEPROM_CNTL—EEPROM Control Register (LAN Controller—B1:D8:F0)

Offset Address: 0Eh — Attribute: RO, R/W, WO
Default Value: 00h — 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 Intel® ICH5’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 ICH5’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 ICH5’s EE_SHCLK signal low.<br>1 = Drives the ICH5’s EE_SHCLK signal high. |

## 7.2.6 MDI_CNTL—Management Data Interface (MDI) Control Register (LAN Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 10–13h | Attribute: | R/W (special) |
| Default Value: | 0000 0000h | 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. |
| 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. |

## 7.2.7 REC_DMA_BC—Receive DMA Byte Count Register (LAN Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 14–17h | 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. |

## 7.2.8 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*** quadwords 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. |

## 7.2.9 FLOW_CNTL—Flow Control Register (LAN Controller—B1:D8:F0)

Offset Address: 19–1Ah
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. |
| 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.<br>**Bits 2:0** / **Free Bytes in Receive FIFO** / **Comment**<br>000 — 0.50 KB — Fast system (recommended default)<br>001 — 1.00 KB<br>010 — 1.25 KB<br>011 — 1.50 KB<br>100 — 1.75 KB<br>101 — 2.00 KB<br>110 — 2.25 KB<br>111 — 2.50 KB — Slow system |

## 7.2.10 PMDR—Power Management Driver Register (LAN Controller—B1:D8:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 1Bh | Attribute: | R/WC |
| Default Value: | 00h | Size: | 8 bits |

The ICH5'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 CSR. |
| 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. |
| 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. |

## 7.2.11 GENCNTL—General Control Register (LAN Controller—B1:D8:F0)

Offset Address: 1Ch
Default Value: 00h
Attribute: R/W
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 1 ms 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 Intel® ICH5'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 |

## 7.2.12 GENSTA—General Status Register (LAN Controller—B1:D8:F0)

Offset Address: 1Dh
Default Value: 00h
Attribute: RO
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 Mbps<br>1 = 100 Mbps |
| 0 | **Link Status Indication** — RO. This bit indicates the status of the link.<br>0 = Invalid<br>1 = Valid |

## 7.2.13 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 ICH5 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. |

## 7.2.14 Statistical Counters (LAN Controller—B1:D8:F0)

The ICH5'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 139. 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 139. 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 ICH5'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.

# Hub Interface to PCI Bridge Registers (D30:F0) 8

The hub interface to PCI Bridge resides in PCI Device 30, Function 0 on bus #0. This portion of the ICH5 implements the buffering and control logic between PCI and the hub interface. The arbitration for the PCI bus is handled by this PCI device. The PCI decoder in this device must decode the ranges for the hub interface. All register contents will be lost when core well power is removed.

## 8.1 PCI Configuration Registers (D30:F0)

***Note:*** Address locations that are not shown in Table 140 should be treated as Reserved (see Section 6.2 for details).

**Table 140. Hub Interface PCI Register Address Map (HUB-PCI—D30:F0) (Sheet 1 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086h | RO |
| 02–03h | DID | Device Identification | 244Eh | RO |
| 04–05h | PCICMD | PCI Command | 0001h | R/W, RO |
| 06–07h | PCISTS | PCI Status | 0080h | R/WC, RO |
| 08h | RID | Revision Identification | See register description | RO |
| 0Ah | SCC | Sub Class Code | 04h | RO |
| 0Bh | BCC | Base Class Code | 06h | RO |
| 0Dh | PMLT | Primary Master Latency Timer | 00h | RO |
| 0Eh | HEADTYP | Header Type | 01h | RO |
| 18h | PBUS_NUM | Primary Bus Number | 00h | RO |
| 19h | SBUS_NUM | Secondary Bus Number | 00h | R/W |
| 1Ah | SUB_BUS_NUM | Subordinate Bus Number | 00h | R/W |
| 1Bh | SMLT | Secondary Master Latency Timer | 00h | R/W |
| 1Ch | IOBASE | I/O Base | F0h | R/W, RO |
| 1Dh | IOLIM | I/O Limit | 00h | R/W, RO |
| 1E–1Fh | SECSTS | Secondary Status | 0280h | R/WC, RO |
| 20–21h | MEMBASE | Memory Base | FFF0h | R/W |
| 22–23h | MEMLIM | Memory Limit | 0000h | R/W |
| 24–25h | PREF_MEM_BASE | Prefetchable Memory Base | FFF0h | R/W |
| 26–27h | PREF_MEM_MLT | Prefetchable Memory Limit | 0000h | R/W |
| 30–31h | IOBASE_HI | I/O Base Upper 16 Bits | 0000h | RO |

**Table 140. Hub Interface PCI Register Address Map (HUB-PCI—D30:F0) (Sheet 2 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 32–33h | IOLIMIT_HI | I/O Limit Upper 16 Bits | 0000h | RO |
| 3Ch | INT_LN | Interrupt Line | 00h | RO |
| 3E–3Fh | BRIDGE_CNT | Bridge Control | 0000h | R/W, RO |
| 40–43h | HI1_CMD | Hub Interface 1 Command Control | 76202802h | R/W, RO |
| 44–45h | DEVICE_HIDE | Secondary PCI Device Hiding | 00 | R/W |
| 50–53h | CNF | Policy Configuration | 00406402h | R/W |
| 70h | MTT | Multi-Transaction Timer | 20h | R/W |
| 82h | PCI_MAST_STS | PCI Master Status | 00h | R/WC |
| 90h | ERR_CMD | Error Command | 00h | R/W |
| 92h | ERR_STS | Error Status | 00h | R/WC |

**NOTE:** Refer to the latest *Intel® ICH5 / ICH5R Specification Update* for the value of the Revision Identification register.

## 8.1.1 VID—Vendor Identification Register (HUB-PCI—D30:F0)

Offset Address: 00–01h
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. |

## 8.1.2 DID—Device Identification Register (HUB-PCI—D30:F0)

Offset Address: 02–03h
Default Value: 244Eh
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | Device ID — RO. This is a 16-bit value assigned to the Intel® ICH5 hub interface to PCI bridge. |

## 8.1.3 PCICMD—PCI Command Register (HUB-PCI—D30:F0)

Offset Address: 04–05h
Default Value: 0001h
Attribute: R/W, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:10 | Reserved |
| 9 | Fast Back to Back Enable (FBE) — RO. Hardwired to 0. The Intel® ICH5 does not support this capability. |
| 8 | **SERR# Enable (SERR_EN)** — R/W.<br>0 = Disable<br>1 = Enable the ICH5 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 |
| 6 | **Parity Error Response (PER)** — R/W.<br>0 = The ICH5 ignores parity errors on the hub interface.<br>1 = The ICH5 is allowed to report parity errors detected on the hub interface.<br><br>**NOTE:** The HP_Unsupported bit (D30:F0:40h bit 20) must be cleared in order for this bit to have any effect. |
| 5 | VGA Palette Snoop (VPS) — RO. Hardwired to 0. |
| 4 | Memory Write and Invalidate Enable (MWE) — RO. Hardwired to 0. |
| 3 | Special Cycle Enable (SCE) — RO. Hardwired to 0 by P2P Bridge specification. |
| 2 | **Bus Master Enable (BME)** — R/W.<br>0 = Disable<br>1 = Enable. Allows the Hub interface-to-PCI bridge to accept cycles from PCI to run on the hub interface.<br><br>**NOTE:** This bit does not affect the CF8h and CFCh I/O accesses.<br>Cycles that generated from the ICH5's Device 31 functionality are not blocked by clearing this bit. (PC/PCI Cascade Mode cycles may be blocked) |
| 1 | **Memory Space Enable (MSE)** — R/W. The ICH5 provides this bit as read/writable for software only. However, the ICH5 ignores the programming of this bit, and runs hub interface memory cycles to PCI. |
| 0 | **I/O Space Enable (IOSE)** — R/W. The ICH5 provides this bit as read/writable for software only. However, the ICH5 ignores the programming of this bit and runs hub interface I/O cycles to PCI that are not intended for USB, IDE, or AC '97. |

## 8.1.4 PCISTS—PCI Status Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 06–07h | Attribute: | R/WC, RO |
| Default Value: | 0080h | 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 Intel® ICH5 detected a parity error on the hub interface and the HP_Unsupported bit (D30:F0:40h bit 20) is 0. This bit gets set even if the Parity Error Response bit (D30:F0:04 bit 6) is not set. |
| 14 | **Signaled System Error (SSE)** — R/WC.<br>0 = Error described below **not** detected.<br>1 = An address, or command parity error, or special cycles data parity error has been detected on the PCI bus, and the Parity Error Response bit (D30:F0, Offset 04h, bit 6) is set. If this bit is set because of parity error and the D30:F0 SERR_EN bit (Offset 04h, bit 8) is also set, the ICH5 will generate an NMI (or SMI# if NMI routed to SMI#). |
| 13 | **Received Master Abort (RMA)** — R/WC.<br>0 = Master abort **not** received from hub interface.<br>1 = ICH5 received a master abort from the hub interface device. |
| 12 | **Received Target Abort (RTA)** — R/WC.<br>0 = Target abort **not** received from hub interface.<br>1 = ICH5 received a target abort from the hub interface device. The TCO logic can cause an SMI#, NMI, or interrupt based on this bit getting set. |
| 11 | **Signaled Target Abort (STA)** — R/WC.<br>0 = ICH5 did **not** signal a target abort on hub interface.<br>1 = ICH5 signals a target abort condition on the hub interface. |
| 10:9 | DEVSEL# Timing Status (DEV_STS) — RO.<br>00h = Fast timing. This register applies to the hub interface; therefore, this field does not matter. |
| 8 | **Master Data Parity Error Detected (MDPD)** — R/WC. Since this register applies to the hub interface, the ICH5 must interpret this bit differently than it is in the *PCI Local Bus Specification, Revision 2.3*.<br>0 = Parity error **not** detected on hub interface.<br>1 = ICH5 detects a parity error on the hub interface and the Parity Error Response bit in the PCI Command Register (offset 04h, bit 6) is set. |
| 7 | Fast Back to Back Capable (FB2BC) — RO. Hardwired to 1. |
| 6 | Reserved |
| 5 | 66 MHz Capable (66MHZ_CAP) — RO. Hardwired to 0. |
| 4:0 | Reserved |

## 8.1.5 RID—Revision Identification Register (HUB-PCI—D30:F0)

Offset Address: 08h Attribute: RO
Default Value: See bit Description Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Revision ID — RO. 8-bit value that indicates the revision number for the Intel® ICH5 hub interface to PCI bridge.<br><br>**NOTE:** Refer to the latest Intel® ICH5 / ICH5R Specification Update for the value of the Revision Identification register. |

## 8.1.6 SCC—Sub-Class Code Register (HUB-PCI—D30:F0)

Offset Address: 0Ah Attribute: RO
Default Value: 04h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Sub Class Code (SCC) — RO. This 8-bit value indicates the category of bridge for the Intel® ICH5 hub interface to PCI bridge.<br>04h = PCI-to-PCI bridge. |

## 8.1.7 BCC—Base-Class Code Register (HUB-PCI—D30:F0)

Offset Address: 0Bh Attribute: RO
Default Value: 06h Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Base Class Code (BCC) — RO. This 8-bit value indicates the type of device for the Intel® ICH5 hub interface to PCI bridge.<br>06h = Bridge device. |

## 8.1.8 PMLT—Primary Master Latency Timer Register (HUB-PCI—D30:F0)

Offset Address: 0Dh Attribute: RO
Default Value: 00h Size: 8 bits

This register does not apply to hub interface.

| Bit | Description |
|---|---|
| 7:3 | Master Latency Timer Count (MLTC) — RO. Not implemented. |
| 2:0 | Reserved |

## 8.1.9 HEADTYP—Header Type Register (HUB-PCI—D30:F0)

Offset Address: 0Eh
Default Value: 01h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7 | Multi-Function Device (MFD) — RO. This bit is 0 to indicate a single function device. |
| 6:0 | Header Type (HTYPE) — RO. This 8-bit field identifies the header layout of the configuration space, which is a PCI-to-PCI bridge in this case. |

## 8.1.10 PBUS_NUM—Primary Bus Number Register (HUB-PCI—D30:F0)

Offset Address: 18h
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Primary Bus Number — RO. This field indicates the bus number of the hub interface and is hardwired to 00h. |

## 8.1.11 SBUS_NUM—Secondary Bus Number Register (HUB-PCI—D30:F0)

Offset Address: 19h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Secondary Bus Number** — R/W. This field indicates the bus number of PCI.<br><br>**NOTE:** When this number is equal to the primary bus number (i.e., bus #0), the Intel® ICH5 will run hub interface configuration cycles to this bus number as Type 1 configuration cycles on PCI. |

## 8.1.12 SUB_BUS_NUM—Subordinate Bus Number Register (HUB-PCI—D30:F0)

Offset Address: 1A
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Subordinate Bus Number** — R/W. This field specifies the highest PCI bus number below the hub interface to PCI bridge. If a Type 1 configuration cycle from the hub interface does not fall in the Secondary-to-Subordinate Bus ranges of Device 30, the Intel® ICH5 indicates a master abort back to the hub interface. |

## 8.1.13 SMLT—Secondary Master Latency Timer Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 1Bh | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |

This Master Latency Timer (MLT) controls the amount of time that the ICH5 will continue to burst data as a master on the PCI bus. When the ICH5 starts the cycle after being granted the bus, the counter is loaded and starts counting down from the assertion of FRAME#. If the internal grant to this device is removed, then the expiration of the MLT counter will result in the deassertion of FRAME#. If the internal grant has not been removed, then the ICH5 can continue to own 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 Intel® ICH5 remains as master of the bus. |
| 2:0 | Reserved |

## 8.1.14 IOBASE—I/O Base Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 1Ch | Attribute: | R/W, RO |
| Default Value: | F0h | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:4 | **I/O Address Base Bits [15:12]** — R/W. I/O This field provides 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 Addressing Capability — RO. This field is hardwired to 0h indicating that the hub interface to PCI bridge does not support 32-bit I/O addressing. This means that the I/O Base and Limit Upper Address registers must be read only. |

## 8.1.15 IOLIM—I/O Limit Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 1Dh | Attribute: | R/W, RO |
| Default Value: | 00h | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:4 | **I/O Address Limit Bits [15:12]** — R/W. I/O This field provides base bits corresponding to address lines 15:12 for 4-KB alignment. Bits 11:0 are assumed to be padded to FFFh. |
| 3:0 | I/O Addressing Capability — RO. This field is hardwired to 0h indicating that the hub interface-to-PCI bridge does not support 32-bit I/O addressing. This means that the I/O Base and Limit Upper Address registers must be read only. |

## 8.1.16 SECSTS—Secondary Status Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 1E–1Fh | Attribute: | R/WC, RO |
| Default Value: | 0280h | 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 = Intel® ICH5 detected a parity error on the PCI bus. |
| 14 | **Received System Error (SSE)** — R/WC.<br>0 = SERR# assertion **not** received<br>1 = SERR# assertion is received on PCI. |
| 13 | **Received Master Abort (RMA)** — R/WC.<br>0 = No master abort.<br>1 = Hub interface to PCI cycle is master-aborted on PCI. |
| 12 | **Received Target Abort (RTA)** — R/WC.<br>0 = No target abort.<br>1 = Hub interface to PCI cycle is target-aborted on PCI. For “completion required” cycles from the hub interface, this event should also set the Signaled Target Abort in the Primary Status Register in this device, and the ICH5 must send the “target abort” status back to the hub interface. |
| 11 | Signaled Target Abort (STA) — RO. The ICH5 does not generate target aborts. |
| 10:9 | **DEVSEL# Timing Status (DEV_STS)** — RO.<br>01h = Medium timing. |
| 8 | **Master Data Parity Error Detected (MDPD)** — R/WC.<br>0 = Conditions described below **not** met.<br>1 = The ICH5 sets this bit when all of the following three conditions are met:<br>- The Parity Error Response Enable bit in the Bridge Control Register (bit 0, offset 3Eh) is set.<br>- USB, AC ’97 or IDE is a Master.<br>- PERR# asserts during a write cycle OR a parity error is detected internally during a read cycle. |
| 7 | Fast Back to Back Capable (FB2BC) — RO. Hardwired to 1 to indicate that the PCI to hub interface target logic is capable of receiving fast back-to-back cycles. |
| 6 | Reserved |
| 5 | 66 MHz Capable (66MHZ_CAP) — RO. Hardwired to 0. |
| 4:0 | Reserved |

## 8.1.17 MEMBASE—Memory Base Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 20–21h | Attribute: | R/W |
| Default Value: | FFF0h | Size: | 16 bits |

This register defines the base of the hub interface to PCI non-prefetchable memory range. Since the ICH5 forwards all hub interface memory accesses that are not taken by integrated functions to PCI, the ICH5 only uses this information for determining when not to accept cycles as a target.

This register must be initialized by the configuration software. For the purpose of address decode, address bits A[19:0] are assumed to be 0. Thus, the bottom of the defined memory address range will be aligned to a 1-MB boundary.

| Bit | Description |
|---|---|
| 15:4 | **Memory Address Base** — R/W. This field defines the base of the memory range for PCI. These 12 bits correspond to address bits 31:20. |
| 3:0 | Reserved |

## 8.1.18 MEMLIM—Memory Limit Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 22–23h | Attribute: | R/W |
| Default Value: | 0000h | Size: | 16 bits |

This register defines the upper limit of the hub interface to PCI non-prefetchable memory range. Since the ICH5 forwards all hub interface memory accesses to PCI, the ICH5 only uses this information for determining when not to accept cycles as a target.

This register must be initialized by the config software. For the purpose of address decode, address bits A[19:0] are assumed to be FFFFFh. Thus, the top of the defined memory address range will be aligned to a 1-MB boundary.

| Bit | Description |
|---|---|
| 15:4 | **Memory Address Limit** — R/W. This field defines the top of the memory range for PCI. These 12 bits correspond to address bits 31:20. |
| 3:0 | Reserved. |

## 8.1.19 PREF_MEM_BASE—Prefetchable Memory Base Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 24–25h | Attribute: | R/W |
| Default Value: | FFF0h | Size: | 16-bit |

| Bit | Description |
|---|---|
| 15:4 | **Prefetchable Memory Address Base** — R/W. This field defines the base address of the prefetchable memory address range for PCI. These 12 bits correspond to address bits 31:20. |
| 3:0 | Reserved. |

## 8.1.20 PREF_MEM_MLT—Prefetchable Memory Limit Register (HUB-PCI—D30:F0)

Offset Address: 26–27h
Default Value: 0000h
Attribute: R/W
Size: 16-bit

| Bit | Description |
|---|---|
| 15:4 | **Prefetchable Memory Address Limit** — RW. This field defines the limit address of the prefetchable memory address range for PCI. These 12 bits correspond to address bits 31:20. |
| 3:0 | Reserved. |

## 8.1.21 IOBASE_HI—I/O Base Upper 16 Bits Register (HUB-PCI—D30:F0)

Offset Address: 30–31h
Default Value: 0000h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | I/O Address Base Upper 16 bits [31:16] — RO. Not supported; hardwired to 0. |

## 8.1.22 IOLIM_HI—I/O Limit Upper 16 Bits Register (HUB-PCI—D30:F0)

Offset Address: 32–33h
Default Value: 0000h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | I/O Address Limit Upper 16 bits [31:16] — RO. Not supported; hardwired to 0. |

## 8.1.23 INT_LN—Interrupt Line Register (HUB-PCI—D30:F0)

Offset Address: 3Ch
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Interrupt Line Routing (INT_LN) — RO. Hardwired to 00h. The bridge does not generate interrupts, and interrupts from downstream devices are routed around the bridge. |

## 8.1.24 BRIDGE_CNT—Bridge Control Register (HUB-PCI—D30:F0)

Offset Address: 3E–3Fh
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:12 | Reserved |
| 11 | **Discard Timer SERR# Enable (DTSE)** — R/W. The Intel® ICH5 does not treat a discarded delayed transaction on the secondary interface as an error (see bit 10 in this register). Therefore, this bit has no effect on the hardware. It is implemented as read/write for software compatibility. |
| 10 | Discard Timer Status (DTS) — RO. Hardwired to 0. ICH5 only performs delayed transactions on behalf of PCI memory reads to prefetchable memory. |
| 9 | **Secondary Discard Timer (SDT)** — R/W. This bit sets the maximum number of PCI clock cycles that the ICH5 waits for an initiator on PCI to repeat a delayed transaction request. The counter starts once the delayed transaction completion is at the head of the queue. If the master has not repeated the transaction at least once before the counter expires, the ICH5 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 — RO. Hardwired to 0. The PCI logic will not generate fast back-to-back cycles on the PCI bus. |
| 6 | Secondary Bus Reset — RO. hardwired to 0. The ICH5 does not follow the PCI-to-PCI bridge reset scheme; Software-controlled resets are implemented in the PCI-LPC device. |
| 5 | **Master Abort Mode** — R/W. This bit is R/W for software compatibility and has no affect on ICH5 behavior. |
| 4 | **VGA 16-Bit Decode.** This bit does not have any functionality relative to address decodes because the ICH5 forwards the cycles to PCI, independent of the decode. Writes of 1 have no impact other than to force the bit to 1. Writes of 0 have no impact other than to force the bit to 0. Reads to this bit will return the previously written value (or 0 if no writes since reset). |
| 3 | **VGA Enable** — R/W.<br>0 = No VGA device on PCI.<br>1 = Indicates that the VGA device is on PCI. Therefore, the PCI to hub interface decoder will not accept memory cycles in the range A0000h–BFFFFh. Note that the ICH5 will never take I/O cycles in the VGA range from PCI. |
| 2 | **ISA Enable** — R/W. The ICH5 ignores this bit. However, this bit is read/write for software compatibility. Since the ICH5 forwards all I/O cycles that are not in the USB, AC ’97, or IDE ranges to PCI, this bit would have no effect. |
| 1 | **SERR# Enable** — R/W.<br>0 = Disable<br>1 = Enable. If this bit is set AND bit 8 in PCICMD register (D30:F0 Offset 04h) is also set, the ICH5 will set the SSE bit in PCISTS register (D30:F0, offset 06h, bit 14) **and** also generate an NMI (or SMI# if NMI routed to SMI) when the SERR# signal is asserted.<br><br>**NOTE:** The internal SERR# will be generated only if the SERR_EN (D30:F0:04h bit 8) bit is also set. |
| 0 | **Parity Error Response Enable** — R/W.<br>0 = Disable<br>1 = Enable the hub interface to PCI bridge for parity error detection and reporting on the PCI bus. |

## 8.1.25 HI1_CMD—Hub Interface 1 Command Control Register (HUB-PCI—D30:F0)

Offset Address: 40–43h
Default Value: 76202802h
Attribute: R/W, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:21 | Reserved |
| 20 | **HP Unsupported (HPUN)** — R/W.<br>1 = Intel® ICH5 will not check parity on the hub interface even if enabled to do so according to the Parity Error Response bit in D30:F0:04h bit 6. |
| 19:16 | **Hub Interface Timeslice (HI_TMSL)** — R/W. This field sets the HI arbiter time-slice value with 4 base-clock granularity. A value of 0h means that the time-slice is immediately expired and that the ICH5 will allow the other master's request to be serviced after every message. |
| 15:14 | Hub Interface Width (HI_Width) — RO. This field is hardwired to 00b, indicating that the hub interface is 8 bits wide. |
| 13 | Hub Interface Rate Valid (HI_Rate_Val) — RO. Hardwired to 1. |
| 12:10 | **Hub Interface Rate (HI_Rate)** — RO. Encoded value representing the clock-to-transfer rate of the HI1 interface:<br>1:4 = 010b<br>The value is loaded at reset by sampling the capability of the device connected to the HI1 port. The value for this field is fixed for 4X mode only. |
| 9:4 | Reserved. |
| 3:1 | Max Data (MAXD) — RO. Hardwired to 001b. This field specifies the maximum amount of data that the ICH5 is allowed to burst in one packet on the hub interface. The ICH5 will always perform 64-byte bursts. |
| 0 | Reserved |

## 8.1.26 DEVICE_HIDE—Secondary PCI Device Hiding Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 44–45h | Attribute: | R/W |
| Default Value: | 00h | Size: | 16 bits |
| Power Well: | 00h | | |

This register allows software to "hide" PCI devices (0 through 5) in terms of configuration space. Specifically, when PCI devices (0–5) are hidden, the configuration space is not accessible because the PCI IDSEL pin does not assert. The ICH5 supports the hiding of six external devices (0 through 5), which matches the number of PCI request/grant pairs, and the ability to hide the integrated LAN device by masking out the configuration space decode of LAN controller. Writing a 1 to this bit will not restrict the configuration cycle to the PCI bus. This differs from bits 0 through 5 in which the configuration cycle is restricted.

Hiding a PCI device can be useful for debugging, bug work-arounds, and system management support. Devices should only be hidden during initialization before any configuration cycles are run. This guarantees that the device is not in a semi-enable state.

| Bit | Description |
|---|---|
| 15:9 | Reserved |
| 8 | **Hide Device 8 (HIDE_DEV8)** — R/W. Same as bit 0 of this register, except for device 8 (AD24), which is hardwired to the integrated LAN device.<br>This bit will not change the way the configuration cycle appears on PCI bus |
| 7:6 | Reserved |
| 5 | **Hide Device 5 (HIDE_DEV5)** — R/W. Same as bit 0 of this register, except for device 5 (AD21). |
| 4 | **Hide Device 4 (HIDE_DEV4)** — R/W. Same as bit 0 of this register, except for device 4 (AD20). |
| 3 | **Hide Device 3 (HIDE_DEV3)** — R/W. Same as bit 0 of this register, except for device 3 (AD19). |
| 2 | **Hide Device 2 (HIDE_DEV2)** — R/W. Same as bit 0 of this register, except for device 2 (AD18). |
| 1 | **Hide Device 1 (HIDE_DEV1**) — R/W. Same as bit 0 of this register, except for device 1 (AD17). |
| 0 | **Hide Device 0 (HIDE_DEV0)** — R/W.<br>0 = The PCI configuration cycles for this slot are not affected.<br>1 = Intel® ICH5 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. AD16 is used as IDSEL for device 0. |

## 8.1.27 CNF—Policy Configuration Register (HUB-PCI—D30:F0)

Offset Address: 50–53h
Default Value: 00406402h
Attribute: R/W
Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | Reserved |
| 23:20 | **Async Reads** — R/W.<br>ICH5 memory async read request. BIOS should set this value to 0111b. |
| 19 | BIOS should set this bit if the platform uses HI11. |
| 18 | BIOS should set this bit if the platform uses HI11. |
| 17:16 | **PCI Prefetch** — R/W.<br>PCI Prefetch request for PCI reads from main memory. BIOS should set this value to 11b. |
| 15:14 | Reserved |
| 13 | **Prefetch Flush Enable** — R/W.<br>0 = Prefetch Flush Disable<br>1 = Causes CPU to PCI logic to only deliver “Demand” data for a delayed transaction if a processor-to-PCI write has occurred since the delayed transaction was initiated. (Default)<br><br>**NOTE:** This bit must be set by system BIOS. |
| 12:10 | Reserved |
| 9 | **High Priority PCI Enable (HP_PCI_EN)** — R/W.<br>0 = All PCI REQ#/GNT pairs have the same arbitration priority.<br>1 = Enables a mode where the REQ0#/GNT0# signal pair has a higher arbitration priority. |
| 8 | **Hole Enable (15 MB–16 MB)** — R/W.<br>0 = Disable<br>1 = Enables the 15-MB to 16-MB hole in the DRAM. |
| 7:3 | Reserved |
| 2 | **Delayed Transaction Discard Timer** — R/W.<br>When set to 1 this bit shortens all delayed transaction discard timers from 32 µs to 4 µs.<br>**NOTE:** Setting this bit may improve system performance issues with certain non-optimally behaved PCI devices, but may violate the *PCI-to-PCI Bridge Architecture Specification, Revision 1.1* (section 5.3.2) |
| 1 | **12-Clock Retry Enable** — R/W. System BIOS must set this bit for PCI compliance.<br>0 = The Intel® ICH5 inserts as many wait-states as needed to complete the PCI to memory cycle.<br>1 = The ICH5 retries a PCI to memory cycle (reads or write) if the ICH5 is not able to complete the transfer in 12 PCI clocks. |
| 0 | Reserved |

## 8.1.28 MTT—Multi-Transaction Timer Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 70h | Attribute: | R/W |
| Default Value: | 20h | Size: | 8 bits |

MTT is an 8-bit register that controls the amount of time that the ICH5's arbiter allows a PCI initiator to perform multiple back-to-back transactions on the PCI bus. The ICH5's MTT mechanism is used to guarantee a fair share of the Primary PCI bandwidth to an initiator that performs multiple back-to-back transactions to fragmented memory ranges (and as a consequence it can not use long burst transfers).

The number of clocks programmed in the MTT represents the guaranteed time slice (measured in PCI clocks) allotted to the current agent, after which the arbiter will grant another agent that is requesting the bus. The MTT value must be programmed with 8-clock granularity in the same manner as MLT. For example, if the MTT is programmed to 18h, then the selected value corresponds to the time period of 24 PCI clocks.The default value of MTT is 20h (32 PCI clocks).

***Note:*** Programming the a value of 00h disables this function, which could cause starvation problems for some PCI master devices. Programming of the MTT to anything less than 16 clocks will not allow the Grant-to-FRAME# latency to be 16 clocks. The MTT timer will timeout before the Grant-to-FRAME# trigger causing a re-arbitration.

| Bit | Description |
|---|---|
| 7:3 | **Multi-Transaction Timer Count Value** — R/W. This field specifies the amount of time that grant will remain asserted to a master continuously asserting its request for multiple transfers. This field specifies the count in an 8-clock (PCI clock) granularity. |
| 2:0 | Reserved |

## 8.1.29 PCI_MAST_STS—PCI Master Status Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 82h | Attribute: | R/WC |
| Default Value: | 00h | Size: | 8 bits |

***Note:*** Software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

| Bit | Description |
|---|---|
| 7 | **Internal PCI Master Request Status (INT_MREQ_STS)** — R/WC.<br>0 = DMA controller or LPC has **not** requested use of the PCI bus.<br>1 = The Intel® ICH5's internal DMA controller or LPC has requested use of the PCI bus. |
| 6 | **Internal LAN Master Request Status (LAN_MREQ_STS)** — R/WC.<br>0 = LAN controller has **not** requested use of the PCI bus.<br>1 = The ICH5's internal LAN controller has requested use of the PCI bus. |
| 5:0 | **PCI Master Request Status (PCI_MREQ_STS)** — R/WC. Allows software to see if a particular bus master has requested use of the PCI bus. For example, bit 0 will be set if the ICH5 has detected REQ0# asserted and bit 5 will be set if ICH5 detected REQ5# asserted.<br>0 = Associated PCI master has **not** requested use of the PCI bus.<br>1 = The associated PCI master has requested use of the PCI bus. |

## 8.1.30 ERR_CMD—Error Command Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 90h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

This register configures the ICH5's Device 30 responses to various system errors. The actual assertion of the internal SERR# (routed to cause NMI# or SMI#) is enabled via the PCI Command register.

| Bit | Description |
|---|---|
| 7:3 | Reserved |
| 2 | **SERR# Enable on Receiving Target Abort (SERR_RTA_EN)** — R/W.<br>0 = Disable<br>1 = Enable. When SERR_EN is set, the Intel® ICH5 will report SERR# when SERR_RTA is set. |
| 1:0 | Reserved |

## 8.1.31 ERR_STS—Error Status Register (HUB-PCI—D30:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 92h | Attribute: | R/WC |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

This register records the cause of system errors in Device 30. The actual assertion of SERR# is enabled via the PCI Command register.

***Note:*** Software must write a 1 to clear bits that are set. Writing a 0 to the bit has no effect.

| Bit | Description |
|---|---|
| 7:3 | Reserved |
| 2 | **SERR# Due to Received Target Abort (SERR_RTA)** — R/WC.<br>0 = Target abort **not** received.<br>1 = Intel® ICH5 received a target abort. If SERR_EN, the ICH5 will also generate an SERR# when SERR_RTA is set. |
| 1 | Reserved |
| 0 | **PCI Parity Inversion State (PAR_INV)** — R/WC.<br>0 = No parity errors on PCI.<br>1 = Parity errors may have occurred on PCI. This bit can be checked as part of the NMI# service routine. |

# LPC Interface Bridge Registers (D31:F0) 9

The LPC Bridge function of the ICH5 resides in PCI Device 31:Function 0. This function contains many other functional units (e.g., 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.

## 9.1 PCI Configuration Registers (LPC I/F—D31:F0)

***Note:*** Address locations that are not shown in Table 141 should be treated as Reserved (See Section 6.2 for details).

**Table 141. LPC Interface PCI Register Address Map (LPC I/F—D31:F0) (Sheet 1 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086h | RO |
| 02–03h | DID | Device Identification | 24D0h | RO |
| 04–05h | PCICMD | PCI Command | 000Fh | R/W, RO |
| 06–07h | PCISTS | PCI Status | 0280h | 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 |
| 0Eh | HEADTYP | Header Type | 80h | RO |
| 40–43h | PMBASE | ACPI Base Address | 00000001h | R/W, RO |
| 44h | ACPI_CNTL | ACPI Control | 00h | R/W |
| 4E–4Fh | BIOS_CNTL | BIOS Control | 0000h | R/W |
| 54h | TCO_CNTL | TCO Control | 00h | R/W |
| 58–5Bh | GPIO_BASE | GPIO Base Address | 00000001h | R/W, RO |
| 5Ch | GPIO_CNTL | GPIO Control | 00h | R/W |
| 60–63h | PIRQ[*n*]_ROUT | PIRQ[A–D] Routing Control | 80h | R/W |
| 64h | SIRQ_CNTL | Serial IRQ Control | 10h | R/W |
| 68–6Bh | PIRQ[*n*]_ROUT | PIRQ[E–H] Routing Control | 80h | R/W |
| 88h | D31_ERR_CFG | Device 31 Error Configuration | 00h | R/W |
| 8Ah | D31_ERR_STS | Device 31 Error Status | 00h | R/WC |
| 90–91h | PCI_DMA_CFG | PCI DMA Configuration | 0000h | R/W |

**Table 141. LPC Interface PCI Register Address Map (LPC I/F—D31:F0) (Sheet 2 of 2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| A0–CFh | | Power Management (See Section 9.8.1) | | |
| D0–D3h | GEN_CNTL | General Control | 00000004h | R/W, R/WO |
| D4h | GEN_STA | General Status | 0Xh | R/W (special), RO |
| D5h | BACK_CNTL | Backed Up Control | See register description | R/W, RO |
| D8h | RTC_CONF | Real Time Clock Configuration | 00h | R/W |
| E0h | COM_DEC | LPC I/F COM Port Decode Ranges | 00h | R/W |
| E1h | LPCFDD_DEC | LPC I/F FDD and LPT Decode Ranges | 00h | R/W |
| E3h | FB_DEC_EN1 | Flash BIOS Decode Enable 1 | FFh | R/W |
| E4–E5h | GEN1_DEC | LPC I/F Generic 1 Decode Range 1 | 0000h | R/W |
| E6–E7h | LPC_EN | LPC I/F Enables | 00h | R/W |
| E8–EBh | FB_SEL1 | Flash BIOS Select 1 | 00112233h | R/W, RO |
| EC–EDh | GEN2_DEC | LPC I/F Generic 2 Decode Range 2 | 0000h | R/W |
| EE–EFh | FB_SEL2 | Flash BIOS Select 2 | 4567h | R/W |
| F0h | FB_DEC_EN2 | Flash BIOS Decode Enable 2 | 0Fh | R/W |
| F2h | FUNC_DIS | Function Disable | 00h | R/W |

**NOTE:** Refer to the latest *Intel® ICH5 / ICH5R Specification Update* for the value of the Revision Identification register.

## 9.1.1 VID—Vendor Identification Register (LPC I/F—D31:F0)

Offset Address: 00–01h
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 |

## 9.1.2 DID—Device Identification Register (LPC I/F—D31:F0)

Offset Address: 02–03h
Default Value: 24D0h
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 Intel® ICH5 LPC bridge. |

## 9.1.3 PCICMD—PCI COMMAND Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 04–05h | Attribute: | R/W, RO |
| Default Value: | 000Fh | 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.<br>0 = Disable<br>1 = Enable. Allow SERR# to be generated. |
| 7 | Wait Cycle Control (WCC) — RO. Hardwired to 0. |
| 6 | **Parity Error Response (PER) —** R/W.<br>0 = No action is taken when detecting a parity error.<br>1 = The Intel® ICH5 will take normal action 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 1. |
| 2 | Bus Master Enable (BME) — RO. Hardwired to 1 to indicate that bus mastering cannot be disabled for function 0 (DMA/ISA Master). |
| 1 | Memory Space Enable (MSE) — RO. Hardwired to 1 to indicate that memory space cannot be disabled for Function 0 (LPC I/F). |
| 0 | I/O Space Enable (IOSE) — RO. Hardwired to 1 to indicate that the I/O space cannot be disabled for function 0 (LPC I/F). |

## 9.1.4 PCISTS—PCI Status Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 06–07h | Attribute: | RO, R/WC |
| Default Value: | 0280h | Size: | 16-bit |
| Lockable: | No | 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.<br>0 = PERR# **not** active<br>1 = PERR# signal goes active. Set even if the PER bit is 0. |
| 14 | **Signaled System Error (SSE)**— R/WC.<br>0 = SERR# on function 0 **not** generated with SERR_EN set.<br>1 = Set by the Intel® ICH5 if the SERR_EN bit is set and the ICH5 generates an SERR# on function 0. The ERR_STS register can be read to determine the cause of the SERR#. The SERR# can be routed to cause SMI#, NMI, or interrupt. |
| 13 | **Master Abort Status (RMA)** — R/WC.<br>0 = Master abort on PCI **not** generated due to LPC I/F master or DMA cycle.<br>1 = ICH5 generated a master abort on PCI due to LPC I/F master or DMA cycles. |
| 12 | **Received Target Abort (RTA)** — R/WC.<br>0 = Target abort **not** received during LPC I/F master or DMA cycles to PCI.<br>1 = ICH5 received a target abort during LPC I/F master or DMA cycles to PCI. |
| 11 | **Signaled Target Abort (STA)** — R/WC.<br>0 = Target abort **not** generated on PCI cycles claimed by ICH5 for conditions listed below.<br>1 = ICH5 generated a target abort condition on PCI cycles claimed by the ICH5 for ICH5 internal registers or for going to LPC I/F. |
| 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>- The ICH5 is the initiator of the cycle,<br>- The ICH5 asserted PERR# (for reads) or observed PERR# (for writes), and<br>- The PER bit is set. |
| 7 | Fast Back to Back Capable (FB2BC) — RO. Hardwired to 1 to indicate that the ICH5, as a target, can accept fast back-to-back transactions. |
| 6 | User Definable Features (UDF). Hardwired to 0. |
| 5 | 66 MHz Capable (66MHZ_CAP) — RO. Hardwired to 0. |
| 4:0 | Reserved. |

## 9.1.5 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 — RO. 8-bit value that indicates the revision number for the LPC bridge. For the A-0 stepping, this value is 00h.<br><br>**NOTE:** Refer to the latest Intel® ICH5 / ICH5R Specification Update for the value of the Revision Identification register. |

## 9.1.6 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. |

## 9.1.7 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 PCI bridge. |

## 9.1.8 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. |

## 9.1.9 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. |

## 9.1.10 PMBASE—ACPI Base Address Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 40–43h | 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. |

## 9.1.11 ACPI_CNTL—ACPI Control Register (LPC I/F — D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 44h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bit |
| Lockable: | No | Usage: | ACPI, Legacy |
| | | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:5 | Reserved |
| 4 | **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. |
| 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.<br>**Bits** — **SCI Map**<br>000 — IRQ9<br>001 — IRQ10<br>010 — IRQ11<br>011 — Reserved<br>100 — IRQ20 (Only available if APIC enabled)<br>101 — IRQ21 (Only available if APIC enabled)<br>110 — IRQ22 (Only available if APIC enabled)<br>111 — IRQ23 (Only available if APIC enabled)<br>**NOTE:** When the TCO interrupt is mapped to APIC interrupts 9, 10 or 11, the signal is in fact active high. When the TCO interrupt is mapped to IRQ 20, 21, 22, or 23, the signal is active low and can be shared with PCI interrupts that may be mapped to those same signals (IRQs). |

## 9.1.12 BIOS_CNTL—BIOS Control Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 4E–4Fh | Attribute: | R/W |
| Default Value: | 0000h | Size: | 16 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:2 | Reserved |
| 1 | **BIOS Lock Enable (BLE)** — R/W.<br>0 = Setting the BIOSWE will not cause Sums. Once set, this bit can only be cleared by a PCIRST#.<br>1 = Enables setting the BIOSWE bit to cause Sums. |
| 0 | **BIOS Write Enable (BIOSWE)** — R/W.<br>0 = Only read cycles result in flash BIOS I/F cycles.<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. |

## 9.1.13 TCO_CNTL — TCO Control Register (LPC I/F — D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 54h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:4 | Reserved |
| 3 | **TCO Interrupt Enable (TCO_INT_EN)** — R/W. This bit enables/disables the TCO interrupt.<br>0 = Disables TCO interrupt.<br>1 = Enables TCO Interrupt, as selected by the TCO_INT_SEL field. |
| 2:0 | **TCO Interrupt Select (TCO_INT_SEL)** — R/W. This field 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. Note that if the TCOSCI_EN bit is set (bit 6 of the GPEO_EN register), then the TCO interrupt will be sent to the same interrupt as the SCI, and the TCO_INT_SEL bits will have no meaning. When the TCO interrupt is mapped to APIC interrupts 9, 10 or 11, the signal is in fact active high. When the TCO interrupt is mapped to IRQ 20, 21, 22, or 23, the signal is active low and can be shared with PCI interrupts that may be mapped to those same signals (IRQs).<br>**Bits** **SCI Map**<br>000 IRQ9<br>001 IRQ10<br>010 IRQ11<br>011 Reserved<br>100 IRQ20 (Only available if APIC enabled)<br>101 IRQ21 (Only available if APIC enabled)<br>110 IRQ22 (Only available if APIC enabled)<br>111 IRQ23 (Only available if APIC enabled) |

## 9.1.14 GPIO_BASE—GPIO Base Address Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 58h–5Bh | Attribute: | R/W, RO |
| Default Value: | 00000001h | Size: | 32 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 31:16 | Reserved |
| 15:6 | **Base Address** — R/W. This field provides the 64 bytes of I/O space for GPIO. |
| 5:1 | Reserved |
| 0 | Resource Type Indicator — RO. Hardwired to 1 to indicate I/O space. |

## 9.1.15 GPIO_CNTL—GPIO Control Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 5Ch | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:5 | Reserved |
| 4 | **GPIO Enable (GPIO_EN)** — R/W. This bit enables/disables decode of the I/O range pointed to by the GPIO base register and enables/disables the GPIO function.<br>0 = Disable<br>1 = Enable |
| 3:0 | Reserved |

## 9.1.16 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.)<br>0000 = Reserved<br>0001 = Reserved<br>0010 = Reserved<br>0011 = IRQ3<br>0100 = IRQ4<br>0101 = IRQ5<br>0110 = IRQ6<br>0111 = IRQ7<br>1000 = Reserved<br>1001 = IRQ9<br>1010 = IRQ10<br>1011 = IRQ11<br>1100 = IRQ12<br>1101 = Reserved<br>1110 = IRQ14<br>1111 = IRQ15 |

## 9.1.17 SIRQ_CNTL—Serial IRQ Control Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 64h | Attribute: | R/W |
| 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 Intel® ICH5 not recognizing SERIRQ interrupts. |
| 5:2 | **Serial IRQ Frame Size (SIRQSZ)** — R/W. Fixed field that indicates the size of the SERIRQ frame. In the ICH5, this field needs to be programmed to 21 frames (0100). This is an offset from a base of 17 which is the smallest data frame size. |
| 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 ICH5 will drive the start frame for the number of clocks specified. In quiet mode, the ICH5 will drive the start frame for the number of clocks specified minus 1, as the first clock was driven by the peripheral.<br>00 = 4 clocks<br>01 = 6 clocks<br>10 = 8 clocks<br>11 = Reserved |

## 9.1.18 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 |
| 3:0 | **IRQ Routing** — R/W. (ISA compatible.)<br>0000 = Reserved<br>0001 = Reserved<br>0010 = Reserved<br>0011 = IRQ3<br>0100 = IRQ4<br>0101 = IRQ5<br>0110 = IRQ6<br>0111 = IRQ7<br>1000 = Reserved<br>1001 = IRQ9<br>1010 = IRQ10<br>1011 = IRQ11<br>1100 = IRQ12<br>1101 = Reserved<br>1110 = IRQ14<br>1111 = IRQ15 |

## 9.1.19 D31_ERR_CFG—Device 31 Error Configuration Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 88h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bit |
| Lockable: | No | Power Well: | Core |

This register configures the ICH5's Device 31 responses to various system errors. The actual assertion of SERR# is enabled via the PCI Command register.

| Bit | Description |
|---|---|
| 7:3 | Reserved |
| 2 | **SERR# on Received Target Abort Enable (SERR_RTA_EN)** — R/W.<br>0 = Disable. No SERR# assertion on Received Target Abort.<br>1 = Enable. Intel® ICH5 generates SERR# when SERR_RTA is set if SERR_EN is set. |
| 1 | **SERR# on Delayed Transaction Timeout Enable (SERR_DTT_EN)** — R/W.<br>0 = Disable. No SERR# assertion on Delayed Transaction Timeout.<br>1 = Enable. ICH5 generates SERR# when SERR_DTT bit is set if SERR_EN is set. |
| 0 | Reserved |

## 9.1.20 D31_ERR_STS—Device 31 Error Status Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 8Ah | Attribute: | R/WC |
| Default Value: | 00h | Size: | 8 bit |
| Lockable: | No | Power Well: | Core |

This register configures the ICH5's Device 31 responses to various system errors. The actual assertion of SERR# is enabled via the PCI Command register.

***Note:*** Software clears set bits in this register by writing a 1 to the bit position.

| Bit | Description |
|---|---|
| 7:3 | Reserved |
| 2 | **SERR# Due to Received Target Abort (SERR_RTA)** — R/WC.<br>0 = Target abort **not** received.<br>1 = The Intel® ICH5 sets this bit when it receives a target abort. If SERR_EN, the ICH5 also generates a SERR# when SERR_RTA is set. |
| 1 | **SERR# Due to Delayed Transaction Timeout (SERR_DTT)** — R/WC.<br>0 = PCI master does **not** violate return for data time described below.<br>1 = When a PCI master does not return for the data within 1 ms of the cycle's completion, the ICH5 clears the delayed transaction and sets this bit. If both SERR_DTT_EN and SERR_EN are set, then ICH5 also generates an SERR# when SERR_DTT is set. |
| 0 | Reserved |

## 9.1.21 PCI_DMA_CFG—PCI DMA Configuration Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 90h–91h | Attribute: | R/W |
| Default Value: | 0000h | Size: | 16 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:14 | **Channel 7 Select** — R/W.<br>00 = Reserved<br>01 = PC/PCI DMA<br>10 = Reserved<br>11 = LPC I/F DMA |
| 13:12 | **Channel 6 Select** — R/W. Same bit decode as for channel 7 |
| 11:10 | **Channel 5 Select** — R/W. Same bit decode as for channel 7 |
| 9:8 | Reserved |
| 7:6 | **Channel 3 Select** — R/W. Same bit decode as for channel 7 |
| 5:4 | **Channel 2 Select** — R/W. Same bit decode as for channel 7 |
| 3:2 | **Channel 1 Select** — R/W. Same bit decode as for channel 7 |
| 1:0 | **Channel 0 Select** — R/W. Same bit decode as for channel 7 |

## 9.1.22 GEN_CNTL — General Control Register (LPC I/F — D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | D0h – D3h | Attribute: | R/W, R/WO |
| Default Value: | 00000004h | Size: | 32 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 31:26 | Reserved |
| 25 | **REQ5#/GNT5# PC/PCI Protocol Select (PCPCIB_SEL)** — R/W.<br>0 = The REQ5#/GNT5# pins function as a standard PCI REQ/GNT signal pair.<br>1 = PCI REQ5#/GNT5# signal pair will use the PC/PCI protocol as REQB#/GNTB. The corresponding bits in the GPIO_USE_SEL register must also be set to a 0. If the corresponding bits in the GPIO_USE_SEL register are set to a 1, the signals will be used as a GPI and GPO. |
| 24 | **Hide ISA Bridge (HIDE_ISA)** — R/W.<br>0 = The Intel® ICH5 will not prevent AD22 from asserting during config cycles to the PCI-to-ISA bridge.<br>1 = Software sets this bit to 1 to disable configuration cycles from being claimed by a PCI-to-ISA bridge. This will prevent the OS PCI PnP from getting confused by seeing two ISA bridges. It is required for the ICH5 PCI address line AD22 to connect to the PCI-to-ISA bridge's IDSEL input. When this bit is set, the ICH5 will not assert AD22 during config cycles to the PCI-to-ISA bridge. |
| 23:22 | Reserved |
| 21 | Reserved |
| 20 | Reserved |
| 19:18 | **Scratchpad** — R/W. ICH5 does not perform any action on these bits. |
| 17 | **HPET Address Enable (HPET_ADDR_EN**) — R/W.<br>0 = Disable<br>1 = Enable. ICH5 decodes the High-Precision Event Timers Memory Address Range selected by bits 16:15 below. |
| 16:15 | **HPET Address Select (HPET_ADDR_SEL)** — R/W. This 2-bit field selects 1 of 4 possible memory address ranges for the High-Precision Event Timers functionality. The encodings are:<br>**Bits [16:15]Memory Address Range**<br>00 FED0_0000h – FED0_03FFh<br>01 FED0_1000h – FED0_13FFh<br>10 FED0_2000h – FED0_23FFh<br>11 FED0_3000h – FED0_33FFh |
| 14 | Reserved |
| 13 | **Coprocessor Error Enable (COPR_ERR_EN)** — R/W.<br>0 = FERR# will not generate IRQ13 nor IGNNE#.<br>1 = When FERR# is low, ICH5 generates IRQ13 internally and holds it until an I/O write to port F0h. It will also drive IGNNE# active. |
| 12 | **Keyboard IRQ1 Latch Enable (IRQ1LEN)** — R/W.<br>0 = IRQ1 will bypass the latch.<br>1 = The active edge of IRQ1 will be latched and held until a port 60h read. |
| 11 | **Mouse IRQ12 Latch Enable (IRQ12LEN)** — R/W.<br>0 = IRQ12 will bypass the latch.<br>1 = The active edge of IRQ12 will be latched and held until a port 60h read. |
| 10 | Reserved |

| Bit | Description |
|---|---|
| 9 | **Top_Swap Lock-Down** — R/WO. This bit can only be written from 0 to 1 once.<br>0 = A hardware reset is required to clear this bit.<br>1 = Prevents the top-swap bit from being changed. |
| 8 | **APIC Enable (APIC_EN)** — R/W.<br>0 = Disables internal I/O (x) APIC.<br>1 = Enables the internal I/O (x) APIC and its address decode.<br>The following behavioral rules apply for bits 8 and 7 in this register:<br>• Rule 1: If bit 8 is 0, then the ICH5 will not decode any of the registers associated with the I/O APIC or I/O (x) APIC. The state of bit 7 is “Don’t Care” in this case.<br>• Rule 2: If bit 8 is 1 and bit 7 is 0, then the ICH5 will decode the memory space associated with the I/O APIC, but not the extra registers associated I/O (x) APIC.<br>• Rule 3: If bit 8 is 1 and bit 7 is 1, then the ICH5 will decode the memory space associated with both the I/O APIC and the I/O (x) APIC. This also enables PCI masters to write directly to the register to cause interrupts (PCI Message Interrupt).<br>**NOTE:** There is no separate way to disable PCI Message Interrupts if the I/O (x) APIC is enabled. This is not considered necessary. |
| 7 | **System Bus Message Disable** — R/W.<br>0 = Has no effect. (Default)<br>1 = Disables the ICH5 IOAPIC controller from generating anymore system bus interrupt messages.<br>**NOTE:** It is possible for the ICH5 to deliver up to 1 system bus interrupt message from the time this configuration bit is set to 1. |
| 6 | **Alternate Access Mode Enable (ALTACC_EN)** — R/W.<br>0 = Disable (default). ALT access mode allows reads to otherwise unreadable registers and writes otherwise unwritable registers.<br>1 = Enable |
| 5:4 | Reserved |
| 3 | Reserved— RO. |
| 2 | **DMA Collection Buffer Enable (DCB_EN)** — R/W.<br>0 = DCB disabled.<br>1 = Enables DMA Collection Buffer (DCB) for LPC I/F and PC/PCI DMA. |
| 1 | **Delayed Transaction Enable (DTE)** — R/W.<br>0 = Disable<br>1 = Enable. ICH5 enables delayed transactions for internal register, flash BIOS and LPC I/F accesses. |
| 0 | **Positive Decode Enable (POS_DEC_EN)** — R/W.<br>0 = Disable. The ICH5 performs subtractive decode on the PCI bus and forwards the cycles to LPC I/F if not to an internal register or other known target on LPC I/F. Accesses to internal registers and to known LPC I/F devices are still positively decoded.<br>1 = Enables ICH5 to only perform positive decode on the PCI bus. |

## 9.1.23 GEN_STA—General Status Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | D4h | Attribute: | R/W (special), RO |
| Default Value: | 0Xh | Size: | 8 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:3 | Reserved |
| 2 | **Safe Mode (SAFE_MODE)** — RO.<br>0 = Intel® ICH5 sampled AC_SDOUT low on the rising edge of PWROK.<br>1 = ICH5 sampled AC_SDOUT high on the rising edge of PWROK. ICH5 will force FREQ_STRAP[3:0] bits to all 1s (safe mode multiplier). |
| 1 | **No Reboot (NO_REBOOT)** — R/W (special).<br>0 = Normal TCO Timer reboot functionality (reboot after 2nd TCO timeout). This bit cannot be set to 0 by software if the strap is set to No Reboot.<br>1 = ICH5 disables the TCO Timer system reboot feature. This bit is set either by hardware when SPKR is sampled high on the rising edge of PWROK, or by software writing a 1 to the bit. |
| 0 | Reserved |

## 9.1.24 BACK_CNTL—Backed Up Control Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | D5h | Attribute: | R/W, RO |
| Default Value: | 0Fh (upon RTCRST# assertion low)<br>2Fh (if Top Swap Strap is active) | Size: | 8 bit |
| Lockable: | No | Power Well: | RTC, Suspend (see bit details) |

| Bit | Description |
|---|---|
| 7 | 0 = Reserved. Hardwired to 0 forcing the reset state of the IDE pins to always be driven/tri-state (depending on the pin). |
| 6 | 0 = Reserved. Hardwired to 0 forcing the reset state of the IDE pins to always be driven/tri-state (depending on the pin). |
| 5 | **Top-Block Swap Mode (TOP_SWAP)** — R/W. If Intel® ICH5 is strapped for Top-Swap (GNTA# 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.<br>This bit can not be overwritten after the Top-Swap Lock-Down bit is set.<br>0 = ICH5 will not invert A16. This bit is cleared by RTCRST# assertion, but not by any other type of reset.<br>1 = ICH5 will invert A16 for cycles targeting flash BIOS space (does not affect access to flash BIOS feature space). |
| 4 | **Enables CPU BIST (CPU_BIST_EN)** — R/W.<br>0 = Disable<br>1 = The INIT# signal will be driven active when CPURST# is active. INIT# will go inactive with the same timings as the other processor interface signals (Hold Time after CPURST# inactive). Note that CPURST# is generated by the memory controller hub, but the ICH5 has a hub interface special cycle that allows the ICH5 to control the assertion/deassertion of CPURST#.<br><br>**NOTE:** This bit is in the Resume well and is reset by RSMRST#, but not by PCIRST# nor CF9h writes. |
| 3:0 | **CPU Frequency Strap (FREQ_STRAP[3:0])** — R/W. These bits determine the internal frequency multiplier of the processor. These bits can be reset to 1111 based on an external pin strap or via the RTCRST# input signal. Software must program this field based on the processor's specified frequency. Note that this field is only writable when the SAFE_MODE bit is cleared to 0, and SAFE_MODE is only cleared by PWROK rising edge. These bits are in the RTC well. |

## 9.1.25 RTC_CONF—Real Time Clock Configuration Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | D8h | Attribute: | R/W, R/WO |
| Default Value: | 00h | Size: | 8 bit |
| Lockable: | Yes | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:5 | Reserved |
| 4 | **Upper 128-byte Lock (U128LOCK)** — R/WO.<br>0 = Access to these bytes in the upper CMOS RAM range have not been locked.<br>1 = Locks reads and writes to bytes 38h–3Fh in the upper 128-byte bank of the RTC CMOS RAM. Write cycles to this range will have no effect and read cycles will not return any particular guaranteed value. This is a write once register that can only be reset by a hardware reset. |
| 3 | **Lower 128-byte Lock (L128LOCK)** — R/WO.<br>0 = Access to these bytes in the lower CMOS RAM range have not been locked.<br>1 = Locks reads and writes to bytes 38h–3Fh in the lower 128-byte bank of the RTC CMOS RAM. Write cycles to this range will have no effect and read cycles will not return any particular guaranteed value. This is a write once register that can only be reset by a hardware reset. |
| 2 | **Upper 128-byte Enable (U128E)** — R/W.<br>0 = Disable<br>1 = Enables access to the upper 128-byte bank of RTC CMOS RAM. |
| 1:0 | Reserved |

## 9.1.26 COM_DEC—LPC I/F Communication Port Decode Ranges Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | E0h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 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 |
| 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) |

## 9.1.27 LPCFDD_DEC—LPC I/F FDD and LPT Decode Ranges Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | E1h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:5 | Reserved |
| 4 | **FDD Decode Range** — R/W. Determines which range to decode for the FDD Port<br>0 = 3F0h – 3F5h, 3F7h (Primary)<br>1 = 370h – 2FFh (Secondary) |
| 3:2 | Reserved |
| 1:0 | **LPT Decode Range** — R/W. This field determines which range to decode for the LPTPort.<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 |

## 9.1.28 FB_DEC_EN1—Flash BIOS Decode Enable 1 Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | E3h | Attribute: | R/W |
| Default Value: | FFh | Size: | 8 bits |

This register determines which memory ranges will be decoded on the PCI bus and forwarded to the flash BIOS. The ICH5 will subtractively decode cycles on PCI unless POS_DEC_EN is set to 1.

| Bit | Description |
|---|---|
| 7 | **FB_F8_EN** — R/W. This bit enables decoding two 512-KB flash BIOS memory ranges, and one 128-KB memory range.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS<br>FFF80000h – FFFFFFFFh<br>FFB80000h – FFBFFFFFh<br>000E0000h – 000FFFFFh |
| 6 | **FB_F0_EN** — R/W. This bit enables decoding two 512-KB flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS:<br>FFF00000h – FFF7FFFFh<br>FFB00000h – FFB7FFFFh |
| 5 | **FB_E8_EN** — R/W. This bit enables decoding two 512-KB flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS:<br>FFE80000h – FFEFFFFh<br>FFA80000h – FFAFFFFFh |
| 4 | **FB_E0_EN** — R/W. This bit enables decoding two 512-KB flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS:<br>FFE00000h – FFE7FFFFh<br>FFA00000h – FFA7FFFFh |
| 3 | **FB_D8_EN** — R/W. This bit enables decoding two 512-KB flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS<br>FFD80000h – FFDFFFFFh<br>FF980000h – FF9FFFFFh |
| 2 | **FB_D0_EN** — R/W. This bit enables decoding two 512-KB flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS<br>FFD00000h – FFD7FFFFh<br>FF900000h – FF97FFFFh |
| 1 | **FB_C8_EN** — R/W. This bit enables decoding two 512-KB flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS<br>FFC80000h – FFCFFFFFh<br>FF880000h – FF8FFFFFh |
| 0 | **FB_C0_EN** — R/W. This bit enables decoding two 512-KB flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS<br>FFC00000h – FFC7FFFFh<br>FF800000h – FF87FFFFh |

## 9.1.29 GEN1_DEC—LPC I/F Generic Decode Range 1 Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | E4h – E5h | Attribute: | R/W |
| Default Value: | 0000h | Size: | 16 bit |
| Lockable: | Yes | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:7 | **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>Note that this generic decode is for I/O addresses only, not memory addresses. The size of this range is 128 bytes. |
| 6:1 | 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 |

## 9.1.30 LPC_EN—LPC I/F Enables Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | E6h – E7h | Attribute: | R/W |
| Default Value: | 00h | Size: | 16 bit |
| Lockable: | Yes | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:14 | Reserved |
| 13 | **CNF2_LPC_EN** — R/W.<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.<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.<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.<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.<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.<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. |

| Bit | Description |
|---|---|
| 7:4 | Reserved |
| 3 | **FDD_LPC_EN** — R/W.<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. |
| 2 | **LPT_LPC_EN** — R/W.<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. |
| 1 | **COMB_LPC_EN** — R/W.<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. |
| 0 | **COMA_LPC_EN** — R/W.<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. |

## 9.1.31 FB_SEL1—Flash BIOS Select 1 Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | E8h | Attribute: | R/W, RO |
| Default Value: | 00112233h | Size: | 32 bits |

| Bit | Description |
|---|---|
| 31:28 | **FB_F8_IDSEL** — RO. IDSEL for two, 512-KB flash BIOS 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 | **FB_F0_IDSEL** — R/W. IDSEL for two, 512-KB flash BIOS 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 | **FB_E8_IDSEL** — R/W. IDSEL for two, 512-KB flash BIOS 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 | **FB_E0_IDSEL** — R/W. IDSEL for two, 512-KB flash BIOS memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFE0 0000h – FFE7 FFFFh<br>FFA0 0000h – FFA7 FFFFh |
| 15:12 | **FB_D8_IDSEL** — R/W. IDSEL for two, 512-KB flash BIOS 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 | **FB_D0_IDSEL** — R/W. IDSEL for two, 512-KB flash BIOS 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 | **FB_C8_IDSEL** — R/W. IDSEL for two, 512-KB flash BIOS 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 | **FB_C0_IDSEL** — R/W. IDSEL for two, 512-KB flash BIOS memory ranges. The IDSEL programmed in this field addresses the following memory ranges:<br>FFC0 0000h – FFC7 FFFFh<br>FF80 0000h – FF87 FFFFh |

## 9.1.32 GEN2_DEC—LPC I/F Generic Decode Range 2 Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | ECh – EDh | Attribute: | R/W |
| Default Value: | 0000h | Size: | 16 bit |
| Lockable: | Yes | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:4 | **Generic I/O Decode Range 2 Base Address (GEN2_BASE)** — R/W. This address is aligned on a 64-byte boundary, and must have address lines 31:16 as 0.<br>Note that this generic decode is for I/O addresses only, not memory addresses. The size of this range is 16 bytes. |
| 3:1 | Reserved. Read as 0 |
| 0 | **Generic I/O Decode Range 2 Enable (GEN2_EN)** — R/W.<br>0 = Disable<br>1 = Accesses to the GEN2 I/O range will be forwarded to the LPC I/F |

## 9.1.33 FB_SEL2—Flash BIOS Select 2 Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | EEh–EFh | Attribute: | R/W |
| Default Value: | 4567h | Size: | 32 bits |

| Bit | Description |
|---|---|
| 15:12 | **FB_70_IDSEL** — R/W. IDSEL for two, 1-M flash BIOS memory ranges.<br>The IDSEL programmed in this field addresses the following memory ranges:<br>FF70 0000h – FF7F FFFFh<br>FF30 0000h – FF3F FFFFh |
| 11:8 | **FB_60_IDSEL** — R/W. IDSEL for two, 1-M flash BIOS memory ranges.<br>The IDSEL programmed in this field addresses the following memory ranges:<br>FF60 0000h – FF6F FFFFh<br>FF20 0000h – FF2F FFFFh |
| 7:4 | **FB_50_IDSEL** — R/W. IDSEL for two, 1-M flash BIOS memory ranges.<br>The IDSEL programmed in this field addresses the following memory ranges:<br>FF50 0000h – FF5F FFFFh<br>FF10 0000h – FF1F FFFFh |
| 3:0 | **FB_40_IDSEL** — R/W. IDSEL for two, 1-M flash BIOS memory ranges.<br>The IDSEL programmed in this field addresses the following memory ranges:<br>FF40 0000h – FF4F FFFFh<br>FF00 0000h – FF0F FFFFh |

## 9.1.34 FB_DEC_EN2—Flash BIOS Decode Enable 2 Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | F0h | Attribute: | R/W |
| Default Value: | 0Fh | Size: | 8 bits |

This register determines which memory ranges will be decoded on the PCI bus and forwarded to the flash BIOS. The ICH5 will subtractively decode cycles on PCI unless POS_DEC_EN is set to 1.

| Bit | Description |
|---|---|
| 7:4 | Reserved |
| 3 | **FB_70_EN** — R/W. Enables decoding two, 1-M flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS<br>FF70 0000h – FF7F FFFFh<br>FF30 0000h – FF3F FFFFh |
| 2 | **FB_60_EN** — R/W. Enables decoding two, 1-M flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS<br>FF60 0000h – FF6F FFFFh<br>FF20 0000h – FF2F FFFFh |
| 1 | **FB_50_EN** — R/W. Enables decoding two, 1-M flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS<br>FF50 0000h – FF5F FFFFh<br>FF10 0000h – FF1F FFFFh |
| 0 | **FB_40_EN** — R/W. Enables decoding two, 1-M flash BIOS memory ranges.<br>0 = Disable<br>1 = Enable the following ranges for the flash BIOS<br>FF40 0000h – FF4F FFFFh<br>FF00 0000h – FF0F FFFFh |

## 9.1.35 FUNC_DIS—Function Disable Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | F2h | Attribute: | R/W |
| Default Value: | 00h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15 | **D29_F7_Disable —** R/W. Software sets this bit to disable the USB EHCI controller function. BIOS must not enable I/O or memory address space decode, interrupt generation, or any other functionality of functions that are to be disabled.<br>0 = USB EHCI controller is enabled<br>1 = USB EHCI controller is disabled |
| 14 | **LPC Bridge Disable (D31F0D) —** R/W. Software sets this to 1 to disable the LPC bridge.<br>0 = Enable<br>1 = Disable. When disabled, the following spaces will no longer be decoded by the LPC bridge:<br>•Device 31, Function 0 Configuration space<br>•Memory cycles below 16 MB (100000h)<br>•I/O cycles below 64 KB (100h)<br>•The Internal I/OxAPIC at FEC0_0000 to FECF_FFFF<br>**NOTE:** 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 | **D29_F3_Disable —** R/W. Software sets this bit to disable the USB UHCI controller #4 function. BIOS must not enable I/O or memory address space decode, interrupt generation, or any other functionality of functions that are to be disabled.<br>0 = USB UHCI controller #4 is enabled<br>1 = USB UHCI controller #4 is disabled |
| 10 | **D29_F2_Disable —** R/W. Software sets this bit to disable the USB UHCI controller #3 function. BIOS must not enable I/O or memory address space decode, interrupt generation, or any other functionality of functions that are to be disabled.<br>0 = USB UHCI controller #3 is enabled<br>1 = USB UHCI controller #3 is disabled |
| 9 | **D29_F1_Disable —** R/W. Software sets this bit to disable the USB UHCI controller #2 function. BIOS must not enable I/O or memory address space decode, interrupt generation, or any other functionality of functions that are to be disabled.<br>0 = USB UHCI controller #2 is enabled<br>1 = USB UHCI controller #2 is disabled |
| 8 | **D29_F0_Disable —** R/W. Software sets this bit to disable the USB UHCI controller #1 function.BIOS must not enable I/O or memory address space decode, interrupt generation, or any other functionality of functions that are to be disabled.<br>0 = USB UHCI controller #1 is enabled<br>1 = USB UHCI controller #1 is disabled |
| 7 | Reserved |
| 6 | **D31_F6_Disable —** R/W. Software sets this bit to disable the AC ’97 modem controller function. BIOS must not enable I/O or memory address space decode, interrupt generation, or any other functionality of functions that are to be disabled.<br>0 = AC’97 Modem is enabled<br>1 = AC’97 Modem is disabled |

| Bit | Description |
|---|---|
| 5 | **D31_F5_Disable —** R/W. Software sets this bit to disable the AC '97 audio controller function. BIOS must not enable I/O or memory address space decode, interrupt generation, or any other functionality of functions that are to be disabled.<br>0 = AC '97 audio controller is enabled<br>1 = AC '97 audio controller is disabled |
| 4 | Reserved |
| 3 | **D31_F3_Disable —** R/W. Software sets this bit to disable the SMBus Host controller function. BIOS must not enable I/O or memory address space decode, interrupt generation, or any other functionality of functions that are to be disabled.<br>0 = SMBus controller is enabled<br>1 = SMBus controller is disabled |
| 2 | **D31_F2_Disable —** R/W. Software sets this bit to disable the SATA Host controller function. BIOS must not enable I/O or memory address space decode, interrupt generation, or any other functionality of functions that are to be disabled.<br>0 = SATA controller is enabled<br>1 = SATA controller is disabled |
| 1 | **D31_F1_Disable —** R/W. Software sets this bit to disable the IDE controller function. BIOS must not enable I/O or memory address space decode, interrupt generation, or any other functionality of functions that are to be disabled.<br>0 = IDE controller is enabled<br>1 = IDE controller is disabled |
| 0 | **SMB_FOR_BIOS —** R/W. This bit is used in conjunction with bit 3 in this register.<br>0 = No effect.<br>1 = Allows the SMBus I/O space to be accessible by software when bit 3 in this register is set. The PCI configuration space is hidden in this case. Note that if bit 3 is set alone, the decode of both SMBus PCI configuration and I/O space will be disabled. |

**NOTE:**
1. Software must always disable all functionality within the function before disabling the configuration space.
2. Configuration writes to internal devices, when the devices are disabled, are illegal and may cause undefined results.

## 9.2 DMA I/O Registers (LPC I/F—D31:F0)

**Table 142. 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 |
| 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 |

**Table 142. DMA Registers (Sheet 2 of 2)**

| Port | Alias | Register Name | Default | Type |
|---|---|---|---|---|
| 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 |

## 9.2.1 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 |

## 9.2.2 DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0 = 01h; Ch. #1 = 03h<br>Ch. #2 = 05h; Ch. #3 = 07h<br>Ch. #5 = C6h; Ch. #6 = CAh<br>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. |

## 9.2.3 DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0 = 87h; Ch. #1 = 83h<br>Ch. #2 = 81h; Ch. #3 = 82h<br>Ch. #5 = 8Bh; Ch. #6 = 89h<br>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. |

## 9.2.4 DMACMD—DMA Command Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0–3 = 08h;<br>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. |

## 9.2.5 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) |

## 9.2.6 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) |

## 9.2.7 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 = Illegal |
| 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) |

## 9.2.8 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. |

## 9.2.9 DMA Master Clear Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0–3 = 0Dh; 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. |

## 9.2.10 DMA_CLMSK—DMA Clear Mask Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Ch. #0–3 = 0Eh; 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. |

## 9.2.11 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>**NOTE:** Disabling channel 4 also disables channels 0–3 due to the cascade of channel's 0 – 3 through channel 4. |

## 9.3 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 |

## 9.3.1 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>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 | **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.

### 9.3.1.1 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. |

### 9.3.1.2 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. |

## 9.3.2 SBYTE_FMT—Interval Timer Status Byte Format Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Counter 0 = 40h,<br>Counter 1 = 41h,<br>Counter 2 = 42h | Attribute:<br>Size: | RO<br>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 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 |
| 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. |

### 9.3.3 Counter Access Ports Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | Counter 0 – 40h,<br>Counter 1 – 41h,<br>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. |

## 9.4 8259 Interrupt Controller (PIC) Registers (LPC I/F—D31:F0)

### 9.4.1 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 143 shows the different register possibilities for each address.

**Table 143. 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 |
| 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 (Section 5.8).

## 9.4.2 ICW1—Initialization Command Word 1 Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | Master Controller – 20h<br>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 000b |
| 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). |
| 2 | ADI — WO.<br>0 = Ignored for the Intel® ICH5. 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. |

## 9.4.3 ICW2—Initialization Command Word 2 Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | Master Controller – 21h<br>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:<br>**Code** / **Master Interrupt** / **Slave Interrupt**<br>000 / IRQ0 / IRQ8<br>001 / IRQ1 / IRQ9<br>010 / IRQ2 / IRQ10<br>011 / IRQ3 / IRQ11<br>100 / IRQ4 / IRQ12<br>101 / IRQ5 / IRQ13<br>110 / IRQ6 / IRQ14<br>111 / IRQ7 / IRQ15 |

## 9.4.4 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. |

## 9.4.5 ICW3—Slave Controller Initialization Command Word 3 Register (LPC I/F—D31:F0)

Offset Address: A1h
Default Value: All bits undefined
Attribute: WO
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. |

## 9.4.6 ICW4—Initialization Command Word 4 Register (LPC I/F—D31:F0)

Offset Address: Master Controller – 021h
Slave Controller – 0A1h
Attribute: WO
Size: 8 bits

| 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 Intel® ICH5. This is non-buffered mode. |
| 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. |

## 9.4.7 OCW1—Operational Control Word 1 (Interrupt Mask) Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | Master Controller – 021h<br>Slave Controller – 0A1h | Attribute:<br>Size: | R/W<br>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. |

## 9.4.8 OCW2—Operational Control Word 2 Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | Master Controller – 020h<br>Slave Controller – 0A0h | Attribute:<br>Size: | WO<br>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 |
| 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.<br>**Bits** **Interrupt Level** **Bits** **Interrupt Level**<br>000 IRQ0/8 100 IRQ4/12<br>001 IRQ1/9 101 IRQ5/13<br>010 IRQ2/10 110 IRQ6/14<br>011 IRQ3/11 111 IRQ7/15 |

## 9.4.9 OCW3—Operational Control Word 3 Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | Master Controller – 020h<br>Slave Controller – 0A0h | Attribute: | WO |
| | | 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 |

## 9.4.10 ELCR1—Master Controller Edge/Level Triggered Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 4D0h | Attribute: | R/W |
| Default Value: | 00h | 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. |

## 9.4.11 ELCR2—Slave Controller Edge/Level Triggered Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 4D1h | Attribute: | R/W |
| Default Value: | 00h | 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. |
| 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. |

# 9.5 Advanced Interrupt Controller (APIC)(D31:F0)

## 9.5.1 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 144.

**Table 144. 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_0020h | IRQPA | IRQ Pin Assertion | 32 bits | WO |
| FECO_0040h | EOIR | EOI | 32 bits | WO |

Table 145 lists the registers which can be accessed within the APIC via the Index Register. When accessing these registers, accesses must be done a DWord at a time. For example, software should never 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 145. 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 |
| 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 |

## 9.5.2 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 145. 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. |

## 9.5.3 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 pointed to by the Index register. |

## 9.5.4 IRQPA—IRQ Pin Assertion Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Memory Address | FEC0_0020h | Attribute: | WO |
| Default Value: | N/A | Size: | 32 bits |

The IRQ Pin Assertion Register is present to provide a mechanism to scale the number of interrupt inputs into the I/O APIC without increasing the number of dedicated input pins. When a device that supports this interrupt assertion protocol requires interrupt service, that device will issue a write to this register. Bits 4:0 written to this register contain the IRQ number for this interrupt. The only valid values are 0–23. Bits 31:5 are ignored. To provide for future expansion, peripherals should always write a value of 0 for Bits 31:5.

See Section 5.9.3 for more details on how PCI devices will use this field.

***Note:*** Writes to this register are only allowed by the processor and by masters on the ICH5's PCI bus. Writes by devices on PCI buses above the ICH5 (e.g., a PCI segment on a P64H2) are not supported.

| Bit | Description |
|---|---|
| 31:5 | Reserved. To provide for future expansion, the processor should always write a value of 0 to Bits 31:5. |
| 4:0 | **IRQ Number** — WO. Bits 4:0 written to this register contain the IRQ number for this interrupt. The only valid values are 0–23. |

## 9.5.5 EOIR—EOI Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Memory Address | FEC0_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 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 ICH5.

***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. |

## 9.5.6 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 |

## 9.5.7 VER—Version Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Index Offset: | 01h | Attribute: | RO |
| Default Value: | 00170002h | 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 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 Intel® ICH5 this field is hardwired to 17h to indicate 24 interrupts. |
| 15 | PRQ — RO. This bit is set to 1 to indicate that this version of the I/O APIC implements the IRQ Assertion register and allows PCI devices to write to it to cause interrupts. |
| 14:8 | Reserved |
| 7:0 | Version — RO. This is a version number that identifies the implementation version. |

## 9.5.8 REDIR_TBL—Redirection Table (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| Index Offset: | 10h–11h (vector 0) through 3E–3Fh (vector 23) | Attribute: | R/W, RO |
| Default Value: | Bit 16 = 1, Bits[15:12] = 0. 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 Front Side 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. |
| 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. |

| Bit | Description |
|---|---|
| 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 0s 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.

# 9.6 Real Time Clock Registers (LPC I/F—D31:F0)

## 9.6.1 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 146.

**Table 146. 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 ISA location for the real-time clock. The map for this bank is shown in Table 147. 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. 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. 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.

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 147.

**Table 147. 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 |

## 9.6.2 RTC_REGA—Register A (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| RTC Index: | 0Ah | 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 ICH5 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 492 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 |
| 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 never toggles<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 |

## 9.6.3 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 1, 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 RSMRST#, 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. |
| 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 Intel® ICH5. It is left in this register bank to provide compatibility with the Motorola 146818B. The ICH5 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 = 12-hour mode. In 12-hour mode, the seventh bit represents AM as 0 and PM as one.<br>1 = 24-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. 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. |

## 9.6.4 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 CH_IRQ_B signal 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. |
| 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. |

## 9.6.5 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 0s to mimic the functionality of the Motorola 146818B. These bits are not affected by RESET. |

# 9.7 Processor Interface Registers (LPC I/F—D31:F0)

Table 148 is the register address map for the processor interface registers.

**Table 148. 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 | WO |
| CF9h | RST_CNT | Reset Control | 00h | R/W |

## 9.7.1 NMI_SC—NMI Status and Control Register (LPC I/F—D31:F0)

I/O Address: 61h
Default Value: 00h
Lockable: No
Attribute: R/W, RO
Size: 8-bit
Power Well: Core

| Bit | Description |
|---|---|
| 7 | **SERR# NMI Source Status (SERR#_NMI_STS)** — RO.<br>1 = PCI agent detected a system error and pulses the PCI SERR# line. 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. |
| 6 | **IOCHK# NMI Source Status (IOCHK_NMI_STS)** — RO.<br>1 = An ISA agent (via SERIRQ) asserted IOCHK# on the ISA bus. This interrupt source is enabled by setting bit 3 to 0. To reset the interrupt, set bit 3 to 0 and then set it to 1. 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 |

## 9.7.2 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. |

## 9.7.3 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 Intel® ICH5 will force INIT# active for 16 PCI clocks. |

## 9.7.4 COPROC_ERR—Coprocessor Error Register (LPC I/F—D31:F0)

| | | | |
|---|---|---|---|
| I/O Address: | F0h | Attribute: | WO |
| Default Value: | 00h | Size: | 8-bits |
| Lockable: | No | Power Well: | Core |

| Bits | Description |
|---|---|
| 7:0 | **Coprocessor Error (COPROC_ERR)** — WO. 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. |

## 9.7.5 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 = Intel® ICH5 will keep SLP_S3#, SLP_S4# and SLP_S5# high.<br>1 = ICH5 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 SYSRESET#, 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 ICH5 performs a soft reset by activating INIT# for 16 PCI clocks.<br>1 = When RST_CPU bit goes from 0 to 1, the ICH5 performs a hard reset by activating PCIRST# for 1 millisecond. It also resets the resume well bits (except for those noted throughout the datasheet). The SLP_S3#, SLP_S4#, and SLP_S5# signals will not go active. |
| 0 | Reserved |

# 9.8 Power Management PCI Configuration 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.

Table 149 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 149. Power Management PCI Register Address Map (PM—D31:F0)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 40h–43h | ACPI_BASE | ACPI Base Address | 00000001h | R/W |
| 44h | ACPI_CNTL | ACPI Control | 00h | R/W |
| A0h | GEN_PMCON_1 | General Power Management Configuration 1 | 0000h | R/W, RO, R/WO |
| A2h | GEN_PMCON_2 | General Power Management Configuration 2 | 0000h | R/W, R/WC |
| A4h | GEN_PMCON_3 | General Power Management Configuration 3 | 00h | R/W, R/WC |
| A8h | STPCLK_DEL | Stop Clock Delay Register | 0Dh | R/W |
| ADh | USB_TDD | USB Transient Disconnect Detect | 00h | R/W |
| AEh | SATA_RD_CFG | SATA RAID Configuration (Intel® 82801ER ICH5R Only) | C0h | R/W |
| B8–BBh | GPI_ROUT | GPI Route Control | 00000000h | R/W |
| C0 | TRP_FWD_EN | I/O Monitor Trap Forwarding Enable | 00h | R/W (special) |
| C4–CAh | MON[*n*]_TRP_RNG | I/O Monitor[4:7] Trap Range | 0000h | R/W |
| CCh | MON_TRP_MSK | I/O Monitor Trap Range Mask | 0000h | R/W |

### 9.8.1 GEN_PMCON_1—General PM Configuration 1 Register (PM—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | A0h | Attribute: | R/W, RO, R/WO |
| Default Value: | 00h | Size: | 16-bit |
| Lockable: | No | Usage: | ACPI, Legacy |
| | | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:11 | Reserved |
| 10 | Reserved |
| 9 | **PWRBTN_LVL** — RO. This bit indicates the current state of the PWRBTN# signal.<br>0 = Low.<br>1 = High. |
| 8:7 | Reserved |
| 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. |
| 5 | **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.<br><br>**NOTE:** CPUSLP# will go active on entry to S3, S4 and S5 even if this bit is not set. |
| 4 | **SMI_LOCK** — R/WO. When this bit is set, writes to the GLB_SMI_EN bit 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 PCIRST#). |
| 3:2 | Reserved |
| 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 |

## 9.8.2 GEN_PMCON_2—General PM Configuration 2 Register (PM—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | A2h | Attribute: | R/W, R/WC |
| Default Value: | 00h | Size: | 8-bit |
| Lockable: | No | 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. |
| 6:5 | Reserved |
| 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 = Intel® ICH5 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><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 PCIRST# is inactive and CPUTHRMTRIP# goes active while the system is in an S0 or S1 state.<br><br>**NOTE:** This bit is also reset by RSMRST# and CF9h resets. It is not reset by the shutdown and reboot associated with the CPUTHRMTRIP# event. |
| 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. When exiting G3, the ICH5 begins the timer when the RSMRST# input deasserts. Note that this bit is functional regardless of the value in the SLP_S4# Assertion Stretch Enable.<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 clears this bit by writing a 1 to it.<br>1 = Indicates that the VRMPWRGD signal from the processor's VRM went low. |
| 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 Section 5.13.11.3 for more details about the PWROK pin functionality.<br>**NOTE:** In the case of true PWROK failure, PWROK will go low first before 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 ICH5.

## 9.8.3 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 |
| 5:4 | **SLP_S4# Minimum Assertion Width** — R/W. This field indicates the minimum assertion width of the SLP_S4# signal to guarantee 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** — RW.<br>1 = the SLP_S4# signal minimally assert for the time specified in bits 5:4 of this register.<br>0 = the SLP_S4# minimum assertion time is 1 to 2 RTCCLK.<br>This bit is cleared by RTCRST#. |
| 2 | **RTC_PWR_STSRTC 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. When the system boots, BIOS can detect that the FREQ_STRAP register contents are 1111 (the default when RTCRST# has been low). If this bit is also set, then BIOS knows the RTC battery had been removed. In that case, BIOS should take steps to reprogram the FREQ_STRAP register with the correct value, and then reboot the system. |
| 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>**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. |

**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 ICH5.

## 9.8.4 STPCLK_DEL—Stop Clock Delay Register (PM—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | A8h | Attribute: | R/W |
| Default Value: | 0Dh | Size: | 8-bit |
| Power Well: | Core | Usage: | ACPI Legacy |

| Bit | Description |
|---|---|
| 7:6 | Reserved |
| 5:0 | **STPCLK_DEL** — R/W. This field selects the value for t190 (CPUSLP# inactive to STPCLK# inactive). The default value of 0Dh yields a default of approximately 50.045 microseconds. The maximum value of 3Fh will result in a time of 245 microseconds.<br><br>**NOTE:** Software must program the value to a range that can be tolerated by the associated processor and chipset. The Intel® ICH5 requires that software does not program a value of 00h or 01h; a minimum programming of 02h yields the minimum possible delay of 3.87 microseconds. |

## 9.8.5 USB_TDD—USB Transient Disconnect Detect (PM—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | ADh | Attribute: | R/W |
| Default Value: | 00h | Size: | 8-bit |
| Power Well: | Resume | | |

| Bit | Description |
|---|---|
| 7:2 | Reserved |
| 1:0 | **Transient Disconnect Detect (TDD)**— R/W. This field prevents a short Single-Ended Zero (SE0) condition on the USB ports from being interrupted by the UHCI host controller as a disconnect. BIOS should set this field to 11b. |

## 9.8.6 SATA_RD_CFG—SATA RAID Configuration (PM—D31:F0) - (Intel® 82801ER ICH5R Only)

| | | | |
|---|---|---|---|
| Offset Address: | AEh | Attribute: | R/W |
| Default Value: | C0h | Size: | 8-bit |
| Power Well: | Resume | Usage: | ICH5R Only |

| Bit | Description |
|---|---|
| 7:6 | **Integrated SATA RAID Configuration (RAID_CFG)** — R/W. When cleared, Intel® RAID Technology is disabled. These bits are reset by the assertion of the RSMRST# pin. These bits are not reset when returning from S3.<br>00 = Intel RAID Technology Disabled<br>11 = Intel RAID Technology Enabled (default) |
| 5:0 | Reserved |

## 9.8.7 GPI_ROUT—GPI Routing Control Register (PM—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | B8h – BBh | Attribute: | R/W |
| Default Value: | 0000h | Size: | 32-bit |
| Lockable: | No | Power Well: | Resume |

| Bit | Description |
|---|---|
| 31:30 | **GPI15 Route** — R/W. See bits 1:0 for description. |
| **Same pattern for GPI14 through GPI3** | |
| 5:4 | **GPI2 Route** — R/W. See bits 1:0 for description. |
| 3:2 | **GPI1 Route** — R/W. See bits 1:0 for description. |
| 1:0 | **GPI0 Route** — R/W. GPIO[15:0] can be routed to cause an SMI or SCI when the GPI[n]_STS bit is set. If the GPIO 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 GPI can cause a Wake event, even if the GPI 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.

## 9.8.8 TRP_FWD_EN—IO Monitor Trap Forwarding Enable Register (PM—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | C0h | Attribute: | R/W (Special) |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Usage: | Legacy Only |
| Power Well: | Core | | |

The ICH5 uses this register to enable the monitors to forward cycles to LPC, independent of the POS_DEC_EN bit and the bits that enable the monitor to generate an SMI#. The only criteria is that the address passes the decoding logic as determined by the MON[*n*]_TRP_RNG and MON_TRP_MSK register settings.

| Bit | Description |
|---|---|
| 7 | **MON7_FWD_EN** — R/W.<br>0 = Disable. Cycles trapped by I/O Monitor 7 will not be forwarded to LPC.<br>1 = Enable. Cycles trapped by I/O Monitor 7 will be forwarded to LPC. |
| 6 | **MON6_FWD_EN** — R/W.<br>0 = Disable. Cycles trapped by I/O Monitor 6 will not be forwarded to LPC.<br>1 = Enable. Cycles trapped by I/O Monitor 6 will be forwarded to LPC. |
| 5 | **MON5_FWD_EN** — R/W.<br>0 = Disable. Cycles trapped by I/O Monitor 5 will not be forwarded to LPC.<br>1 = Enable. Cycles trapped by I/O Monitor 5 will be forwarded to LPC. |
| 4 | **MON4_FWD_EN** — R/W.<br>0 = Disable. Cycles trapped by I/O Monitor 4 will not be forwarded to LPC.<br>1 = Enable. Cycles trapped by I/O Monitor 4 will be forwarded to LPC. |
| 3:0 | Reserved |

## 9.8.9 MON[*n*]_TRP_RNG—I/O Monitor [4:7] Trap Range Register for Devices 4–7 (PM—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | C4h, C6h, C8h, CAh | Attribute: | R/W |
| Default Value: | 00h | Size: | 16 bits |
| Lockable: | No | Usage: | Legacy Only |
| Power Well: | Core | | |

These registers set the ranges that Device Monitors 4–7 should trap. Offset 4Ch corresponds to Monitor 4. Offset C6h corresponds to Monitor 5, etc.

If the trap is enabled in the MON_SMI register and the address is in the trap range (and passes the mask set in the MON_TRP_MSK register), the ICH5 will generate an SMI#. This SMI# occurs if the address is positively decoded by another device on PCI or by the ICH5 (because it would be forwarded to LPC or some other ICH5 internal registers). The trap ranges should not point to registers in the ICH5's internal IDE, USB, AC '97 or LAN I/O space. If the cycle is to be claimed by the ICH5 and targets one of the permitted ICH5 internal registers (interrupt controller, RTC, etc.), the cycle will complete to the intended target and an SMI# will be generated (this is the same functionality as the ICH component). If the cycle is to be claimed by the ICH5 and the intended target is on LPC, an SMI# will be generated but the cycle will only be forwarded to the intended target if forwarding to LPC is enabled via the TRP_FWD_EN register settings.

| Bit | Description |
|---|---|
| 15:0 | **MON[*n*]_TRAP_BASE** — R/W. Base I/O locations that MON[*n*] traps (where *n* = 4, 5, 6 or 7). The range can be mapped anywhere in the processor I/O space (0–64 KB).<br>Any access to the range will generate an SMI# if enabled by the associated DEV[*n*]_TRAP_EN bit in the MON_SMI register (PMBASE +40h). |

## 9.8.10 MON_TRP_MSK—I/O Monitor Trap Range Mask Register for Devices 4–7 (PM—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | CCh | Attribute: | R/W |
| Default Value: | 00h | Size: | 16 bits |
| Lockable: | No | Usage: | Legacy Only |
| Power Well: | Core | | |

| Bit | Description |
|---|---|
| 15:12 | **MON7_MASK** — R/W. This field selects low 4–bit mask for the I/O locations that MON7 will trap. Similar to MON4_MASK. |
| 11:8 | **MON6_MASK** — R/W. This field selects low 4–bit mask for the I/O locations that MON6 will trap. Similar to MON4_MASK. |
| 7:4 | **MON5_MASK** — R/W. This field selects low 4–bit mask for the I/O locations that MON5 will trap. Similar to MON4_MASK. |
| 3:0 | **MON4_MASK** — R/W. This field selects low 4–bit mask for the I/O locations that MON7 will trap. When a mask bit is set to a 1, the corresponding bit in the base I/O selection will not be decoded.<br>For example, if MON4_TRAP_BASE = 1230h, and MON4_MSK = 0011b, the Intel® ICH5 will decode 1230h, 1231h, 1232h, and 1233h for Monitor 4. |

## 9.9 APM I/O Decode

Table 150 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 150. 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 |

### 9.9.1 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 generate an SMI# when the APMC_EN bit is set. |

### 9.9.2 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). |

# 9.10 Power Management I/O Registers

Table 151 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 be compliant with the *Advanced Configuration and Power Interface, Version 2.0b,* 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 151. ACPI and Legacy I/O Register Map**

| PMBASE + Offset | Mnemonic | Register Name | ACPI Pointer | Default | Type |
|---|---|---|---|---|---|
| 00–01h | PM1_STS | PM1 Status | PM1a_EVT_BLK | 0000h | R/WC |
| 02–03h | PM1_EN | PM1 Enable | PM1a_EVT_BLK+2 | 0000h | R/W |
| 04–07h | PM1_CNT | PM1 Control | PM1a_CNT_BLK | 00000000h | R/W, WO |
| 08–0Bh | PM1_TMR | PM1 Timer | PMTMR_BLK | 00000000h | RO |
| 0C–0Fh | — | Reserved | — | — | — |
| 10h–13h | PROC_CNT | Processor Control | P_BLK | 00000000h | R/W, RO, WO |
| 14h–16h | — | Reserved | — | — | — |
| 17–1Fh | — | Reserved | — | — | — |
| 20h | — | Reserved | — | — | — |
| 28–2Bh | GPE0_STS | General Purpose Event 0 Status | GPE0_BLK | 00000000h | R/W, R/WC |
| 2C–2Fh | GPE0_EN | General Purpose Event 0 Enables | GPE0_BLK+4 | 00000000h | R/W |
| 30–33h | SMI_EN | SMI# Control and Enable | | 0000h | R/W, WO, R/W (special) |
| 34–37h | SMI_STS | SMI Status | | 0000h | R/WC, RO |
| 38–39h | ALT_GP_SMI_EN | Alternate GPI SMI Enable | | 0000h | R/W |
| 3A–3Bh | ALT_GP_SMI_STS | Alternate GPI SMI Status | | 0000h | R/WC |
| 3C–3Fh | — | Reserved | — | — | — |
| 40h | MON_SMI | Monitor SMI Status | | 0000h | R/W, R/WC |
| 42h–43h | | Reserved | | | |
| 44h | DEVACT_STS | Device Activity Status | | 0000h | R/WC |
| 48h | DEVTRAP_EN | Device Trap Enable | | 0000h | R/W |
| 50h | — | Reserved | — | — | — |
| 51h–5Fh | — | Reserved | — | — | — |
| 60h–7Fh | — | Reserved for TCO | — | — | — |

## 9.10.1 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 ICH5 will generate a Wake Event. Once back in an S0 state (or if already in an S0 state when the event occurs), the ICH5 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 Intel® ICH5 will transition the system to the ON state.<br>If the AFTERG3_EN bit is not set and a power failure 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). |
| 14:12 | 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. |
| 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 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 and PWRBTN_STS are both set, a wake event is generated. |
| 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 | Reserved |
| 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 is set, then the setting of the TMROF_STS bit will additionally generate an SCI or SMI# (depending on the SCI_EN). |

## 9.10.2 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 |
|---|---|
| 15:11 | 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 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 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 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 as described below:<br>**TMROF_EN / SCI_EN / Effect when TMROF_STS is set**<br>0 / x / No SMI# or SCI<br>1 / 0 / SMI#<br>1 / 1 / SCI |

## 9.10.3 PM1_CNT—Power Management 1 Control

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 04h (*ACPI PM1a_CNT_BLK*) | Attribute: | R/W, WO |
| Default Value: | 0000h | 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 |
|---|---|
| 15:14 | 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#.<br>000 = ON: Typically maps to S0 state.<br>001 = it asserts STPCLK#. Puts processor in Stop-Grant state. Optional to assert CPUSLP# to put processor in sleep state: Typically maps to S1 state.<br>010 = Reserved<br>011 = Reserved<br>100 = Reserved<br>101 = Suspend-To-RAM. Assert SLP_S3#: Typically maps to S3 state.<br>110 = Suspend-To-Disk. Assert SLP_S3# and SLP_S4#: Typically maps to S4 state.<br>111 = 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. |
| 1 | Reserved |
| 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. |

## 9.10.4 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.<br>Anytime bit 22 of the timer goes HIGH to LOW (bits referenced from 0 to 23), the TMROF_STS bit is set. The High-to-Low transition will occur every 2.3435 seconds. If the TMROF_EN bit is set, an SCI interrupt is also generated. |

## 9.10.5 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 |
| 17 | **Throttle Status (THTL_STS)** — RO.<br>0 = No clock throttling is occurring (maximum processor performance).<br>1 = Indicates that the clock state machine is in some type of low power state (where the processor is not running at its maximum performance): thermal throttling or hardware throttling. |
| 16:9 | Reserved |
| 8 | **Force Thermal Throttling (FORCE_THTL)** — R/W. Software can set this bit to force the thermal throttling function. This has the same effect as the THRM# signal being active for 2 seconds.<br>0 = No forced throttling.<br>1 = Throttling at the duty cycle specified in THRM_DTY starts immediately (no 2 second delay), and no SMI# is generated. |

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7:5</td><td>THRM_DTY — WO. This write-once field determines the duty cycle of the throttling when the thermal override condition occurs. 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.<br>There is no enable bit for thermal throttling, because it should not be disabled. Once the THRM_DTY field is written, any subsequent writes will have no effect until PCIRST# goes active.<br>THRM_DTY Throttle Mode PCI Clocks<br>000 50% (Default) 512<br>001 87.5% 896<br>010 75.0% 768<br>011 62.5% 640<br>100 50% 512<br>101 37.5% 384<br>110 25% 256<br>111 12.5% 128</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.<br>THTL_DTY Throttle Mode PCI Clocks<br>000 50% (Default) 512<br>001 87.5% 896<br>010 75.0% 768<br>011 62.5% 640<br>100 50% 512<br>101 37.5% 384<br>110 25% 256<br>111 12.5% 128</td></tr>
<tr><td>0</td><td>Reserved</td></tr>
</table>

## 9.10.6 GPE0_STS—General Purpose Event 0 Status Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 28h (*ACPI GPE0_BLK*) | Attribute: | R/W, 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. If the corresponding _EN bit is set, then when the _STS bit get set, the ICH5 will generate a Wake Event. Once back in an S0 state (or if already in an S0 state when the event occurs), the ICH5 will also generate an SCI if the SCI_EN bit is set, or an SMI# if the SCI_EN bit is not set. There will be no SCI/SMI# or wake event on THRMOR_STS since there is no corresponding _EN bit. None of these bits are reset by CF9h write. All are reset by RSMRST#.

| Bit | Description |
|---|---|
| 31:16 | **GPIn_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 GPI[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 GPI_ROUT bits for the corresponding GPI. |
| 15 | Reserved |
| 14 | **USB4_STS** — R/W.<br>0 = Disable<br>1 = Set by hardware and can be reset by writing a 1 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/W. This bit will be set to 1 by the Intel® ICH5 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/W.<br>0 = Disable<br>1 = Set by hardware and can be reset by writing a 1 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. |
| 10:9 | Reserved |

| Bit | Description |
|---|---|
| 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 ICH5's SMBus logic.<br>1 = Set by hardware to indicate that the wake event was caused by the ICH5'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>**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. |
| 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 two cases:<br>1.The PMEE bit for the function is set, and o The AC-link bit clock has been shut and the routed AC_SDIN line is high (for audio, if routing is disabled, no wake events are allowed.<br>2.For modem, if audio routing is disabled, then the wake event is an OR of all AC_SDIN lines. If routing is enabled, then the wake event for modem is the remaining non-routed AC_SDIN line), or o GPI Status Change Interrupt bit (NABMBAR + 30h, bit 0) is 1.<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. |

| Bit | Description |
|---|---|
| 2 | Reserved |
| 1 | **Thermal Interrupt Override Status** (**THRMOR_STS)** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = Thermal over-ride condition did **not** occur and start throttling the processor's clock at the THRM_DTY ratio<br>1 = This bit is set by hardware anytime a thermal over-ride condition occurs and starts throttling the processor's clock at the THRM_DTY ratio. This will not cause an SMI#, SCI, or wake event. |
| 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#). |

## 9.10.7 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, 12, 16–31 Resume, Bits 8–11, 13–15 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#. |
| 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. |

| Bit | Description |
|---|---|
| 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 | Reserved |
| 9 | Reserved |
| 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 |
| 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. |
| 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 | **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. |
| 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). |

## 9.10.8 SMI_EN—SMI Control and Enable Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 30h | Attribute: | R/W, R/W (special), WO |
| Default Value: | 0000h | Size: | 32 bit |
| Lockable: | No | Usage: | ACPI or Legacy |
| Power Well: | Core | | |

| Bit | Description |
|---|---|
| 31:19 | Reserved |
| 18 | **INTEL_USB2_EN** — R/W.<br>0 = Disable<br>1 = Enables Intel-Specific USB2 SMI logic to cause SMI#. |
| 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 Intel® ICH5 to generate an SMI# when the PERIODIC_STS bit is set in the SMI_STS register. |
| 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 ICH5 to trap accesses to the microcontroller range (62h or 66h) and generate an SMI#. Note that “trapped’ cycles will be claimed by the ICH5 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 1 is written to this bit position by BIOS software. |
| 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. |
| 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 Intel® ICH5 to assert SMI# low to the processor after SMI# has been asserted previously.<br>0 = Once the ICH5 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>**NOTE:** ICH5 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 ICH5. This bit is reset by a PCI reset event.<br>1 = Enables the generation of SMI# in the system upon any enabled SMI event. |

## 9.10.9 SMI_STS—SMI Status Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE + 34h | Attribute: | RO, R/WC |
| Default Value: | 0000h | 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 ICH5 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 ICH5 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 |
|---|---|
| 31:19 | 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, 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 Intel® ICH5 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. This is not a sticky bit.<br>1 = Indicates that the SMI# was caused by the SERIRQ decoder. |
| 14 | **PERIODIC_STS** — R/WC.<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 is also set, the ICH5 generates an SMI#. |
| 13 | **TCO_STS** — RO.<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. |
| 12 | **Device Monitor Status (DEVMON_STS)** — RO.<br>0 = SMI# not caused by Device Monitor.<br>1 = Set under any of the following conditions:<br>– Any of the DEV[7:4]_TRAP_STS bits are set and the corresponding DEV[7:4]_TRAP_EN bits are also set.<br>– Any of the DEVTRAP_STS bits are set and the corresponding DEVTRAP_EN bits are also set. |

| Bit | Description |
|---|---|
| 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). If this bit is set, and the MCSMI_EN bit is also set, the ICH5 will generate an SMI#. |
| 10 | **GPE0_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 in the GPE0_STS register that also have the corresponding bit set in the GPE0_EN register.<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. |
| 2 | **BIOS_STS** — R/WC.<br>0 = No SMI# generated due to ACPI software requesting attention.<br>1 = SMI# was generated due to ACPI software requesting attention (writing a 1 to the GBL_RLS bit with the BIOS_EN bit set). |
| 1:0 | Reserved |

## 9.10.10 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. |

## 9.10.11 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 GPIs.<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 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>All bits are in the resume well. Default for these bits is dependent on the state of the GPI pins. |

## 9.10.12 MON_SMI—Device Monitor SMI Status and Enable Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE +40h | Attribute: | R/W, R/WC |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | Usage: | Legacy Only |
| Power Well: | Core | | |

| Bit | Description |
|---|---|
| 15:12 | **DEV[7:4]_TRAP_STS** — R/WC. Bit 12 corresponds to Monitor 4, bit 13 corresponds to Monitor 5 etc.<br>0 = SMI# was not caused by the associated device monitor. Software clears this bit by writing a 1 to it.<br>1 = SMI# was caused by an access to the corresponding device monitor's I/O range. |
| 11:8 | **DEV[7:4]_TRAP_EN** — R/W. Bit 8 corresponds to Monitor 4, bit 9 corresponds to Monitor 5 etc.<br>0 = Disable<br>1 = Enables SMI# due to an access to the corresponding device monitor's I/O range. |
| 7:0 | Reserved |

## 9.10.13 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 | | |

This register is used in conjunction with the Periodic SMI# timer to detect any system activity for legacy power management.

***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. |
| 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 | **LEG_ACT_STS** — R/WC. Parallel Port, Serial Port 1, Serial Port 2, Floppy Disk controller.<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. |
| 4 | Reserved |
| 3 | **IDES1_ACT_STS** — R/WC. IDE Secondary 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. Clear this bit by writing a 1 to the bit location. |
| 2 | **IDES0_ACT_STS** — R/WC. IDE Secondary Drive 0.<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. |
| 1 | **IDEP1_ACT_STS** — R/WC. IDE Primary 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. Clear this bit by writing a 1 to the bit location. |
| 0 | **IDEP0_ACT_STS** — R/WC. IDE Primary Drive 0.<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. |

## 9.10.14 DEVTRAP_EN—Device Trap Enable Register

| | | | |
|---|---|---|---|
| I/O Address: | PMBASE +48h | Attribute: | R/W |
| Default Value | 0000h | Size: | 16-bit |
| Lockable: | No | Usage: | Legacy Only |
| Power Well: | Core | | |

This register enables the individual trap ranges to generate an SMI# when the corresponding status bit in the DEVACT_STS register is set. When a range is enabled, I/O cycles associated with that range will not be forwarded to LPC or IDE.

| Bit | Description |
|---|---|
| 15:13 | Reserved |
| 12 | **KBC_TRP_EN** — R/W. KBC (60/64h).<br>0 = Disable<br>1 = Enable |
| 11:10 | Reserved |
| 9:6 | Reserved |
| 5 | **LEG_IO_TRP_EN** — R/W. Parallel Port, Serial Port 1, Serial Port 2, Floppy Disk controller.<br>0 = Disable<br>1 = Enable |
| 4 | Reserved |
| 3 | **IDES1_TRP_EN** — R/W. IDE Secondary Drive 1.<br>0 = Disable<br>1 = Enable |
| 2 | **IDES0_TRP_EN** — R/W. IDE Secondary Drive 0.<br>0 = Disable<br>1 = Enable |
| 1 | **IDEP1_TRP_EN** — R/W. IDE Primary Drive 1.<br>0 = Disable<br>1 = Enable |
| 0 | **IDEP0_TRP_EN** — R/W. IDE Primary Drive 0.<br>0 = Disable<br>1 = Enable |

# 9.11 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.

The TCO I/O registers reside in a 32-byte range pointed to by a TCOBASE value, which is, ACPIBASE + 60h in the PCI config space. The following table shows the mapping of the registers within that 32-byte range. Each register is described in the following sections.

**Table 152. TCO I/O Register Address Map**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00h | TCO_RLD | TCO Timer Reload and Current Value | 00h | R/W |
| 01h | TCO_TMR | TCO Timer Initial Value | 04h | 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 |
| 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_WDSTS | Watchdog Status Register | 00h | R/W |
| 0Fh | — | Reserved | — | — |
| 10h | SW_IRQ_GEN | Software IRQ Generation Register | 11h | R/W |
| 11h–1Fh | — | Reserved | — | — |

## 9.11.1 TCO_RLD—TCO Timer Reload and Current Value Register

I/O Address: TCOBASE +00h
Default Value: 00h
Lockable: No
Attribute: R/W
Size: 8-bit
Power Well: Core

| Bit | Description |
|---|---|
| 7:0 | **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. Bits 7:6 will always be 0. |

## 9.11.2 TCO_TMR—TCO Timer Initial Value Register

| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +01h | Attribute: | R/W |
| Default Value: | 04h | Size: | 8-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:6 | Reserved |
| 5:0 | **TCO Timer Initial Value** — R/W. This field provides the value that is loaded into the timer each time the TCO_RLD register is written. Values of 0h–3h are ignored and should not be attempted. The timer is clocked at approximately 0.6 seconds, and this allows timeouts ranging from 2.4 seconds to 38 seconds. |

## 9.11.3 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. |

## 9.11.4 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. |

## 9.11.5 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 (Except bit 7, in RTC) |

| Bit | Description |
|---|---|
| 15:13 | Reserved |
| 12 | **HUBSERR_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Intel® ICH5 received an SERR# message via the hub interface. The software must read the memory controller hub (or its equivalent) to determine the reason for the SERR#.<br>**NOTE:** If this bit is set AND the SERR_EN bit in CMD register (D30:F0, Offset 04h, bit 8) is also set, the ICH5 will set the SSE bit in SECSTS register (D30:F0, offset 1Eh, bit 14) AND will also generate a NMI (or SMI# if NMI routed to SMI#). |
| 11 | **HUBNMI_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = ICH5 received an NMI message via the hub interface. The software must read the memory controller hub (or its equivalent) to determine the reason for the NMI. |
| 10 | **HUBSMI_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = ICH5 received an SMI message via the hub interface. The software must read the memory controller hub (or its equivalent) to determine the reason for the SMI#. |
| 9 | **HUBSCI_STS** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = ICH5 received an SCI message via the hub interface. The software must read the memory controller hub (or its equivalent) 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 = ICH5 sets this bit and generates and SMI# to indicate an illegal 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>**NOTE:** On write cycles attempted to the 4-MB lower alias to the BIOS space, the BIOSWR_STS will not be set. |
| 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>Note that 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, 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 ICH5 to indicate that the SMI was caused by the TCO timer reaching 0. |

| Bit | Description |
|---|---|
| 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. |
| 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. |
| 0 | **NMI2SMI_STS** — RO.<br>0 = Cleared by clearing the associated NMI status bit.<br>1 = Set by the ICH5 when an SMI# occurs because an event occurred that would otherwise have caused an NMI (because NMI2SMI_EN is set). |

## 9.11.6 TCO2_STS—TCO2 Status Register

| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +06h | Attribute: | R/WC |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | Power Well: | Resume<br>(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 = Intel® ICH5 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 ICH5 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 ICH5 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 illegal 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 = The ICH5 sets this bit to a 1 to indicate that the TCO timer timed out a second time (probably due to system lock). If this bit is set and the NO_REBOOT configuration bit is 0, then the ICH5 will reboot the system after the second timeout. The reboot is done by asserting PCIRST#. |
| 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 ICH5 to indicate that an intrusion was detected. This bit is set even if the system is in G3 state. |

## 9.11.7 TCO1_CNT—TCO1 Control Register

| | | | |
|---|---|---|---|
| I/O Address: | TCOBASE +08h | Attribute: | R/W, R/W (special), R/WC |
| Default Value: | 0000h | Size: | 16-bit |
| Lockable: | No | 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 Intel® ICH5 will clear this bit when it has completed sending the message. Software must not set this bit to 1 again until the ICH5 has set it back to 0.<br>1 = Writing a 1 to this bit will cause the ICH5 to send an Alert On LAN Event message over the SMLINK interface, with the Software Event bit set.<br>Setting the SEND_NOW bit causes the ICH5 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:<br>**NMI_EN / GBL_SMI_EN / Description**<br>0 / 0 / No SMI# at all because GBL_SMI_EN = 0<br>0 / 1 / SMI# will be caused due to NMI events<br>1 / 0 / No SMI# at all because GBL_SMI_EN = 0<br>1 / 1 / No SMI# due to NMI because NMI_EN = 1 |
| 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 |

## 9.11.8 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 |
|---|---|
| 15:4 | Reserved |
| 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 |

## 9.11.9 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. |

## 9.11.10 TCO_WDSTS—TCO Watchdog Status 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). |

### 9.11.11 SW_IRQ_GEN—Software IRQ Generation Register

Offset Address: TCOBASE + 10h
Default Value: 11h
Power Well: Core
Attribute: R/W
Size: 8 bits

| 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 Intel® ICH5's SERIRQ logic. This bit must be a 1 (default) if the ICH5 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 ICH5's SERIRQ logic. This bit must be a 1 (default) if the ICH5 is expected to receive IRQ1 assertions from a SERIRQ device. |

## 9.12 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 GPIO_BAR register.

**Table 153. Registers to Control GPIO Address Map**

| Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| **General Registers** | | | | |
| 00–03h | GPIO_USE_SEL | GPIO Use Select | 1A003180h | R/W |
| 04–07h | GP_IO_SEL | GPIO Input/Output Select | 0000 FFFFh | R/W |
| 08–0Bh | — | Reserved | — | — |
| 0C–0Fh | GP_LVL | GPIO Level for Input or Output | 1F1F 0000h | R/W |
| 10–13h | | Reserved | 00h | RO |
| Output Control Registers | | | | |
| 14–17h | GPO_TTL | GPIO TTL Select | 06630000h | RO |
| 18–1Bh | GPO_BLINK | GPIO Blink Enable | 00000000h | R/W |
| 1C–1Fh | — | Reserved | — | — |
| **Input Control Registers** | | | | |
| 20–2Bh | — | Reserved | — | — |
| 2C–2Fh | GPI_INV | GPIO Signal Invert | 00000000h | R/W |
| 30–33h | GPIO_USE_SEL2 | GPIO Use Select 2 | 00000007h | R/W |
| 34–37h | GP_IO_SEL2 | GPIO Input/Output Select 2 | 00000300h | R/W |
| 38–3Bh | GP_LVL2 | GPIO Level for Input or Output 2 | 00030207h | R/W |

## 9.12.1 GPIO_USE_SEL—GPIO Use Select Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE + 00h | Attribute: | R/W |
| Default Value: | 1A003180h | Size: | 32-bit |
| Lockable: | No | Power Well: | Core for 0:7 and 16:23<br>Resume for 8:15 and 24:31 |

| Bit | Description |
|---|---|
| 23:21, 14:15, 11:9, 5:0 | **GPIO_USE_SEL[23:21, 15:14, 11:9, 5:0]** — 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>**NOTE:** Bits 31:29, 26 are not implemented because there is no corresponding GPIO.<br>**NOTE:** Bits 28:27, 25, 13:12, and 8:7 are not implemented because the corresponding GPIOs are not multiplexed.<br>**NOTE:** Bits 16:17 are not implemented because the GPIO selection is controlled by bits 0:1. The REQ/GNT# pairs are enabled/disabled together. For example, if bit 0 is set to 1 then the REQ/GNTA# pair will function as GPIO0 and GPIO16.<br>After a full reset (RSMRST#) all multiplexed signals in the resume and core wells are configured as their native function rather than as a GPIO. After just a PCIRST#, the GPIO in the core well are configured as their native function. |

## 9.12.2 GP_IO_SEL—GPIO Input/Output Select Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +04h | Attribute: | R/W |
| Default Value: | 0000FFFFh | Size: | 32-bit |
| Lockable: | No | Power Well: | Resume |

| Bit | Description |
|---|---|
| 31:29, 26 | Reserved |
| 28:27<br>25:24 | **GPIO[n]_SEL** — R/W.<br>0 = Output. The corresponding GPIO signal is an output.<br>1 = Input. The corresponding GPIO signal is an input. |
| 23:16 | Always 0. The GPIOs are fixed as outputs. |
| 15:0 | Always 1. These GPIOs are fixed as inputs. |

## 9.12.3 GP_LVL—GPIO Level for Input or Output Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +0Ch | Attribute: | R/W |
| Default Value: | 1B3F 0000h | Size: | 32-bit |
| Lockable: | No | Power Well: | See bit descriptions |

| Bit | Description |
|---|---|
| 31:29, 26 | Reserved |
| 28:27, 25:24 | **GP_LVL[n]** — R/W. If GPIOn is programmed to be an output (via the corresponding bit in the GP_IO_SEL register) then the bit can be updated by software to drive a high or low value on the output pin. If GPIOn is programmed as an input, then software can read the bit to determine the level on the corresponding input pin. These bits correspond to GPIO that are in the Resume well, and will be reset to their default values by RSMRST# and also by a write to the CF9h register.<br>0 = Low<br>1 = High |
| 23:16 | **GP_LVL[n]** — R/W. These bits can be updated by software to drive a high or low value on the output pin. These bits correspond to GPIO that are in the core well, and will be reset to their default values by PCIRST#.<br>0 = Low<br>1 = High |
| 15:0 | Reserved. For GPI[13:11] and [8:0], the active status of a GPI is read from the corresponding bit in GPE0_STS register. |

## 9.12.4 GPO_BLINK—GPO Blink Enable Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +18h | Attribute: | R/W |
| Default Value: | 0004 0000h | Size: | 32-bit |
| Lockable: | No | Power Well: | See bit description |

| Bit | Description |
|---|---|
| 31:29, 26, 24:20, 17:0 | Reserved |
| 28:27, 25 | **GP_BLINK[n]** — R/W. The setting of these bits will have no effect if the corresponding GPIO is programmed as an input. These bits correspond to GPIO 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 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. |
| 19:18 | **GP_BLINK[n]** — R/W. The setting of these bits will have no effect if the corresponding GPIO is programmed as an input. These bits correspond to GPIO that are in the Core well, and will be reset to their default values by PCIRST#.<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 are approximately 0.5 seconds each. The GP_LVL bit is not altered when this bit is set. |

**NOTE:** GPIO18 will blink by default immediately after reset. This signal could be connected to an LED to indicate a failed boot (by programming BIOS to clear GP_BLINK18 after successful POST).

## 9.12.5 GPI_INV—GPIO Signal Invert Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +2Ch | Attribute: | R/W |
| Default Value: | 00000000h | Size: | 32-bit |
| Lockable: | No | Power Well: | See bit description |

| Bit | Description |
|---|---|
| 31:16 | Reserved |
| 15:8 | **GP_INV[n]** — R/W. 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 2 PCI clocks to ensure detection by the Intel® ICH5. 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 GPIO 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 ICH5 detects the state of the input pin to be high.<br>1 = The corresponding GPI_STS bit is set when the ICH5 detects the state of the input pin to be low. |
| 7:0 | **GP_INV[n]** — R/W. 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 2 PCI clocks to ensure detection by the ICH5. The setting of these bits will have no effect if the corresponding GPIO is programmed as an output. These bits correspond to GPIO that are in the Core well, and will be reset to their default values by PCIRST#.<br>0 = The corresponding GPI_STS bit is set when the ICH5 detects the state of the input pin to be high.<br>1 = The corresponding GPI_STS bit is set when the ICH5 detects the state of the input pin to be low. |

## 9.12.6 GPIO_USE_SEL2—GPIO Use Select 2 Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +30h | Attribute: | R/W |
| Default Value: | 00000007h | Size: | 32-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 31:0 | **GPIO_USE_SEL2[49:48, 41:40]—** 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 not implemented because there is no corresponding GPIO: 31:18, 16:10, 7:3.<br>2. The following bits are always 1 because they are unmuxed: 2:0.<br>3. If GPIOn does not exist, then the bit in this register will always read as 0 and writes will have no effect.<br>After a full reset (RSMRST#) all multiplexed signals in the resume and core wells are configured as their native function rather than as a GPIO. After just a PCIRST#, the GPIO in the core well are configured as their native function. |

## 9.12.7 GP_IO_SEL2—GPIO Input/Output Select 2 Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +34h | Attribute: | R/W |
| Default Value: | 00000300h | Size: | 32-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 31:18 | Always 0. No corresponding GPIO. |
| 17:16 | Always 0. Outputs. |
| 15:10 | Always 0. No corresponding GPIO. |
| 9:8 | Always 0. Inputs. |
| 7:3 | Always 0. No corresponding GPIO. |
| 2 | **GP_IO_SEL2**[34] — 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. |
| 1 | No Corresponding GPIO |
| 0 | **GP_IO_SEL2**[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. |

## 9.12.8 GP_LVL2—GPIO Level for Input or Output 2 Register

| | | | |
|---|---|---|---|
| Offset Address: | GPIOBASE +38h | Attribute: | R/W |
| Default Value: | 00030207h | Size: | 32-bit |
| Lockable: | No | Power Well: | See below |

| Bit | Description |
|---|---|
| 31:18 | Reserved. Read-only 0 |
| 17:16 | **GP_LVL[49:48]** — R/W. The corresponding GP_LVL[n] bit can be updated by software to drive a high or low value on the output pin. Since these bits correspond to GPIO that are in the processor I/O and core well, respectively, these bits will be reset by PCIRST#.<br>0 = Low<br>1 = High |
| 15:10 | Reserved. Read-only 0 |
| 9:8 | **GP_LVL[41:40]** — R/W. The corresponding GP_LVL[n] bit reflects the state of the input signal. Writes will have no effect. Since these bits correspond to GPIO that are in the core well, these bits will be reset by PCIRST#.<br>0 = Low<br>1 = High |
| 7:3 | Reserved. Read-only 0 |
| 2 | **GP_LVL**[34] — R/W. If GPIOn 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. If GPIOn is programmed as an input, then the corresponding GP_LVL bit reflects the state of the input signal (1 = high, 0 = low). Writes will have no effect.<br>0 = Low<br>1 = High<br>Since these bits correspond to GPIO that are in the core well, these bits will be reset by PCIRST#. |
| 1 | Reserved |
| 0 | **GP_LVL**[32] — R/W. If GPIOn 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. If GPIOn is programmed as an input, then the corresponding GP_LVL bit reflects the state of the input signal (1 = high, 0 = low). Writes will have no effect.<br>0 = Low<br>1 = High<br>Since these bits correspond to GPIO that are in the core well, these bits will be reset by PCIRST#. |

# IDE Controller Registers (D31:F1) 10

## 10.1 PCI Configuration Registers (IDE—D31:F1)

*Note:* Address locations that are not shown in Table 154 should be treated as Reserved (See Section 6.2 for details).

All of the IDE registers are in the core well. None of the registers can be locked.

**Table 154. IDE Controller PCI Register Address Map (IDE-D31:F1)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086h | RO |
| 02–03h | DID | Device Identification | 24DBh | RO |
| 04–05h | PCICMD | PCI Command | 00h | R/W, RO |
| 06–07h | PCISTS | PCI Status | 0280h | R/W, RO |
| 08h | RID | Revision Identification | See register description. | RO |
| 09h | PI | Programming Interface | 8Ah | R/W, RO |
| 0Ah | SCC | Sub Class Code | 01h | RO |
| 0Bh | BCC | Base Class Code | 01h | RO |
| 0Dh | PMLT | Primary Master Latency Timer | 00h | RO |
| 10–13h | PCMD_BAR | Primary Command Block Base Address | 00000001h | R/W, RO |
| 14–17h | PCNL_BAR | Primary Control Block Base Address | 00000001h | R/W, RO |
| 18–1Bh | SCMD_BAR | Secondary Command Block Base Address | 00000001h | R/W, RO |
| 1C–1Fh | SCNL_BAR | Secondary Control Block Base Address | 00000001h | R/W, RO |
| 20–23h | BM_BASE | Bus Master Base Address | 00000001h | R/W, RO |
| 2C–2Dh | IDE_SVID | Subsystem Vendor ID | 00h | R/WO |
| 2E–2Fh | IDE_SID | Subsystem ID | 00h | R/WO |
| 3C | INTR_LN | Interrupt Line | 00h | R/W |
| 3D | INTR_PN | Interrupt Pin | 01h | RO |
| 40–41h | IDE_TIMP | Primary IDE Timing | 0000h | R/W |
| 42–43h | IDE_TIMS | Secondary IDE Timing | 0000h | R/W |
| 44h | SLV_IDETIM | Slave IDE Timing | 00h | R/W |
| 48h | SDMA_CNT | Synchronous DMA Control | 00h | R/W |
| 4A–4Bh | SDMA_TIM | Synchronous DMA Timing | 0000h | R/W |
| 54h | IDE_CONFIG | IDE I/O Configuration | 00h | R/W |

**NOTES:**
1. Refer to the latest *Intel® ICH5 / ICH5R Specification Update* for the value of the Revision Identification register.
2. The ICH5 IDE controller is not arbitrated as a PCI device; therefore, it does not need a master latency timer.

## 10.1.1 VID—Vendor Identification Register (IDE—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 00–01h | 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 |

## 10.1.2 DID—Device Identification Register (IDE—D31:F0)

| | | | |
|---|---|---|---|
| Offset Address: | 02–03h | Attribute: | RO |
| Default Value: | 24DBh | Size: | 16-bit |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | Device ID — RO. This is a 16-bit value assigned to the Intel® ICH5 IDE controller. |

## 10.1.3 PCICMD—PCI Command Register (IDE—D31:F1)

Address Offset: 04h–05h
Default Value: 00h
Attribute: RO, R/W
Size: 16 bits

| Bit | Description |
|---|---|
| 15:11 | Reserved |
| 10 | **Interrupt Disable (ID)** — R/W.<br>0 = Enables the IDE controller to assert INTA# (native mode) or IRQ14/15 (legacy mode).<br>1 = Disable. The interrupt will be deasserted. |
| 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) — RO. Reserved as 0. |
| 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. |
| 2 | **Bus Master Enable (BME)** — R/W. Controls the Intel® ICH5's ability to act as a PCI master for IDE Bus Master transfers. |
| 1 | **Memory Space Enable (MSE)** — R/W.<br>0 = Disables access.<br>1 = Enables access to the IDE Expansion memory range. The EXBAR register (Offset 24h) must be programmed before this bit is set.<br><br>**NOTE:** BIOS should set this bit to a 1. |
| 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 IO registers.<br>1 = Enable. Note that the Base Address register for the Bus Master registers should be programmed before this bit is set.<br>**NOTES:**<br>1. Separate bits are provided (IDE Decode Enable, in the IDE Timing register) to independently disable the Primary or Secondary I/O spaces.<br>2. When this bit is 0 and the IDE controller is in Native Mode, the Interrupt Pin Register (see Section 10.1.18) will be masked (the interrupt will not be asserted). If an interrupt occurs while the masking is in place and the interrupt is still active when the masking ends, the interrupt will be allowed to be asserted. |

## 10.1.4 PCISTS — PCI Status Register (IDE—D31:F1)

Address Offset: 06–07h
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) — RO. Reserved as 0. |
| 14 | Signaled System Error (SSE) — RO. Reserved as 0. |
| 13 | **Received Master Abort (RMA)** — R/WC.<br>0 = Master abort **not** generated by Bus Master IDE interface function.<br>1 = Bus Master IDE interface function, as a master, generated a master abort. |
| 12 | Reserved as 0 — RO. |
| 11 | **Signaled Target Abort (STA)** — R/WC.<br>0 = Intel® ICH5 did **not** target abort a transaction targeting the IDE interface function.<br>1 = IDE interface function is targeted with a transaction that the ICH5 terminates with a target abort. |
| 10:9 | **DEVSEL# Timing Status (DEV_STS)** — RO.<br>01 = Hardwired; however, the ICH5 does not have a real DEVSEL# signal associated with the IDE unit, so these bits have no effect. |
| 8 | Data Parity Error Detected (DPED) — RO. Reserved as 0. |
| 7 | Fast Back to Back Capable (FB2BC) — RO. Reserved as 1. |
| 6 | User Definable Features (UDF) — RO. Reserved as 0. |
| 5 | 66 MHz Capable (66MHZ_CAP) — RO. Reserved as 0. |
| 4 | Reserved |
| 3 | **Interrupt Status (INTS)** — RO. This bit is independent of the state of the Interrupt Disable bit in the command register.<br>0 = Interrupt is cleared.<br>1 = Interrupt/MSI is asserted. |
| 2:0 | Reserved |

## 10.1.5 RID—Revision Identification Register (IDE—D31:F1)

Offset Address: 08h
Default Value: See bit description
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Revision ID — RO. This 8-bit value indicates the revision number for IDE controller.<br><br>**NOTE:** Refer to the latest Intel® ICH5 / ICH5R Specification Update for the value of the Revision Identification register. |

## 10.1.6 PI—Programming Interface Register (IDE—D31:F1)

Address Offset: 09h
Default Value: 8Ah
Attribute: R/W, RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7 | This read-only bit is a 1 to indicate that the Intel® ICH5 supports bus master operation. |
| 6:4 | Reserved. Hardwired to 000b. |
| 3 | SOP_MODE_CAP — RO. This read-only bit is a 1 to indicate that the secondary controller supports both legacy and native modes. |
| 2 | **SOP_MODE_SEL** — R/W. This read/write bit determines the mode that the secondary IDE channel is operating in.<br>0 = Legacy-PCI mode (default)<br>1 = Native-PCI mode |
| 1 | POP_MODE_CAP — RO. This read-only bit is a 1 to indicate that the primary controller supports both legacy and native modes. |
| 0 | **POP_MODE_SEL** — R/W. This read/write bits determines the mode that the primary IDE channel is operating in.<br>0 = Legacy-PCI mode (default)<br>1 = Native-PCI mode |

## 10.1.7 SCC—Sub Class Code Register (IDE—D31:F1)

Address Offset: 0Ah
Default Value: 01h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Sub Class Code (SCC) — RO.<br>01h = IDE device, in the context of a mass storage device. |

## 10.1.8 BCC—Base Class Code Register (IDE—D31:F1)

Address Offset: 0Bh
Default Value: 01h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Base Class Code (BCC) — RO.<br>01 = Mass storage device |

## 10.1.9 PMLT—Primary Master Latency Timer Register (IDE—D31:F1)

Address Offset: 0Dh
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Master Latency Timer Count (MLTC) — RO.<br>00h =Hardwired. The IDE controller is implemented internally, and is not arbitrated as a PCI device, so it does not need a Master Latency Timer. |

## 10.1.10 PCMD_BAR—Primary Command Block Base Address Register (IDE—D31:F1)

Address Offset: 10h–13h
Default Value: 00000001h
Attribute: R/W, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:16 | Reserved |
| 15:3 | **Base Address** — R/W. Base address of the I/O space (8 consecutive I/O locations). |
| 2:1 | Reserved |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1 indicating 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.

## 10.1.11 PCNL_BAR—Primary Control Block Base Address Register (IDE—D31:F1)

Address Offset: 14h–17h
Default Value: 00000001h
Attribute: R/W, RO
Size: 32 bits

| Bit | Description |
|---|---|
| 31:16 | Reserved |
| 15:2 | **Base Address** — R/W. Base address of the I/O space (4 consecutive I/O locations). |
| 1 | Reserved |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1 indicating 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.

## 10.1.12 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. Base address of the I/O space (eight, consecutive I/O locations). |
| 2:1 | Reserved |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1 indicating 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.

## 10.1.13 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. Base address of the I/O space (four, consecutive I/O locations). |
| 1 | Reserved |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1 indicating 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.

## 10.1.14 BM_BASE — Bus Master Base Address Register (IDE—D31:F1)

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 I/O 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 |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1 indicating a request for I/O space. |

## 10.1.15 IDE_SVID — Subsystem Vendor Identification (IDE—D31:F1)

Address Offset: 2Ch–2Dh
Default Value: 00h
Lockable: No
Attribute: R/WO
Size: 16 bits
Power Well: Core

| Bit | Description |
|---|---|
| 15:0 | **Subsystem Vendor ID (SVID)** — R/WO. The SVID register, in combination with the Subsystem ID (SID) register, enables the operating system (OS) to distinguish subsystems from each other. Software (BIOS) sets the value in this register. After that, the value can be read, but subsequent writes to this register have no effect. The value written to this register will also be readable via the corresponding SVID registers for the USB#1, USB#2, and SMBus functions. |

## 10.1.16 IDE_SID — Subsystem Identification Register (IDE—D31:F1)

Address Offset: 2Eh–2Fh
Default Value: 00h
Lockable: No
Attribute: R/WO
Size: 16 bits
Power Well: Core

| Bit | Description |
|---|---|
| 15:0 | **Subsystem ID (SID)** — R/WO. The SID register, in combination with the SVID register, enables the operating system (OS) to distinguish subsystems from each other. Software (BIOS) sets the value in this register. After that, the value can be read, but subsequent writes to this register have no effect. The value written to this register will also be readable via the corresponding SID registers for the USB#1, USB#2, and SMBus functions. |

## 10.1.17 INTR_LN—Interrupt Line Register (IDE—D31:F1)

Address Offset: 3Ch
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Line (INT_LN)** — R/W. This field is used to communicate to software the interrupt line that the interrupt pin is connected to. |

## 10.1.18 INTR_PN—Interrupt Pin Register (IDE—D31:F1)

Address Offset: 3Dh
Default Value: 01h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | Reserved |
| 2:0 | Interrupt Pin (INT_PIN) — RO. Hardwired to 01h to indicate to “software” that the Intel® ICH5 will drive INTA#. Note that this is only used in native mode. Also note that the routing to the internal interrupt controller doesn’t necessarily relate to the value in this register. The IDE interrupt is in fact routed to PIRQC# (IRQ18 in APIC mode). |

## 10.1.19 IDE_TIM — IDE Timing Register (IDE—D31:F1)

Address Offset: Primary: 40–41h
Secondary: 42–43h
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.

| Bit | Description |
|---|---|
| 15 | **IDE Decode Enable (IDE)** — R/W. This bit enables/disables the Primary or Secondary decode. The IDE I/O Space Enable bit in the Command register must be set in order for this bit to have any effect. Additionally, separate configuration bits are provided (in the IDE I/O Configuration register) to individually disable the primary or secondary IDE interface signals, even if the IDE Decode Enable bit is set.<br>0 = Disable<br>1 = Enables the Intel® ICH5 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. It also effects the corresponding primary or secondary memory decode range for IDE Expansion. |
| 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 determine 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. |

| Bit | Description |
|---|---|
| 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. |
| 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 |

## 10.1.20 SLV_IDETIM—Slave (Drive 1) IDE Timing Register (IDE—D31:F1)

Address Offset: 44h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| 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 |

| Bit | Description |
|---|---|
| 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 |
| 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 |

## 10.1.21 SDMA_CNT—Synchronous DMA Control Register (IDE—D31:F1)

Address Offset: 48h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| 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. |

## 10.1.22 SDMA_TIM—Synchronous DMA Timing Register (IDE—D31:F1)

Address Offset: 4A–4Bh
Default Value: 0000h
Attribute: R/W
Size: 16 bits

***Note:*** For FAST_SCB1 = 1 (133 MHz clk) in bits [13:12, 9:8, 5:4, 1:0], refer to Section 5.16.6 for details.

| Bit | Description |
|---|---|
| 15:14 | Reserved |
| 13:12 | **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.<br>**SCB1 = 0 (33 MHz clk)** / **SCB1 = 1 (66 MHz clk)** / **FAST_SCB1 = 1 (133 MHz clk)**<br>00 = CT 4 clocks, RP 6 clocks / 00 = Reserved / 00 = Reserved<br>01 = CT 3 clocks, RP 5 clocks / 01 = CT 3 clocks, RP 8 clocks / 01 = CT 3 clks, RP 16 clks<br>10 = CT 2 clocks, RP 4 clocks / 10 = CT 2 clocks, RP 8 clocks / 10 = Reserved<br>11 = Reserved / 11 = Reserved / 11 = Reserved |
| 11:10 | Reserved |
| 9:8 | **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.<br>**SCB1 = 0 (33 MHz clk)** / **SCB1 = 1 (66 MHz clk)** / **FAST_SCB1 = 1 (133 MHz clk)**<br>00 = CT 4 clocks, RP 6 clocks / 00 = Reserved / 00 = Reserved<br>01 = CT 3 clocks, RP 5 clocks / 01 = CT 3 clocks, RP 8 clocks / 01 = CT 3 clks, RP 16 clks<br>10 = CT 2 clocks, RP 4 clocks / 10 = CT 2 clocks, RP 8 clocks / 10 = Reserved<br>11 = Reserved / 11 = Reserved / 11 = Reserved |
| 7:6 | Reserved |
| 5:4 | **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.<br>**PCB1 = 0 (33 MHz clk)** / **PCB1 = 1 (66 MHz clk)** / **FAST_PCB1 = 1 (133 MHz clk)**<br>00 = CT 4 clocks, RP 6 clocks / 00 = Reserved / 00 = Reserved<br>01 = CT 3 clocks, RP 5 clocks / 01 = CT 3 clocks, RP 8 clocks / 01 = CT 3 clks, RP 16 clks<br>10 = CT 2 clocks, RP 4 clocks / 10 = CT 2 clocks, RP 8 clocks / 10 = Reserved<br>11 = Reserved / 11 = Reserved / 11 = Reserved |
| 3:2 | Reserved |
| 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.<br>**PCB1 = 0 (33 MHz clk)** / **PCB1 = 1 (66 MHz clk)** / **FAST_PCB1 = 1 (133 MHz clk)**<br>00 = CT 4 clocks, RP 6 clocks / 00 = Reserved / 00 = Reserved<br>01 = CT 3 clocks, RP 5 clocks / 01 = CT 3 clocks, RP 8 clocks / 01 = CT 3 clks, RP 16 clks<br>10 = CT 2 clocks, RP 4 clocks / 10 = CT 2 clocks, RP 8 clocks / 10 = Reserved<br>11 = Reserved / 11 = Reserved / 11 = Reserved |

## 10.1.23 IDE_CONFIG—IDE I/O Configuration Register (IDE—D31:F1)

Address Offset: 54h
Default Value: 00h
Attribute: R/W
Size: 32 bits

| Bit | Description |
|---|---|
| 31:20 | Reserved |
| 19:18 | **SEC_SIG_MODE** — R/W. These bits are used to control mode of the Secondary IDE signal pins for swap bay support.<br>If the SRS bit (bit 15, offset D0h of D31:F0) 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 swap bay support.<br>If the PRS bit (bit 14, offset D0h of D31:F0) 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 |
| 15 | **Fast Secondary Drive 1 Base Clock (FAST_SCB1)** — R/W. This bit is used in conjunction with the SCT1 bits 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 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 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 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 | **Secondary Slave Channel Cable Reporting** — R/W. BIOS should program this bit to tell the IDE driver which cable is plugged into the channel.<br>0 = 40 conductor cable is present.<br>1 = 80 conductor cable is present. |
| 6 | **Secondary Master Channel Cable Reporting** — R/W. Same description as bit 7. |
| 5 | **Primary Slave Channel Cable Reporting** — R/W. Same description as bit 7. |
| 4 | **Primary Master Channel Cable Reporting** — R/W. Same description as bit 7. |
| 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. |

| Bit | Description |
|---|---|
| 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. |

## 10.2 Bus Master IDE I/O Registers (IDE—D31:F1)

The bus master IDE function uses 16 bytes of I/O space, allocated via the BMIBA register, located in Device 31:Function 1 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 affect (but should not be attempted). The description of the I/O registers is shown in Table 155.

### Table 155. Bus Master IDE I/O Registers

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00 | BMICP | Bus Master IDE Command Primary | 00h | R/W |
| 01 | — | Reserved | 00h | RO |
| 02 | BMISP | Bus Master IDE Status Primary | 00h | R/WC |
| 03 | — | Reserved | 00h | RO |
| 04–07 | BMIDP | Bus Master IDE Descriptor Table Pointer Primary | xxxxxxxxh | R/W |
| 08 | BMICS | Bus Master IDE Command Secondary | 00h | R/W |
| 09 | — | Reserved | 00h | RO |
| 0A | BMISS | Bus Master IDE Status Secondary | 00h | R/WC |
| 0B | — | Reserved | 00h | RO |
| 0C–0F | BMIDS | Bus Master IDE Descriptor Table Pointer Secondary | xxxxxxxxh | R/W |

## 10.2.1 BMIC[P,S]—Bus Master IDE Command Register (IDE—D31:F1)

| | | | |
|---|---|---|---|
| Address Offset: | Primary: 00h<br>Secondary: 08h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:4 | Reserved. Returns 0. |
| 3 | **Read / Write Control (RWC)** — 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 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: Offset 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. |

## 10.2.2 BMIS[P,S]—Bus Master IDE Status Register (IDE—D31:F1)

Address Offset: Primary: 02h, Secondary: 0Ah — Attribute: R/WC
Default Value: 00h — Size: 8 bits

| Bit | Description |
|---|---|
| 7 | Reserved. Returns 0. |
| 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 Intel® ICH5 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 ICH5 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. Software can use this bit to determine if an IDE device has asserted its interrupt line (IRQ 14 for the Primary channel, and IRQ 15 for Secondary).<br>0 = Software clears this bit by writing a 1 to it. If this bit is cleared while the interrupt is still active, this bit will remain clear until another assertion edge is detected on the interrupt line.<br>1 = Set by the rising edge of the IDE interrupt line, regardless of whether or not the interrupt is masked in the 8259 or the internal I/O APIC. When this bit is read as 1, all data transferred from the drive is visible in system memory. |
| 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 ICH5 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 ICH5 when the Start bit is cleared in the Command register. When this bit is read as 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 ICH5 when the Start bit is written to the Command register. |

## 10.2.3 BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register (IDE—D31:F1)

Address Offset: Primary: 04h, Secondary: 0Ch — Attribute: R/W
Default Value: All bits undefined — Size: 32 bits

| Bit | Description |
|---|---|
| 31:2 | **Address of Descriptor Table (ADDR)** — R/W. Corresponds 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 |

# SATA Controller Registers (D31:F2) 11

## 11.1 PCI Configuration Registers (SATA–D31:F2)

*Note:* Address locations that are not shown in Table 156 should be treated as Reserved (see Section 6.2 for details).

All of the SATA registers are in the core well. None of the registers can be locked.

**Table 156. SATA Controller PCI Register Address Map (SATA–D31:F2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086h | RO |
| 02–03h | DID | Device Identification | 24D1h (Intel® 82801EB ICH5)<br>24DFh (82801ER ICH5R) | RO |
| 04–05h | PCICMD | PCI Command | 00h | R/W, RO |
| 06–07h | PCISTS | PCI Status | 02B0h | R/WC, RO |
| 08h | RID | Revision Identification | See register description | RO |
| 09h | PI | Programming Interface | 8Ah | R/W, RO |
| 0Ah | SCC | Sub Class Code | 01h (82801EB ICH5)<br>04h (82801ER ICH5R) | RO |
| 0Bh | BCC | Base Class Code | 01h | RO |
| 0Dh | PMLT | Primary Master Latency Timer | 00h | RO |
| 10–13h | PCMD_BAR | Primary Command Block Base Address | 00000001h | R/W, RO |
| 14–17h | PCNL_BAR | Primary Control Block Base Address | 00000001h | R/W, RO |
| 18–1Bh | SCMD_BAR | Secondary Command Block Base Address | 00000001h | R/W, RO |
| 1C–1Fh | SCNL_BAR | Secondary Control Block Base Address | 00000001h | R/W, RO |
| 20–23h | BAR | Legacy Bus Master Base Address | 00000001h | R/W, RO |
| 2C–2Dh | SVID | Subsystem Vendor Identification | 0000h | R/WO |
| 2E–2Fh | SID | Subsystem Identification | 0000h | R/WO |
| 34h | CAP | Capabilities Pointer | 80h | RO |
| 3C | INT_LN | Interrupt Line | 00h | R/W |
| 3D | INT_PN | Interrupt Pin | 01h | RO |
| 40–41h | IDE_TIMP | Primary IDE Timing | 0000h | R/W |
| 42–43h | 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 |
| 4A–4Bh | SDMA_TIM | Synchronous DMA Timing | 0000h | R/W |
| 54–57h | IDE_CONFIG | IDE I/O Configuration | 00000000h | R/W |

**Table 156. SATA Controller PCI Register Address Map (SATA–D31:F2)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 70–71h | PID | PCI Power Management Capability ID | 0001h | RO |
| 72–73h | PC | PCI Power Management Capabilities | 0002h | RO |
| 74–75h | PMCS | PCI Power Management Control and Status | 0000h | R/W, RO |
| 80–81h | MID | Message Signaled Interrupt ID | 7005h | RO |
| 82–83h | MC | Message Signaled Interrupt Message Control | 0000h | R/W, RO |
| 84–87h | MA | Message Signaled Interrupt Message Address | 00000000h | R/W |
| 88–89h | MD | 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 |
| A0h | SRI | SATA Registers Index | 00h | R/W |
| A4h | SRD | SATA Registers Data | XXh | 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 |

**NOTES:**
1. Refer to the latest *Intel® ICH5 / ICH5R Specification Update* for the value of the Revision Identification register.
2. The ICH5 SATA controller is not arbitrated as a PCI device, therefore it does not need a master latency timer.

## 11.1.1 VID—Vendor Identification Register (SATA—D31:F2)

Offset Address: 00–01h
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 |

## 11.1.2 DID—Device Identification Register (SATA—D31:F2)

Offset Address: 02–03h
Default Value: 24D1h (82801EB ICH5 only)
24DFh (82801ER ICH5R only)
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 Intel® ICH5 SATA controller. |

## 11.1.3 PCICMD—PCI Command Register (SATA–D31:F2)

Address Offset: 04h–05h
Default Value: 0000h
Attribute: RO, R/W
Size: 16 bits

| Bit | Description |
|---|---|
| 15:11 | Reserved |
| 10 | **Interrupt Disable** — R/W.<br>0 = Enables the SATA host controller to assert INTA# (native mode), IRQ14/15 (legacy mode), and MSI (if MSI is enabled).<br>1 = The interrupt will be deasserted and it may not generate MSIs. |
| 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. |
| 2 | **Bus Master Enable (BME)** — R/W. This bit controls the Intel® ICH5'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)** — RO. The SATA controller does not contain memory space. |
| 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. |

## 11.1.4 PCISTS — PCI Status Register (SATA–D31:F2)

Address Offset: 06–07h
Default Value: 02A0h
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 = Bus Master IDE interface function, 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 Intel® ICH5, this bit can only be set on read completions received from SiBUS where there is a parity error.<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 | 66 MHz Capable (66MHZ_CAP) — RO. Reserved as 1. |
| 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.<br>0 = A capabilities list is not present<br>1 = A capabilities list is present<br><br>**NOTE:** This bit is hardwired to 0. |
| 3 | Interrupt Status (INTS)— RO. This bit is independent of the state of the Interrupt Disable bit in the command register.<br>0 = Interrupt is cleared.<br>1 = Interrupt/MSI is asserted. |
| 2:0 | Reserved |

## 11.1.5 RID—Revision Identification Register (SATA—D31:F2)

Offset Address: 08h
Default Value: See bit description
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Revision ID (RID) — RO. 8-bit value that indicates the revision number for SATA.<br><br>**NOTE:** Refer to the latest Intel® ICH5 / ICH5R Specification Update for the value of the Revision Identification register. |

## 11.1.6 PI—Programming Interface Register (SATA–D31:F2)

Address Offset: 09h
Default Value: 8Ah
Attribute: R/W, RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7 | This read-only bit is a 1 to indicate that the Intel® ICH5 supports bus master operation |
| 6:4 | Reserved. Will always return 0. |
| 3 | **SOP_MODE_CAP** — RO. Hardwired to 1 to indicate that the secondary controller supports both legacy and native modes. |
| 2 | **SOP_MODE_SEL** — R/W. This bit determines the operating mode of the secondary IDE channel.<br>0 = Legacy-PCI mode (default)<br>1 = Native-PCI mode |
| 1 | **POP_MODE_CAP** — RO. Hardwired to 1 indicate that the primary controller supports both legacy and native modes. |
| 0 | **POP_MODE_SEL** — R/W. This bit determines the operating mode of the primary IDE channel.<br>0 = Legacy-PCI mode (default)<br>1 = Native-PCI mode |

## 11.1.7 SCC—Sub Class Code Register (SATA–D31:F2)

Address Offset: 0Ah
Default Value: 01h (82801EB ICH5 only)
04h (82801ER ICH5R only)
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Sub Class Code (SCC)** — RO.<br>01h = IDE device, in the context of a mass storage device. (Intel® 82801EB ICH5 only)<br>04h = Intel® RAID Technology device, in the context of a mass storage device. (Intel® 82801ER ICH5R only) |

## 11.1.8 BCC—Base Class Code Register (SATA–D31:F2)

Address Offset: 0Bh
Default Value: 01h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BCC)** — RO.<br>01h = Mass storage device |

## 11.1.9 PMLT—Primary Master Latency Timer Register (SATA–D31:F2)

Address Offset: 0Dh
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Master Latency Timer Count (MLTC) — RO.<br>00h = Hardwired. The IDE controller is implemented internally, and is not arbitrated as a PCI device, so it does not need a Master Latency Timer. |

## 11.1.10 PCMD_BAR—Primary Command Block Base Address Register (SATA–D31:F2)

Address Offset: 10h–13h
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 (eight, 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.

## 11.1.11 PCNL_BAR—Primary Control Block Base Address Register (SATA–D31:F2)

Address Offset: 14h–17h
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 (four, 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.

## 11.1.12 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 (eight, 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.

## 11.1.13 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 (four, 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.

## 11.1.14 BAR — Legacy Bus Master Base Address Register (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | 20h–23h | Attribute: | R/W, RO |
| Default Value: | 00000001h | 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 |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1 to indicate a request for I/O space. |

## 11.1.15 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. The SVID register, in combination with the Subsystem ID (SID) register, enables the operating system (OS) to distinguish subsystems from each other. Software (BIOS) sets the value in this register. After that, the value can be read, but subsequent writes to this register have no effect. |

## 11.1.16 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. The SID register, in combination with the SVID register, enables the operating system (OS) to distinguish subsystems from each other. Software (BIOS) sets the value in this register. After that, the value can be read, but subsequent writes to this register have no effect. |

## 11.1.17 CAP—Capabilities Pointer Register (SATA–D31:F2)

Address Offset: 34h
Default Value: 00h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Capabilities Pointer (CAP_PTR) — RO. This bit indicates that the first capability pointer is set to 00h and therefore is not pointing to anything (i.e., disabled). |

## 11.1.18 INT_LN—Interrupt Line Register (SATA–D31:F2)

Address Offset: 3Ch
Default Value: 00h
Attribute: R/W
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. |

## 11.1.19 INT_PN—Interrupt Pin Register (SATA–D31:F2)

Address Offset: 3Dh
Default Value: 01h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:3 | Reserved |
| 2:0 | Interrupt Pin — RO. Hardwired to 01h indicating to “software” that the Intel® ICH5 will drive INTA#. Note that this is only used in native mode. Also note that the routing to the internal interrupt controller doesn’t necessarily relate to the value in this register. |

## 11.1.20 IDE_TIM — IDE Timing Register (SATA–D31:F2)

Address Offset: Primary: 40–41h
Secondary: 42–43h
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 Intel® ICH5 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 6.16 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. |

| Bit | Description |
|---|---|
| 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. |
| 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 |

## 11.1.21 SIDETIM—Slave IDE Timing Register (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | 44h | Attribute: | R/W |
| Default Value: | 00h | 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 |
| 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 |

## 11.1.22 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 |

## 11.1.23 SDMA_TIM—Synchronous DMA Timing Register (SATA–D31:F2)

Address Offset: 4A–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.

| Bit | Description |
|---|---|
| 15:14 | Reserved |
| 13:12 | **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.<br>**SCB1 = 0 (33 MHz clk)** / **SCB1 = 1 (66 MHz clk)** / **FAST_SCB1 = 1 (133 MHz clk)**<br>00 = CT 4 clocks, RP 6 clocks / 00 = Reserved / 00 = Reserved<br>01 = CT 3 clocks, RP 5 clocks / 01 = CT 3 clocks, RP 8 clocks / 01 = CT 3 clks, RP 16 clks<br>10 = CT 2 clocks, RP 4 clocks / 10 = CT 2 clocks, RP 8 clocks / 10 = Reserved<br>11 = Reserved / 11 = Reserved / 11 = Reserved |
| 11:10 | Reserved |
| 9:8 | **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.<br>**SCB1 = 0 (33 MHz clk)** / **SCB1 = 1 (66 MHz clk)** / **FAST_SCB1 = 1 (133 MHz clk)**<br>00 = CT 4 clocks, RP 6 clocks / 00 = Reserved / 00 = Reserved<br>01 = CT 3 clocks, RP 5 clocks / 01 = CT 3 clocks, RP 8 clocks / 01 = CT 3 clks, RP 16 clks<br>10 = CT 2 clocks, RP 4 clocks / 10 = CT 2 clocks, RP 8 clocks / 10 = Reserved<br>11 = Reserved / 11 = Reserved / 11 = Reserved |
| 7:6 | Reserved |
| 5:4 | **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.<br>**PCB1 = 0 (33 MHz clk)** / **PCB1 = 1 (66 MHz clk)** / **FAST_PCB1 = 1 (133 MHz clk)**<br>00 = CT 4 clocks, RP 6 clocks / 00 = Reserved / 00 = Reserved<br>01 = CT 3 clocks, RP 5 clocks / 01 = CT 3 clocks, RP 8 clocks / 01 = CT 3 clks, RP 16 clks<br>10 = CT 2 clocks, RP 4 clocks / 10 = CT 2 clocks, RP 8 clocks / 10 = Reserved<br>11 = Reserved / 11 = Reserved / 11 = Reserved |
| 3:2 | Reserved |
| 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.<br>**PCB1 = 0 (33 MHz clk)** / **PCB1 = 1 (66 MHz clk)** / **FAST_PCB1 = 1 (133 MHz clk)**<br>00 = CT 4 clocks, RP 6 clocks / 00 = Reserved / 00 = Reserved<br>01 = CT 3 clocks, RP 5 clocks / 01 = CT 3 clocks, RP 8 clocks / 01 = CT 3 clks, RP 16 clks<br>10 = CT 2 clocks, RP 4 clocks / 10 = CT 2 clocks, RP 8 clocks / 10 = Reserved<br>11 = Reserved / 11 = Reserved / 11 = Reserved |

## 11.1.24 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). Intel® ICH5 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 swap bay support.<br>If the SRS bit (bit 15, offset D0h of D31:F0) 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 (bit 14, offset D0h of D31:F0) 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 |
| 15 | **Fast Secondary Drive 1 Base Clock (FAST_SCB1)** — R/W. This bit is used in conjunction with the SCT1 bits 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 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 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 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). ICH5 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. |

| Bit | Description |
|---|---|
| 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. |

## 11.1.25 PID—PCI Power Management Capability Identification Register (SATA–D31:F2)

Address Offset: 70–71h Attribute: RO
Default Value: 0001h Size: 16 bits

| Bits | Description |
|---|---|
| 15:8 | Next Capability (NEXT) — RO. Indicates that this is the last item in the list. |
| 7:0 | Capability ID (CID) — RO. Indicates that this pointer is a PCI power management. |

## 11.1.26 PC—PCI Power Management Capabilities Register (SATA–D31:F2)

Address Offset: 72–73h Attribute: RO
Default Value: 0002h Size: 16 bits

| Bits | Description |
|---|---|
| 15:11 | PME Support (PME_SUP) — RO. Hardwired to 0s to indicate PME# cannot be generated form the SATA host controller. When in low power state, resume events are not allowed. |
| 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. Hardwired to 000 to indicate 375 mA maximum Suspend well current required when in the D3 cold state. |
| 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 the *PCI Power Management Specification, Revision 1.1*. |

## 11.1.27 PMCS—PCI Power Management Control and Status Register (SATA–D31:F2)

Address Offset: 74–75h
Default Value: 0000h
Attribute: RO, R/W
Size: 16 bits

| Bits | Description |
|---|---|
| 15 | PME Status (PMES) — RO. Reserved as 0. |
| 14:9 | Reserved |
| 8 | PME Enable (PMEE) — RO. Reserved as 0. |
| 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>01 = D1 state<br>10 = D2 state<br>11 = D3hot state<br>When in the D3hot state, the controller's configuration space is available, but the I/O and memory spaces are not. Additionally, interrupts are blocked. |

## 11.1.28 MID—Message Signaled Interrupt Identifiers Register (SATA–D31:F2)

Address Offset: 80–81h
Default Value: 7005h
Attribute: RO
Size: 16 bits

| Bits | Description |
|---|---|
| 15:8 | Next Pointer (NEXT) — RO. This field indicates that the next item in the list the PCI power management pointer. |
| 7:0 | Capability ID (CID) — RO. The Capabilities ID indicates MSI. |

## 11.1.29 MC—Message Signaled Interrupt Message Control Register (SATA–D31:F2)

Address Offset: 82–83h
Default Value: 0000h
Attribute: RO, R/W
Size: 16 bits

| Bits | Description |
|---|---|
| 15:8 | Reserved |
| 7 | 64 Bit Address Capable (C64) — RO. Hardwired to 0 to indicate capability of generating 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 Intel® ICH5. |
| 3:1 | Multiple Message Capable (MMC) — RO. Only one message is required. |
| 0 | **MSI Enable (MSIE)** — R/W.<br>0 = Disabled.<br>1 = MSI is enabled and traditional interrupt pins are not used to generate interrupts. |

## 11.1.30 MA—Message Signaled Interrupt Message Address Register (SATA–D31:F2)

Address Offset: 84–87h
Default Value: 00000000h
Attribute: R/W
Size: 32 bits

| Bits | Description |
|---|---|
| 31:2 | **Address (ADDR)** — R/W. Lower 32 bits of the system specified message address, always DWord aligned. |
| 1:0 | Reserved |

## 11.1.31 MD—Message Signaled Interrupt Message Data Register (SATA–D31:F2)

Address Offset: 88–89h
Default Value: 0000h
Attribute: R/W
Size: 16 bits

| Bits | Description |
|---|---|
| 15:0 | **Data (DATA)** — R/W. This 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. |

## 11.1.32 MAP—Address Map Register (SATA–D31:F2)

Address Offset: 90h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bits | Description |
|---|---|
| 7:3 | Reserved. |
| 2:0 | **Map Value** — R/W. The value of these bits indicate the address range the SATA port responds to, and whether or not the SATA and IDE functions are combined.<br>000 = Non-combined. P0 is primary master. P1 is secondary master.<br>001 = Non-combined. P0 is secondary master. P1 is primary master.<br>100 = Combined. P0 is primary master. P1 is primary slave. IDE is secondary; Primary IDE channel disabled.<br>101 = Combined. P0 is primary slave. P1 is primary master. IDE is secondary.<br>110 = Combined. IDE is primary. P0 is secondary master. P1 is secondary slave; Secondary IDE channel disabled.<br>111 = Combined. IDE is primary. P0 is secondary slave. P1 is secondary master. |

## 11.1.33 PCS—Port Control and Status Register (SATA–D31:F2)

Address Offset: 92h–93h
Default Value: 0000h
Attribute: R/W, RO
Size: 16 bits

| Bits | Description |
|---|---|
| 15:6 | Reserved. |
| 5 | **Port 1 Present (P1P)** — RO.<br>0 = Device not detected. This bit is cleared when the port is disabled via the P1E bit (bit 1 of this register).<br>1 = Device present. The SATA host has detected the presence of a device on port 1. It may change at any time.<br>**NOTE:** SATA device presence detection is dependant on the amount of time a device needs to prepare to be detected. Device preparation time is device-specific, and is not specified. |
| 4 | **Port 0 Present (P0P)** — RO.<br>0 = Device not detected.This bit is cleared when the port is disabled via the P0E bit (bit 0 of this register).<br>1 = Device present. The SATA host has detected the presence of a device on port 1. It may change at any time.<br>**NOTE:** SATA device presence detection is dependant on the amount of time a device needs to prepare to be detected. Device preparation time is device-specific, and is not specified. |
| 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. |
| 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 |

## 11.1.34 SRI—SATA Registers Index (SATA–D31:F2)

Address Offset: A0h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bits | Description |
|---|---|
| 7 | Reserved. |
| 6:0 | **Index (IDX)** — R/W. This field is a 7-bit index pointer into the SATA Registers space. Data is written into the SRD register (D31:F2:A4h) and read from the SRD register. |

**Table 157. SATA Indexed Registers**

| Index | Name |
|---|---|
| 00h–17h | Reserved |
| 18h–1Bh | SATA Initialization Register A (SIRA) |
| 1Ch–3Fh | Reserved |
| 40h–43h | SATA Initialization Register B (SIRB) |
| 44h–57h | Reserved |
| 58h–5Bh | Power Management Register Port 0 (PMR0) |
| 5Ch–67h | Reserved |
| 68h–6Bh | Power Management Register Port 1 (PMR1) |
| 6Ch–FFh | Reserved |

## 11.1.35 SRD—SATA Registers Data (SATA–D31:F2)

Address Offset: A4h–A7h
Default Value: XXh
Attribute: R/W
Size: 8 bits

| Bits | Description |
|---|---|
| 31:0 | **Data (DTA)** — R/W. This field is a 32-bit data value that is written to the register pointed to by SRI (D31:F2:A0h) or read from the register pointed to by SRI. |

## 11.1.36 SIRA—SATA Initialization Register A (SATA–D31:F2)

Index Address: Index 18h–1Bh
Default Value: 0000025Bh
Attribute: R/W
Size: 32 bits

| Bit | Description |
|---|---|
| 31:8 | Reserved |
| 7:0 | **SATA Setup Data A (SSDA)** — R/W. This field is written by BIOS during SATA initialization. Contact your Intel Field Representative for additional BIOS information. |

## 11.1.37 SIRB — SATA Initialization Register B (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Index Address: | Index 40h–43h | Attribute: | R/W |
| Default Value: | 0011017Dh | Size: | 32 bits |

| Bit | Description |
|---|---|
| 31:24 | Reserved |
| 23:16 | **SATA Setup Data B (SSDB)** — R/W. This field is written by BIOS during SATA initialization. Contact your Intel Field Representative for additional BIOS information. |
| 15:0 | Reserved |

## 11.1.38 PMR0 — Power Management Register Port 0 (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Index Offset: | Index 58h–5Bh | Attribute: | R/W |
| Default Value: | XXXXXXXXh | Size: | 32 bits |

| Bits | Description |
|---|---|
| 31:16 | Reserved |
| 15:8 | **Device Partial/Slumber Request Port 0** — R/W. The Intel® ICH5 Port 0 configuration to respond to device-initiated requests to transition to partial/slumber power management states.<br><br>**NOTE:** BIOS must program this field to 03h. |
| 7:0 | Reserved |

## 11.1.39 PMR1 — Power Management Register Port 1 (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Index Offset: | Index 68h–6Bh | Attribute: | R/W |
| Default Value: | XXXXXXXXh | Size: | 32 bits |

| Bits | Description |
|---|---|
| 31:16 | Reserved |
| 15:8 | **Device Partial/Slumber Request Port 1** — R/W. The Intel® ICH5 Port 1 configuration to respond to device-initiated requests to transition to partial/slumber power management states.<br><br>**NOTE:** BIOS must program this field to 03h. |
| 7:0 | Reserved |

## 11.1.40 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 = This bit is set any time a BIST FIS transmitted by Intel® ICH5 receives 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 = This bit is set any time a BIST FIS transmitted by ICH5 receives 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 ICH5 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 ICH5 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 ICH5 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 ICH5 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 ICH5. This field is not port specific — its contents will be used for any BIST FIS initiated on port 0 on port 1. 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 |

## 11.1.41 BFTD1—BIST FIS Transmit Data1 Register (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | E4h–E7h | Attribute: | R/W |
| Default Value: | 00000000h | 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 Intel® ICH5. This register is not port specific — its contents will be used for BIST FIS initiated on port 0 or port 1. 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. |

## 11.1.42 BFTD2—BIST FIS Transmit Data2 Register (SATA–D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | E8h–EBh | Attribute: | R/W |
| Default Value: | 00000000h | 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 Intel® ICH5. This register is not port specific — its contents will be used for BIST FIS initiated on port 0 or port 1. 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. |

## 11.2 Bus Master IDE I/O Registers (D31:F2)

The bus master IDE function uses 16 bytes of I/O space, allocated via the BMIBA register, located in Device 31:Function 1 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 affect (but should not be attempted). The description of the I/O registers is shown in Table 158.

**Table 158. Bus Master IDE I/O Register Address Map**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00 | BMICP | Command Register Primary | 01h | R/W |
| 01 | — | Reserved | — | RO |
| 02 | BMISP | Bus Master IDE Status Register Primary | 00h | R/W, R/WC, RO |
| 03 | — | Reserved | — | RO |
| 04–07 | BMIDP | Bus Master IDE Descriptor Table Pointer Primary | xx | R/W |
| 08 | BMICS | Command Register Secondary | 01h | R/W |
| 09 | — | Reserved | — | RO |
| 0A | BMISS | Bus Master IDE Status Register Secondary | 00h | R/W, R/WC, RO |
| 0B | — | Reserved | — | RO |
| 0C–0F | BMIDS | Bus Master IDE Descriptor Table Pointer Secondary | xx | R/W |

## 11.2.1 BMIC[P,S]—Bus Master IDE Command Register (D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | Primary: 00h<br>Secondary: 08h | Attribute: | R/W |
| Default Value: | 01h | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:4 | Reserved. Returns 0. |
| 3 | **Read / Write Control (RWC)** — 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 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: Offset 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. |

## 11.2.2 BMIS[P,S]—Bus Master IDE Status Register (D31:F2)

Address Offset: Primary: 02h
Secondary: 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 Intel® ICH5 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 ICH5 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. Software can use this bit to determine if an IDE device has asserted its interrupt line (IRQ 14 for the Primary channel, and IRQ 15 for Secondary).<br>0 = Software clears this bit by writing a 1 to it. If this bit is cleared while the interrupt is still active, this bit will remain clear until another assertion edge is detected on the interrupt line.<br>1 = Set by the rising edge of the IDE interrupt line, regardless of whether or not the interrupt is masked in the 8259 or the internal I/O APIC. When this bit is read as a 1, all data transferred from the drive is visible in system memory. |
| 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 ICH5 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 ICH5 when the Start bit 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 ICH5 when the Start bit is written to the Command register. |

## 11.2.3 BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register (D31:F2)

| | | | |
|---|---|---|---|
| Address Offset: | Primary: 04h–07h<br>Secondary: 0Ch–0Fh | Attribute: | R/W |
| 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 |

This page is intentionally left blank.

# UHCI Controllers Registers 12

## 12.1 PCI Configuration Registers (USB—D29:F0/F1/F2/F3)

*Note:* Register address locations that are not shown in Table 159 and should be treated as Reserved (see Section 6.2 for details).

**Table 159. 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 | 24D2h | 24D4h | 24D7h | 24DEh | RO |
| 04–05h | PCICMD | PCI Command | 0400h | 0400h | 0400h | 0400h | 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 |
| 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 | RO |
| 2E–2Fh | SID | Subsystem Identification | 0000h | 0000h | 0000h | 0000h | RO |
| 3Ch | INT_LN | Interrupt Line | 00h | 00h | 00h | 00h | R/W |
| 3Dh | INT_PN | Interrupt Pin | 01h | 02h | 03h | 01h | 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 |

**NOTE:** Refer to the latest *Intel® ICH5 / ICH5R Specification Update* for the value of the Revision Identification register.

## 12.1.1 VID—Vendor Identification Register (USB—D29:F0/F1/F2/F3)

Address Offset: 00–01h Attribute: RO
Default Value: 8086h Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | Vendor ID — RO. This is a 16-bit value assigned to Intel |

## 12.1.2 DID—Device Identification Register (USB—D29:F0/F1/F2/F3)

Address Offset: 02–03h Attribute: RO
Default Value: UHCI #1 = 24D2h Size: 16 bits
UHCI #2 = 24D4h
UHCI #3 = 24D7h
UHCI #4 = 24DEh

| Bit | Description |
|---|---|
| 15:0 | Device ID — RO. This is a 16-bit value assigned to the Intel® ICH5 USB host controllers |

## 12.1.3 PCICMD—PCI Command Register (USB—D29:F0/F1/F2/F3)

Address Offset: 04–05h Attribute: R/W, RO
Default Value: 0400h Size: 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. |
| 3 | Special Cycle Enable (SCE) — RO. Hardwired to 0. |
| 2 | **Bus Master Enable (BME)** — R/W.<br>0 = Disable<br>1 = Enable. Intel® ICH5 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. |

## 12.1.4 PCISTS—PCI Status Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| Address Offset: | 06–07h | Attribute: | R/WC, RO |
| Default Value: | 0280h | 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 |
|---|---|
| 15:14 | Reserved as 00b. 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 = Intel® ICH5 did Not terminate transaction for USB function with a target abort.<br>1 = USB function is targeted with a transaction that the ICH5 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 ICH5's DEVSEL# timing when performing a positive decode. ICH5 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. |
| 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 deasserted.<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 |

## 12.1.5 RID—Revision Identification Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| Address Offset: | 08h | Attribute: | RO |
| Default Value: | See bit description | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | Revision ID — RO.These bits contain device stepping information and are hardwired to the default value.<br><br>**NOTE:** Refer to the latest Intel® ICH5 / ICH5R Specification Update for the value of the Revision Identification register. |

## 12.1.6 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. |

## 12.1.7 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. |

## 12.1.8 BCC—Base Class Code Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| 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. |

## 12.1.9 HEADTYP—Header Type Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| Address Offset: | 0Eh | Attribute: | RO |
| Default Value: | FN 0: 80h<br>FN 1: 00h<br>FN 2: 00h<br>FN 3: 00h | Size: | 8 bits |

For functions 1, 2, and 3, this register is hardwired to 00h. For function 0, bit 7 is determined by the values in bits 15, 10, and 9 of the function disable register (D31:F0:F2h).

| Bit | Description |
|---|---|
| 7 | **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 Device 31, Function 0, Offset F2h as follows:<br>**D29:F7_Disable (bit 15) / D29:F3_Disable (bit 11) / D29:F2_Disable (bit 10) / D29:F1_Disable (bit 9) / Multi-Function Bit**<br>0 / X / X / X / 1<br>X / 0 / X / X / 1<br>X / X / 0 / X / 1<br>X / X / X / 0 / 1<br>1 / 1 / 1 / 1 / 0 |
| 6:0 | Configuration Layout. Hardwired to 00h, which indicates the standard PCI configuration layout. |

## 12.1.10 BASE—Base Address Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| Address Offset: | 20–23h | Attribute: | R/W, RO |
| Default Value: | 00000001h | Size: | 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. |

## 12.1.11 SVID — Subsystem Vendor Identification Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| 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. |

## 12.1.12 SID — Subsystem Identification Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| Address Offset: | 2Eh–2Fh | Attribute: | RO |
| Default Value: | 0000h | 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. |

## 12.1.13 INT_LN—Interrupt Line Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| Address Offset: | 3Ch | Attribute: | 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 Intel® ICH5. It is to communicate to software the interrupt line that the interrupt pin is connected to. |

## 12.1.14 INT_PN—Interrupt Pin Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| Address Offset: | 3Dh | Attribute: | RO |
| Default Value: | Function 0: 01h<br>Function 1: 02h<br>Function 2: 03h<br>Function 3: 01h | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:3 | Reserved |
| 2:0 | **Interrupt Pin (INT_PN)** — RO. The values of 01h, 02h, 03h, and 01h in function 0, 1, 2, and 3 respectively, indicate to software that the corresponding Intel® ICH5 classic USB controllers drive the INTA#, INTB#, INTC#, and INTA# PCI signals.<br>Note that this does not determine the mapping to the ICH5 PIRQ inputs. Function 0 drives PIRQA; function 1 drives PIRQD; function 2 drives PIRQC; function 3 drives PIRQA. |

## 12.1.15 USB_RELNUM—Serial Bus Release Number Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| Address Offset: | 60h | Attribute: | RO |
| Default Value: | 10h | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | **Serial Bus Release Number** — RO.<br>10h = USB controller is compliant with the *Universal Serial Bus Revision 2.0 Specification*. |

## 12.1.16 USB_LEGKEY—USB Legacy Keyboard/Mouse Control Register (USB—D29:F0/F1/F2/F3)

| | | | |
|---|---|---|---|
| Address Offset: | C0–C1h | Attribute: | R/W, R/WC, RO |
| Default Value: | 2000h | Size: | 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. |
| 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. |

| Bit | Description |
|---|---|
| 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. |

### 12.1.17 USB_RES—USB Resume Enable Register (USB—D29:F0/F1/F2/F3)

Address Offset: C4h
Default Value: 00h
Attribute: R/W
Size: 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. |

## 12.2 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 160. USB I/O Registers**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00–01 | USBCMD | USB Command | 0000h | R/W |
| 02–03 | USBSTS | USB Status | 0020h | R/WC |
| 04–05 | USBINTR | USB Interrupt Enable | 0000h | R/W |
| 06–07 | FRNUM | Frame Number | 0000h | R/W (see Note 1) |
| 08–0B | FRBASEADD | Frame List Base Address | Undefined | R/W |
| 0C | SOFMOD | Start of Frame Modify | 40h | R/W |
| 0D–0F | — | Reserved | — | — |
| 10–11 | PORTSC0 | Port 0 Status/Control | 0080h | R/WC, RO, R/W (see Note 1) |
| 12–13 | PORTSC1 | Port 1 Status/Control | 0080h | R/WC, RO, R/W (see Note 1) |
| 14–17 | — | Reserved | — | — |
| 18h | LOOPDATA | Loop Back Test Data | 00h | RO |

**NOTES:**
1. These registers are WORD writable only. Byte writes to these registers have unpredictable effects.

## 12.2.1 USBCMD—USB Command Register

| | | | |
|---|---|---|---|
| I/O Offset: | Base + (00–01h) | Attribute: | R/W |
| Default Value: | 0000h | Size: | 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 = Intel® ICH5 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 |
| 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. |

| Bit | Description |
|---|---|
| 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 *Universal Serial Bus Revision 2.0 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 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. |
| 0 | **Run/Stop (RS)** — R/W. When set to 1, the ICH5 proceeds with execution of the schedule. The ICH5 continues execution as long as this bit is set. When this bit is cleared, the ICH5 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 161. 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 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.

## 12.2.2 USBSTS—USB Status Register

| | | | |
|---|---|---|---|
| I/O Offset: | Base + (02–03h) | Attribute: | R/WC |
| Default Value: | 0020h | Size: | 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. |
| 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 illegal 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 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 (bit 3 of Command register = 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 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. |

## 12.2.3 USBINTR—USB Interrupt Enable Register

| | | | |
|---|---|---|---|
| I/O Offset: | Base + (04–05h) | Attribute: | R/W |
| Default Value: | 0000h | Size: | 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-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 |
|---|---|
| 15:4 | Reserved |
| 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. |

## 12.2.4 FRNUM—Frame Number Register

| | | | |
|---|---|---|---|
| I/O Offset: | Base + (06–07h) | Attribute: | R/W (Writes must be Word Writes) |
| Default Value: | 0000h | Size: | 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 (USBSTS register). A write to this register while the Run/Stop bit is set (USBCMD register) 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]. |

## 12.2.5 FRBASEADD—Frame List Base Address Register

| | | | |
|---|---|---|---|
| I/O Offset: | Base + (08–0Bh) | Attribute: | R/W |
| Default Value: | Undefined | Size: | 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 0s (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 |

## 12.2.6 SOFMOD—Start of Frame Modify Register

| | | | |
|---|---|---|---|
| I/O Offset: | Base + (0Ch) | Attribute: | R/W |
| Default Value: | 40h | Size: | 8 bits |

This 1-byte register is used to modify the value used in the generation of SOF timing on the USB. Only the seven, least significant bits are used. When a new value is written into these seven 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 *Universal Serial Bus Revision 2.0 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 *Universal Serial Bus Revision 2.0 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.<br>**Frame Length (# 12 MHz Clocks) (decimal)** / **SOF Reg. Value (decimal)**<br>11936 / 0<br>11937 / 1<br>. / .<br>. / .<br>11999 / 63<br>12000 / 64<br>12001 / 65<br>. / .<br>. / .<br>12062 / 126<br>12063 / 127 |

## 12.2.7 PORTSC[0,1]—Port Status and Control Register

| | | | |
|---|---|---|---|
| I/O Offset: | Port 0/2/4/6: Base + (10–11h)<br>Port 1/3/5/7: Base + (12–13h) | Attribute: | R/WC, RO,<br>R/W (Word writes only) |
| Default Value: | 0080h | Size: | 16 bits |

***Note:*** For Function 0, this applies to ICH5 USB ports 0 and 1; for Function 1, this applies to ICH5 USB ports 2 and 3; for Function 2, this applies to ICH5 USB ports 4 and 5; and for Function 3, this applies to ICH5 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 (SE0).

| Bit | Description |
|---|---|
| 15:13 | Reserved — RO. |
| 12 | **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>**Bits [12,2]** **Hub State**<br>X0 Disable<br>01 Enable<br>11 Suspend<br>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>**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 specific error condition (out transaction with babble), the Intel® ICH5 may issue a start-of-frame, and then suspend the port. |
| 11 | **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. |
| 10 | **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). |
| 9 | **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. |
| 8 | **Low Speed Device Attached (LS)** — RO.<br>0 = Full speed device is attached.<br>1 = Low speed device is attached to this port. |
| 7 | Reserved — RO. Always read as 1. |
| 6 | **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 ICH5 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. |

| Bit | Description |
|---|---|
| 5:4 | **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 *Universal Serial Bus Revision 2.0 Specification*). |
| 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 *Universal Serial Bus Revision 2.0 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. |

# EHCI Controller Registers (D29:F7) 13

## 13.1 USB EHCI Configuration Registers (USB EHCI—D29:F7)

*Note:* Register address locations that are not shown in Table 162 should be treated as Reserved (see Section 6.2 for details).

**Table 162. USB EHCI PCI Register Address Map (USB EHCI—D29:F7)**

| Offset | Mnemonic | Register Name | Default Value | Type |
|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086h | RO |
| 02–03h | DID | Device Identification | 24DDh | RO |
| 04–05h | PCICMD | PCI Command | 0400h | R/W, RO |
| 06–07h | PCISTS | PCI Status | 0290h | R/WC, 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 |
| 10–13h | MEM_BASE | Memory Base Address | 00000000h | R/W, RO |
| 2C–2Dh | SVID | USB EHCI Subsystem Vendor Identification | XXXXh | R/W (special) |
| 2E–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 | 04h | RO |
| 50h | PWR_CAPID | PCI Power Management Capability ID | 01h | RO |
| 51h | NXT_PTR1 | Next Item Pointer | 58h | R/W (special) |
| 52–53h | PWR_CAP | Power Management Capabilities | C9C2h | R/W (special) |
| 54–55h | 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 |
| 5A–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 |
| 62–63h | PWAKE_CAP | Port Wake Capabilities | 01FFh | R/W |
| 66–67h | — | Reserved | — | — |

**Table 162. USB EHCI PCI Register Address Map (USB EHCI—D29:F7)**

| Offset | Mnemonic | Register Name | Default Value | Type |
|---|---|---|---|---|
| 68–6Bh | LEG_EXT_CAP | USB EHCI Legacy Support Extended Capability | 00000001h | R/W, RO |
| 6C–6Fh | LEG_EXT_CS | USB EHCI Legacy Extended Support Control/Status | 00000000h | R/W, R/WC, RO |
| 70–73h | SPECIAL_SMI | Intel Specific USB 2.0 SMI | 00000000h | R/W, R/WC |
| 74–7Fh | — | Reserved | — | — |
| 80h | ACCESS_CNTL | Access Control | 00h | R/W |

## 13.1.1 VID—Vendor Identification Register (USB EHCI—D29:F7)

Offset Address: 00–01h
Default Value: 8086h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | Vendor ID — RO. This is a 16-bit value assigned to Intel. |

## 13.1.2 DID—Device Identification Register (USB EHCI—D29:F7)

Offset Address: 02–03h
Default Value: 24DDh
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | Device ID — RO. This is a 16-bit value assigned to the Intel® ICH5 USB EHCI controller. |

## 13.1.3 PCICMD—PCI Command Register (USB EHCI—D29:F7)

Address Offset: 04–05h
Default Value: 0400h
Attribute: R/W, RO
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 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 the hub interface (and subsequently on its internal interface). |
| 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. |
| 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 Intel® ICH5 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 for USB 2.0 should be programmed before this bit is set. |
| 0 | I/O Space Enable (IOSE) — RO. Hardwired to 0. |

## 13.1.4 PCISTS—PCI Status Register (USB EHCI—D29:F7)

Address Offset: 06–07h
Default Value: 0290h
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) — RO. Hardwired to 0. |
| 14 | **Signaled System Error (SSE)** — R/WC.<br>0 = No SERR# signaled by Intel® ICH5.<br>1 = This bit is set by the ICH5 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/WC.<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/WC.<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. |
| 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 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/WC.<br>0 = No data parity error detected on USB2.0 read completion packet.<br>1 = This bit is set by the ICH5 when a data parity error is detected on a USB 2.0 read completion packet on the internal interface to the EHCI host controller (due to an equivalent data parity error on hub interface) 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 deasserted.<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 |

### 13.1.5 RID—Revision Identification Register (USB EHCI—D29:F7)

Offset Address: 08h
Default Value: See table below
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Revision ID — RO. These bits contain device stepping information and are hardwired to the default value.<br><br>**NOTE:** Refer to the latest Intel® ICH5 / ICH5R Specification Update for the value of the Revision Identification register. |

### 13.1.6 PI—Programming Interface Register (USB EHCI—D29:F7)

Address Offset: 09h
Default Value: 20h
Attribute: RO
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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0*. |

### 13.1.7 SCC—Sub Class Code Register (USB EHCI—D29:F7)

Address Offset: 0Ah
Default Value: 03h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Sub Class Code (SCC) — RO.<br>03h = Universal serial bus host controller. |

### 13.1.8 BCC—Base Class Code Register (USB EHCI—D29:F7)

Address Offset: 0Bh
Default Value: 0Ch
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Base Class Code (BCC) — RO.<br>0Ch = Serial bus controller. |

## 13.1.9 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 via hub interface (and not PCI), it does not need a master latency timer. |

## 13.1.10 MEM_BASE—Memory Base Address Register (USB EHCI—D29:F7)

Address Offset: 10–13h
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. |

## 13.1.11 SVID—USB EHCI Subsystem Vendor ID Register (USB EHCI—D29:F7)

Address Offset: 2C–2Dh
Default Value: XXXXh
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 (offset 80h, bit 0) is set to 1. |

## 13.1.12 SID—USB EHCI Subsystem ID Register (USB EHCI—D29:F7)

Address Offset: 2E–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 (offset 80h, bit 0) is set to 1. |

## 13.1.13 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. |

## 13.1.14 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 Intel® ICH5. It is used as a scratchpad register to communicate to software the interrupt line that the interrupt pin is connected to. |

## 13.1.15 INT_PN—Interrupt Pin Register (USB EHCI—D29:F7)

Address Offset: 3Dh
Default Value: 04h
Attribute: RO
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | Interrupt Pin — RO. The value of 04h indicates that the EHCI function within the Intel® ICH5's multi-function USB device will drive the fourth interrupt pin from the device- INTD# in PCI terms. The value of 04h in function 7 is required because the *PCI Local Bus Specification, Revision 2.3* doesn't recognize more than 4 interrupts and older APM-based OSs require that each function within a multi-function device has a different Interrupt Pin Register value.<br><br>**NOTE:** Internally the EHCI controller uses PIRQH#. |

## 13.1.16 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. |

## 13.1.17 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 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. |

## 13.1.18 PWR_CAP—Power Management Capabilities Register (USB EHCI—D29:F7)

Address Offset: 52–53h
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 Intel® ICH5 EHC does not support the D1 or D2 states. For all other states, the ICH5 EHC is capable of generating PME#. Software should never 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 ICH5 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 ICH5 reports 0, indicating that no device-specific initialization is required. |
| 4 | Reserved |
| 3 | **PME Clock (PME_CLK)**— R/W (special). The ICH5 reports 0, indicating that no PCI clock is required to generate PME#. |
| 2:0 | **Version (VER)** — R/W (special). The ICH5 reports 010b, indicating that it complies with the *PCI Power Management Specification, Revision 1.1* |

**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 ICH5 is used, bits 15:11 and 8:6 in this register are writable when the WRT_RDONLY bit 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.

## 13.1.19 PWR_CNTL_STS—Power Management Control/Status Register (USB EHCI—D29:F7)

Address Offset: 54–55h
Default Value: 0000h
Attribute: R/W, R/WC, RO
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 Intel® ICH5 EHC would normally assert the PME# signal independent of the state of the PME_En bit.<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 ICH5 EHC to generate an internal PME signal when PME_Status is 1.<br>**NOTE:** This bit must be explicitly cleared by the operating system each time it is initially loaded. |
| 7:2 | Reserved |
| 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 ICH5 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 ICH5 when not in the D0 state.<br>When software changes this value from the D3hot 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.

## 13.1.20 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. |

## 13.1.21 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. |

## 13.1.22 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. |

## 13.1.23 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*. |

## 13.1.24 FL_ADJ—Frame Length Adjustment Register (USB EHCI—D29:F7)

| | | | |
|---|---|---|---|
| Address Offset: | 61h | Attribute: | R/W |
| Default Value: | 20h | 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 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.

| 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.<br>**Frame Length (# 480 MHz Clocks)** / **FLADJ Value decimal (hex)**<br>59488 — 0 (00h)<br>59504 — 1 (01h)<br>59520 — 2 (02h)<br>…<br>59984 — 31 (1Fh)<br>60000 — 32 (20h)<br>…<br>60480 — 62 (3Eh)<br>60496 — 63 (3Fh) |

## 13.1.25 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.

| 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 1, bit position 2 corresponds to port 2, etc. |
| 0 | **Port Wake Implemented** — R/W. A 1 in this bit indicates that this register is implemented to software. |

## 13.1.26 LEG_EXT_CAP—USB EHCI Legacy Support Extended Capability Register (USB EHCI—D29:F7)

Address Offset: 68–6Bh
Default Value: 00000001h
Power Well: Suspend
Attribute: R/W, RO
Size: 32 bits

| 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. |

## 13.1.27 LEG_EXT_CS—USB EHCI Legacy Support Extended Control / Status Register (USB EHCI—D29:F7)

Address Offset: 6C–6Fh
Default Value: 00000000h
Power Well: Suspend
Attribute: R/W, R/WC, RO
Size: 32 bits

| 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 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 in the USB2.0_STS register.<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.<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 in the USB2.0_STS register.<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 in the USB2.0_STS register.<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 in the USB2.0_STS register.<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 in the USB2.0_STS register.<br>**NOTE:** To clear this bit system software must write a 1 to the USB Interrupt bit in the USB2.0_STS register. |
| 15 | **SMI on BAR Enable** — R/W.<br>0 = Disable<br>1 = Enable. When this bit is 1 and SMI on BAR 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 is 1, then the host controller will issue an SMI. |

| Bit | Description |
|---|---|
| 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 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 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 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 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 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 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 is a 1, the host controller will issue an SMI immediately. |

## 13.1.28 SPECIAL_SMI—Intel Specific USB 2.0 SMI Register (USB EHCI—D29:F7)

Address Offset: 70–73h
Default Value: 00000000h
Power Well: Suspend
Attribute: R/W, R/WC
Size: 32 bits

| 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. |
| 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. |

| Bit | Description |
|---|---|
| 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 |
| 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— R/W. |

## 13.1.29 ACCESS_CNTL—Access Control Register (USB EHCI—D29:F7)

Address Offset: 80h
Default Value: 00h
Attribute: R/W
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 |

## 13.2 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 ICH5 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 is not set in the Command register in configuration space, the memory range will not be decoded by the ICH5 Enhanced Host Controller (EHC). If the MSE bit is not set, then the ICH5 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.

### 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. This register is 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 163. Enhanced Host Controller Capability Registers**

| Offset | Mnemonic | Register | Default | Type |
|---|---|---|---|---|
| 00h | CAPLENGTH | Capabilities Registers Length | 20h | RO |
| 02–03h | HCIVERSION | Host Controller Interface Version Number | 0100h | RO |
| 04–07h | HCSPARAMS | Host Controller Structural Parameters | 00104208h | R/W (special), RO |
| 08–0Bh | 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.

## 13.2.1 CAPLENGTH—Capability Registers Length Register

Offset: 00h
Default Value: 20h
Attribute: RO
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 to find the beginning of the Operational Register Space. This field is hardwired to 20h indicating that the Operation Registers begin at offset 20h. |

## 13.2.2 HCIVERSION—Host Controller Interface Version Number Register

Offset: 02–03h
Default Value: 0100h
Attribute: RO
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. |

## 13.2.3 HCSPARAMS—Host Controller Structural Parameters

Offset: 04–07h
Default Value: 00104208h
Attribute: R/W (special), RO
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) — R/W (special). Hardwired to 1h indicating that the Debug Port is on the lowest numbered port on the Intel® ICH5. |
| 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 larger than 1 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 ICH5 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 ICH5 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.

## 13.2.4 HCCPARAMS—Host Controller Capability Parameters Register

Offset: 08–0Bh
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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* 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 *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><br>**NOTE:** Intel® ICH5 only implements 44 bits of addressing. Bits 63:44 will always be 0. |

## 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. All registers are 32 bits in length.

### Table 164. Enhanced Host Controller Operational Register Address Map

| Offset (CAPLENGTH+) | Mnemonic | Register Name | Default | Special Notes | Type |
|---|---|---|---|---|---|
| 00–03h | USB2.0_CMD | USB 2.0 Command | 00080000h | | R/W, RO |
| 04–07h | USB2.0_STS | USB 2.0 Status | 00001000h | | R/WC, RO |
| 08–0Bh | USB2.0_INTR | USB 2.0 Interrupt Enable | 00000000h | | R/W |
| 0C–0Fh | FRINDEX | USB 2.0 Frame Index | 00000000h | | R/W, |
| 10–13h | CTRLDS-SEGMENT | Control Data Structure Segment | 00000000h | | R/W, RO |
| 14–17h | PERODI-CLISTBASE | Period Frame List Base Address | 00000000h | | R/W |
| 18–1Bh | ASYNCLIS-TADDR | Current Asynchronous List Address | 00000000h | | R/W |
| 1C–3Fh | — | Reserved | 0h | | RO |
| 40–43h | CONFIGGLAG | Configure Flag | 00000000h | Suspend | R/W |
| 44–47h | PORT0SC | Port 0 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 48–4Bh | PORT1SC | Port 1 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 4C–4Fh | PORT2SC | Port 2 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 50–53h | PORT3SC | Port 3 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 54–57h | PORT4SC | Port 4 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 58–5Bh | PORT5SC | Port 5 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 5C–5Fh | PORT6SC | Port 6 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 60–63h | PORT7SC | Port 7 Status and Control | 00003000h | Suspend | R/W, R/WC, RO |
| 64–7Fh | — | Reserved | Undefined | | RO |
| 80–93h | — | Debug Port Registers | Undefined | | See register description |
| 94–3FFh | — | 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 00:3Fh 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 40h 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

## 13.2.5 USB2.0_CMD—USB 2.0 Command Register

Offset: CAPLENGTH + 00–03h Attribute: RW, RO
Default Value: 00080000h Size: 32 bits

| Bit | Description |
|---|---|
| 31:24 | Reserved. These bits are reserved and should be set to 0 when writing this register. |
| 23:16 | **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.<br>**Value** **Maximum Interrupt Interval**<br>00h Reserved<br>01h 1 micro-frame<br>02h 2 micro-frames<br>04h 4 micro-frames (default)<br>08h 8 micro-frames (default, equates to 1 ms)<br>10h 16 micro-frames (2 ms)<br>20h 32 micro-frames (4 ms)<br>40h 64 micro-frames (8 ms) |
| 15:8 | Reserved. These bits are reserved and should be set to 0 when writing this register. |
| 11:8 | Unimplemented Asynchronous Park Mode Bits. Hardwired to 000b indicating the host controller does not support this optional feature. |
| 7 | Light Host Controller Reset — RO. Hardwired to 0. The Intel® ICH5 does not implement this optional reset. |
| 6 | **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 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 is a 1 then the host controller will assert an interrupt at the next interrupt threshold. See the *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* for operational details.<br>**NOTE:** Software should not write a 1 to this bit when the asynchronous schedule is inactive. Doing so will yield undefined results. |
| 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. |

| Bit | Description |
|---|---|
| 3:2 | **Frame List Size** — RO. The ICH5 hardwires this field to 00b because it only supports the 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 (i.e., RSMRST# assertion and PWROK deassertion on the ICH5).<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>**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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0.* 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 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. |
| 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 (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>**Run/Stop** / **Halted** / **Interpretation**<br>0 / 0 / Valid- in the process of halting<br>0 / 1 / Valid- halted<br>1 / 0 / Valid- running<br>1 / 1 / Invalid- the HCHalted bit clears immediately.<br>Memory read cycles initiated by the EHC that receive any status other than Successful will result in this bit being cleared. |

**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.

## 13.2.6 USB2.0_STS—USB 2.0 Status Register

Offset: CAPLENGTH + 04–07h 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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* 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 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). |
| 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 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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0.* |
| 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 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 to prevent further execution of the scheduled TDs. A hardware interrupt is generated to the system (if enabled in the Interrupt Enable Register). |

| Bit | Description |
|---|---|
| 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 Intel® ICH5 only supports the 1024-entry Frame List Size, the *Frame List Index* rolls over every time FRNUM13 toggles. |
| 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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* 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). |

## 13.2.7 USB2.0_INTR—USB 2.0 Interrupt Enable Register

Offset: CAPLENGTH + 08–0Bh 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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0*, 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 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. |
| 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 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 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 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 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 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. |

## 13.2.8 FRINDEX—Frame Index Register

| | | | |
|---|---|---|---|
| Offset: | CAPLENGTH + 0C–0Fh | 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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* 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 never 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 ICH5 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 (USB2.0_STS register). A write to this register while the Run/Stop bit 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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* 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. |

## 13.2.9 CTRLDSSEGMENT—Control Data Structure Segment Register

| | | | |
|---|---|---|---|
| Offset: | CAPLENGTH + 10–13h | 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 ICH5 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 Intel® ICH5 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. |

## 13.2.10 PERIODICLISTBASE—Periodic Frame List Base Address Register

| | | | |
|---|---|---|---|
| Offset: | CAPLENGTH + 14–17h | 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 ICH5 host controller operates in 64-bit mode (as indicated by the 1 in the 64-bit Addressing Capability field in the HCCSPARAMS register), 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. |

## 13.2.11 ASYNCLISTADDR—Current Asynchronous List Address Register

Offset: CAPLENGTH + 18–1Bh 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 ICH5 host controller operates in 64-bit mode (as indicated by a 1 in 64-bit Addressing Capability field in the HCCPARAMS register), then the most significant 32 bits of every control data structure address comes from the CTRLDSSEGMENT register. Bits [4:0] of this register cannot be modified by system software and will always return 0s 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. |

## 13.2.12 CONFIGFLAG—Configure Flag Register

Offset: CAPLENGTH + 40–43h 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 ICH5 host controller operates in 64-bit mode (as indicated by a 1 in 64-bit Addressing Capability field in the HCCPARAMS register), then the most significant 32 bits of every control data structure address comes from the CTRLDSSEGMENT register. Bits [4:0] of this register cannot be modified by system software and will always return 0s when read. The memory structure referenced by this physical memory pointer is assumed to be 32-byte aligned.

| 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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* 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. |

## 13.2.13 PORTSC—Port N Status and Control Register

Offset: Port 0: CAPLENGTH + 44–47h
Port 1: CAPLENGTH + 48–4Bh
Port 2: CAPLENGTH + 4C–4Fh
Port 3: CAPLENGTH + 50–53h
Port 4: CAPLENGTH + 54–57h
Port 5: CAPLENGTH + 58–5Bh
Port 6: CAPLENGTH + 5C–5Fh
Port 7: CAPLENGTH + 60–63h
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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* 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 0s when read. |
| 22 | **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. |
| 21 | **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). |
| 20 | **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). |

| Bit | Description |
|---|---|
| 19:16 | **Port Test Control** — R/W. When this field is 0s, 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>**Bits** **Test Mode**<br>0000b Test mode not enabled (Default)<br>0001b Test J_STATE<br>0010b Test K_STATE<br>0011b Test SE0_NAK<br>0100b Test Packet<br>0101b Test FORCE_ENABLE<br>Refer to *Universal Serial Bus Revision 2.0 Specification*, Chapter 7 for details on each test mode. |
| 15:14 | Reserved — R/W. Should be written to =00b. |
| 13 | **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 *Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 1.0* for operational details. |
| 12 | **Port Power (PP)** — RO. Read-only with a value of 1. This indicates that the port does have power. |
| 11:10 | **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 |
| 9 | Reserved. This bit will return a 0 when read. |
| 8 | **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 *Universal Serial Bus Revision 2.0 Specification* 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 guarantee the reset sequence completes as specified in the *Universal Serial Bus Revision 2.0 Specification*.<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>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 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. |

| Bit | Description |
|---|---|
| 7 | **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>**Port Enabled, Suspend Bits** — **Port State**<br>0, X — Disable<br>1, 0 — Enable<br>1, 1 — Suspend<br>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. |
| 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 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>**NOTE:** When the EHCI controller owns the port, the resume sequence follows the defined sequence documented in the *Universal Serial Bus Revision 2.0 Specification*. 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 Intel® ICH5 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 *Universal Serial Bus Revision 2.0 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. |

| Bit | Description |
|---|---|
| 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). |
| 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. |

## 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 (BAR), 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. The specific EHCI port that supports this debug capability 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 165.

### Table 165. Debug Port Register Address Map

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| A0h | CNTL_STS | Control/Status | 0000h | R/W, R/WC, RO, WO |
| A4h | USBPID | USB PIDs | 00h | R/W, RO |
| A8h | DATABUF[3:0] | Data Buffer (Bytes 3:0) | 00000000h | R/W |
| ACh | DATABUF[7:4] | Data Buffer (Bytes 7:4) | 00000000h | R/W |
| B0h | CONFIG | Configuration | 00007F01h | R/W |

**NOTES:**
1. All of these registers are implemented in the core well and reset by PCIRST#, 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 illegally is undefined.

## 13.2.14 CNTL_STS—Control/Status Register

Offset: A0h
Default Value: 0000h
Attribute: R/W, R/WC, RO, WO
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. |
| 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. |

| Bit | Description |
|---|---|
| 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 affect 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 |
| 3:0 | **DATA_LEN_CNT —** R/W. This field is used to indicate the size of the data to be transferred. 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 illegal 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.

## 13.2.15 USBPID—USB PIDs Register

| | | | |
|---|---|---|---|
| Offset: | A4h | Attribute: | RW, 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. |

## 13.2.16 DATABUF[7:0]—Data Buffer Bytes[7:0] Register

| | | | |
|---|---|---|---|
| Offset: | A8–AFh | Attribute: | R/W |
| Default Value: | 0000000000000000h | Size: | 64 bits |

This register can be accessed as eight, separate 8-bit registers or two, 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 is cleared by the hardware, ERROR_GOOD#_STS is cleared by the hardware, and the DATA_LENGTH_CNT field indicates the number of bytes that are valid. |

## 13.2.17 CONFIG—Configuration Register

| | | | |
|---|---|---|---|
| Offset: | 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) |

# SMBus Controller Registers (D31:F3) 14

## 14.1 PCI Configuration Registers (SMBUS—D31:F3)

*Note:* Registers address locations that are not shown in Table 166 should be treated as Reserved (See Section 6.2 for details).

**Table 166. SMBus Controller PCI Register Address Map (SMBUS—D31:F3)**

| Offset | Mnemonic | Register Name | Default | Type |
|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086 | RO |
| 02–03h | DID | Device Identification | 24D3h | RO |
| 04–05h | PCICMD | PC I Command | 0000h | R/W, RO |
| 06–07h | PCISTS | PCI Status | 0280h | RO, R/WC |
| 08h | RID | Revision Identification | See register description | RO |
| 0Ah | SCC | Sub Class Code | 05h | RO |
| 0Bh | BCC | Base Class Code | 0Ch | RO |
| 20–23h | SMB_BASE | SMBus Base Address | 00000001h | R/W, RO |
| 2C–2Dh | SVID | Subsystem Vendor Identification | 00h | RO |
| 2E–2Fh | SID | Subsystem Identification | 00h | R/WO |
| 3Ch | INT_LN | Interrupt Line | 00h | R/W |
| 3Dh | INT_PN | Interrupt Pin | 02h | RO |
| 40h | HOSTC | Host Configuration | 00h | R/W |

### 14.1.1 VID—Vendor Identification Register (SMBUS—D31:F3)

Address: 00–01h
Default Value: 8086h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | Vendor ID — RO. This is a 16-bit value assigned to Intel. |

## 14.1.2 DID—Device Identification Register (SMBUS—D31:F3)

Address: 02–03h
Default Value: 24D3h
Attribute: RO
Size: 16 bits

| Bit | Description |
|---|---|
| 15:0 | Device ID — RO. |

## 14.1.3 PCICMD—PCI Command Register (SMBUS—D31:F3)

Address: 04–05h
Default Value: 0000h
Attributes: RO, R/W
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) — RO. Hardwired to 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. |
| 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. |

## 14.1.4 PCISTS—PCI Status Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Address: | 06–07h | Attributes: | RO, R/WC |
| Default Value: | 0280h | 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) — RO. Hardwired to 0. |
| 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 = Intel® ICH5 did **not** terminate transaction for this function with a target abort.<br>1 = The function is targeted with a transaction that the ICH5 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 |

## 14.1.5 RID—Revision Identification Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Offset Address: | 08h | Attribute: | RO |
| Default Value: | See Bit Description | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | Revision ID (RID) — RO. Refer to the latest Intel® ICH5 / ICH5R Specification Update for the value of the Revision Identification register. |

## 14.1.6 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. |

## 14.1.7 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. |

## 14.1.8 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 Intel® ICH5 SMB logic. |
| 4:1 | Reserved — RO |
| 0 | IO Space Indicator — RO. Hardwired to 1 indicating that the SMB logic is I/O mapped. |

## 14.1.9 SVID — Subsystem Vendor Identification Register (SMBUS—D31:F2/F4)

Address Offset: 2Ch–2Dh
Default Value: 00h
Lockable: No
Attribute: RO
Size: 16 bits
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. |

## 14.1.10 SID — Subsystem Identification Register (SMBUS—D31:F2/F4)

| | | | |
|---|---|---|---|
| Address Offset: | 2Eh–2Fh | Attribute: | RO |
| 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. |

## 14.1.11 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 Intel® ICH5. It is to communicate to software the interrupt line that the interrupt pin is connected to PIRQB#. |

## 14.1.12 INT_PN—Interrupt Pin Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Address Offset: | 3Dh | Attributes: | RO |
| Default Value: | 02h | Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | Interrupt PIN (INT_PN) — RO.<br>02h = Indicates that the Intel® ICH5 SMBus controller will drive PIRQB# as its interrupt line. |

## 14.1.13 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 | **$I^2C$_EN** — R/W.<br>0 = SMBus behavior.<br>1 = The Intel® ICH5 is enabled to communicate with $I^2C$ 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 (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 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. |

## 14.2 SMBus I/O Registers

### Table 167. SMBus I/O Register Address Map

| 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 | SLV_DATA | Receive Slave Data | 0000h | RO |
| 0Ch | AUX_STS | Auxiliary Status | 00h | R/WC |
| 0Dh | AUX_CTL | Auxiliary Control | 00h | R/W |
| 0Eh | SMLINK_PIN_CTL | SMLink Pin Control (TCO Compatible Mode) | See register description | R/W, RO |
| 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 |

## 14.2.1 HST_STS—Host Status Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 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.

| 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 D110 interface heartbeat.<br>This bit has no meaning for block transfers when the 32-byte buffer is enabled.<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 BYTE_DONE_STS 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 Intel® ICH5 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 ICH5'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 never be set. |
| 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. |

| Bit | Description |
|---|---|
| 2 | **DEV_ERR** — R/WC.<br>0 = Software clears this bit by writing a 1 to it. The ICH5 will then deassert the interrupt or SMI#.<br>1 = The source of the interrupt or SMI# was due to one of the following:<br>•Illegal 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 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 ICH5 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 ICH5 when the current transaction is completed.<br>1 = Indicates that the ICH5 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. |

## 14.2.2 HST_CNT—Host Control Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 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 Intel® ICH5 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>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 ICH5 to send a NACK (instead of an ACK) after receiving the last byte.<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 ICH5 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 ICH5 is to perform. If enabled, the ICH5 will generate an interrupt or SMI# when the command has completed If the value is for a non-supported or reserved command, the ICH5 will set the device error (DEV_ERR) status bit and generate an interrupt when the START bit is set. The ICH5 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 ICH5 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. |

## 14.2.3 HST_CMD—Host Command Register (SMBUS—D31:F3)

Register Offset: 03h
Default Value: 00h
Attribute: R/W
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. |

## 14.2.4 XMIT_SLVA—Transmit Slave Address Register (SMBUS—D31:F3)

Register Offset: 04h
Default Value: 00h
Attribute: R/W
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 |

## 14.2.5 HST_D0—Host Data 0 Register (SMBUS—D31:F3)

Register Offset: 05h
Default Value: 00h
Attribute: R/W
Size: 8 bits

| Bit | Description |
|---|---|
| 7:0 | **Data0/Count** — R/W. This field contains the eight 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 illegal block counts. |

## 14.2.6 HST_D1—Host Data 1 Register (SMBUS—D31:F3)

Register Offset: 06h
Default Value: 00h
Attribute: R/W
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. |

## 14.2.7 Host_BLOCK_DB—Host Block Data Byte Register (SMBUS—D31:F3)

Register Offset: 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 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.<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. |

## 14.2.8 PEC—Packet Error Check (PEC) Register (SMBUS—D31:F3)

Register Offset: 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. |

## 14.2.9 RCV_SLVA—Receive Slave Address Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 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 Intel® ICH5 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 PCIRST#. |

## 14.2.10 SLV_DATA—Receive Slave Data Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 0Ah | 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 PCIRST#

| Bit | Description |
|---|---|
| 15:8 | **Data Message Byte 1 (DATA_MSG1)** — RO. See Section 5.21.7 for a discussion of this field. |
| 7:0 | **Data Message Byte 0 (DATA_MSG0)** — RO. See Section 5.21.7 for a discussion of this field. |

## 14.2.11 AUX_STS—Auxiliary Status Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 0Ch | Attribute: | RW/C, 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 = Intel® ICH5 is in the compatible TCO mode.<br>1 = ICH5 is in the advanced TCO mode. |
| 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 ICH5 has received the final data bit transmitted by an external slave. |

## 14.2.12 AUX_CTL—Auxiliary Control Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 0Dh | Attribute: | RW |
| 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 Intel® ICH5 generates an interrupt. |
| 0 | **Automatically Append CRC (AAC)** — R/W.<br>0 = ICH5 will **not** automatically append the CRC.<br>1 = The ICH5 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. |

## 14.2.13 SMLINK_PIN_CTL—SMLink Pin Control Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 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 = Intel® ICH5 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 |

## 14.2.14 SMBUS_PIN_CTL—SMBUS Pin Control Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 0Fh | Attribute: | R/W, RO |
| Default Value: | See below | 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 = Intel® ICH5 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 |

## 14.2.15 SLV_STS—Slave Status Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 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 Intel® ICH5 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 ICH5 will allow the Notify Address and Data registers to be over-written once this bit has been cleared. When this bit is 1, the ICH5 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. |

## 14.2.16 SLV_CMD—Slave Command Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 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. 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 |
| 0 | **HOST_NOTIFY_INTREN** — R/W. Software sets this bit to 1 to enable the generation of interrupt or SMI# when HOST_NOTIFY_STS 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, Off40h, B1). 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 |

## 14.2.17 NOTIFY_DADDR—Notify Device Address Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 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 *System Management Bus (SMBus) Specification, Version 2.0*. Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1. |
| 0 | Reserved |

## 14.2.18 NOTIFY_DLOW—Notify Data Low Byte Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 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 *System Management Bus (SMBus) Specification, Version 2.0.* Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1. |

## 14.2.19 NOTIFY_DHIGH—Notify Data High Byte Register (SMBUS—D31:F3)

| | | | |
|---|---|---|---|
| Register Offset: | 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 *System Management Bus (SMBus) Specification, Version 2.0.* Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1. |

This page is intentionally left blank.

# AC '97 Audio Controller Registers (D31:F5) 15

## 15.1 AC '97 Audio PCI Configuration Space (Audio— D31:F5)

*Note:* Address locations that are not shown in Table 168 should be treated as Reserved (see Section 6.2 for details).

**Table 168. AC '97 Audio PCI Register Address Map (Audio—D31:F5)**

| Offset | Mnemonic | Register Name | Default | Access |
|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086h | RO |
| 02–03h | DID | Device Identification | 24D5h | RO |
| 04–05h | PCICMD | PCI Command | 0000 | R/W, RO |
| 06–07h | PCISTS | PCI Status | 0280h | R/WC, RO |
| 08h | RID | Revision Identification | See register description | RO |
| 09h | PI | Programming Interface | 00 | RO |
| 0Ah | SCC | Sub Class Code | 01h | RO |
| 0Bh | BCC | Base Class Code | 04h | RO |
| 0Eh | HEADTYP | Header Type | 00h | RO |
| 10–13h | NAMBBAR | Native Audio Mixer Base Address | 00000001h | RO |
| 14–17h | NAMBBAR | Native Audio Bus Mastering Base Address | 00000001h | R/W, RO |
| 18–1Bh | MMBAR | Mixer Base Address (Mem) | 00000000h | R/W, RO |
| 1C–1Fh | MBBAR | Bus Master Base Address (Mem) | 00000000h | R/W, RO |
| 2C–2Dh | SVID | Subsystem Vendor Identification | 0000h | R/WO |
| 2E–2Fh | SID | Subsystem Identification | 0000h | R/WO |
| 34h | CAP_PTR | Capabilities Pointer | 50h | RO |
| 3Ch | INT_LN | Interrupt Line | 00h | R/W |
| 3Dh | INT_PN | Interrupt Pin | 02h | RO |
| 40h | PCID | Programmable Codec ID | 09h | R/W |
| 41h | CFG | Configuration | 00h | R/W |
| 50–51h | PID | PCI Power Management Capability ID | 0001h | RO |
| 52–53h | PC | PC -Power Management Capabilities | C9C2h | RO |
| 54–55h | PCS | Power Management Control and Status | 0000h | R/W, R/WC |

*Note:* Internal reset as a result of $D3_{HOT}$ to D0 transition will reset all the core well registers except the following BIOS programmed registers as BIOS may not be invoked following the D3-to-D0 transition. All resume well registers will not be reset by the $D3_{HOT}$ to D0 transition.

Core well registers **not** reset by the $D3_{HOT}$ to D0 transition:

- offset 2Ch–2Dh – Subsystem Vendor ID (SVID)
- offset 2Eh–2Fh – Subsystem ID (SID)
- offset 40h – Programmable Codec ID (PCID)
- offset 41h – Configuration (CFG)

Resume well registers **will not** be reset by the $D3_{HOT}$ to D0 transition:

- offset 54h–55h – Power Management Control and Status (PCS)
- Bus Mastering Register: Global Status Register, bits 17:16
- Bus Mastering Register: SDATA_IN MAP register, bits 7:3

## 15.1.1 VID—Vendor Identification Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Offset: | 00–01h | Attribute: | RO |
| Default Value: | 8086h | Size: | 16 Bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | Vendor ID. This is a 16-bit value assigned to Intel. |

## 15.1.2 DID—Device Identification Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Offset: | 02–03h | Attribute: | RO |
| Default Value: | 24D5h | Size: | 16 Bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | Device ID. |

## 15.1.3 PCICMD—PCI Command Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 04–05h | Attribute: | R/W, RO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

PCICMD is a 16-bit control register. Refer to the *PCI Local Bus Specification, Revision 2.3 f*or complete details on each bit.

| Bit | Description |
|---|---|
| 15:11 | Reserved. Read 0. |
| 10 | **Interrupt Disable (ID)** — R/W.<br>0 = The INTx# signals may be asserted and MSIs may be generated.<br>1 = The AC '97 controller's INTx# signal will be de-asserted and it may not generate MSIs. |
| 9 | Fast Back to Back Enable (FBE) — RO. Not implemented. Hardwired to 0. |
| 8 | SERR# Enable (SERR_EN) — RO. Not implemented. Hardwired to 0. |
| 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. |
| 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 bus mastering capabilities.<br>0 = Disable<br>1 = Enable |
| 1 | **Memory Space Enable (MSE)** — R/W. Enables memory space addresses to the AC '97 audio controller.<br>0 = Disable<br>1 = Enable |
| 0 | **I/O Space Enable (IOSE)** — R/W. This bit controls access to the AC '97 audio controller I/O space registers.<br>0 = Disable (Default).<br>1 = Enable access to I/O space. The Native PCI Mode Base Address register should be programmed prior to setting this bit.<br><br>**NOTE:** This bit becomes writable when the IOSE bit in offset 41h is set. If at any point software decides to clear the IOSE bit, software must first clear the IOS bit. |

## 15.1.4 PCISTS—PCI Status Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Offset: | 06–07h | Attribute: | RO, R/WC |
| Default Value | 0280h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

PCISTA is a 16-bit status register. Refer to the *PCI Local Bus Specification, Revision 2.3* for complete details on each bit.

| Bit | Description |
|---|---|
| 15 | Detected Parity Error (DPE). Not implemented. Hardwired to 0. |
| 14 | Signaled System Error (SSE) — RO. Not implemented. Hardwired to 0. |
| 13 | **Master Abort Status (MAS)** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = No master abort generated.<br>1 = Bus Master AC '97 2.3 interface function, as a master, generates a master abort. |
| 12 | Reserved — RO. Will always read as 0. |
| 11 | Signaled Target Abort (STA) — RO. Not implemented. Hardwired to 0. |
| 10:9 | DEVSEL# Timing Status (DEV_STS) — RO. This 2-bit field reflects the Intel® ICH5's DEVSEL# timing when performing a positive decode.<br>01b = Medium timing. |
| 8 | Data Parity Error Detected (DPED) — RO. Not implemented. Hardwired to 0. |
| 7 | Fast Back to Back Capable (FB2BC) — RO. Hardwired to 1. This bit indicates that the ICH5 as a target is capable of fast back-to-back transactions. |
| 6 | UDF Supported — RO. Not implemented. Hardwired to 0. |
| 5 | 66 MHz Capable (66MHZ_CAP) — RO. Hardwired to 0. |
| 4 | Capabilities List (CAP_LIST) — RO. 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. |
| 2:0 | Reserved. |

## 15.1.5 RID—Revision Identification Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Offset: | 08h | Attribute: | RO |
| Default Value: | See bit description | Size: | 8 Bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Revision ID — RO. Refer to the latest Intel® ICH5 / ICH5R Specification Update for the value of the Revision Identification register. |

## 15.1.6 PI—Programming Interface Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Offset: | 09h | Attribute: | RO |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Programming Interface — RO. |

## 15.1.7 SCC—Sub Class Code Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 0Ah | Attribute: | RO |
| Default Value: | 01h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Sub Class Code (SCC) — RO.<br>01h = Audio Device |

## 15.1.8 BCC—Base Class Code Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 0Bh | Attribute: | RO |
| Default Value: | 04h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Base Class Code (BCC) — RO.<br>04h = Multimedia device |

## 15.1.9 HEADTYP—Header Type Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 0Eh | Attribute: | RO |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Header Type — RO. Hardwired to 00h. |

## 15.1.10 NAMBAR—Native Audio Mixer Base Address Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 10–13h | Attribute: | R/W, RO |
| Default Value: | 00000001h | Size: | 32 bits |
| Lockable: | No | Power Well: | Core |

The Native PCI Mode Audio function uses PCI Base Address register #1 to request a contiguous block of I/O space that is to be used for the Native Audio Mixer software interface. The mixer requires 256 bytes of I/O space. Native Audio Mixer and Modem codec I/O registers are located from 00h to 7Fh and reside in the codec. Access to these registers will be decoded by the AC '97 controller and forwarded over the AC-link to the codec. The codec will then respond with the register value.

In the case of the split codec implementation, accesses to the different codecs are differentiated by the controller by using address offsets 00h–7Fh for the primary codec and address offsets 80h–FEh for the secondary codec.

***Note:*** The tertiary codec cannot be addressed via this address space. The tertiary space is only available from the new MMBAR register. This register powers up as read only and only becomes write-able when the IOSE bit in offset 41h is set.

For description of these I/O registers, refer to the *AC '97 v2.3 Specification*.

| Bit | Description |
|---|---|
| 31:16 | Hardwired to 0s. |
| 15:8 | **Base Address** — R/W. These bits are used in the I/O space decode of the Native Audio Mixer interface registers. The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. For the AC '97 mixer, the upper 16 bits are hardwired to 0, while bits 15:8 are programmable. This configuration yields a maximum I/O block size of 256 bytes for this base address. |
| 7:1 | Reserved. Read as 0s. |
| 0 | **Resource Type Indicator (RTE)** — RO. This bit defaults to 0 and changes to 1 if the IOSE bit is set (D31:F5:Offset 41h, bit 0). When 1, this bit indicates a request for I/O space. |

## 15.1.11 NABMBAR—Native Audio Bus Mastering Base Address Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 14–17h | Attribute: | R/W, RO |
| Default Value: | 00000001h | Size: | 32 bits |
| Lockable: | No | Power Well: | Core |

The Native PCI Mode Audio function uses PCI Base Address register #1 to request a contiguous block of I/O space that is to be used for the Native Mode Audio software interface.

***Note:*** The DMA registers for S/PDIF and Microphone In 2 cannot be addressed via this address space. These DMA functions are only available from the new MBBAR register. This register powers up as read only and only becomes write-able when the IOSE bit in offset 41h is set.

| Bit | Description |
|---|---|
| 31:16 | Hardwired to 0s |
| 15:6 | **Base Address** — R/W. These bits are used in the I/O space decode of the Native Audio Bus Mastering interface registers. The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. For AC '97 bus mastering, the upper 16 bits are hardwired to 0, while bits 15:6 are programmable. This configuration yields a maximum I/O block size of 64 bytes for this base address. |
| 5:1 | Reserved. Read as 0s. |
| 0 | **Resource Type Indicator (RTE)** — RO. This bit defaults to 0 and changes to 1 if the IOSE bit is set (D31:F5:Offset 41h, bit 0). When 1, this bit indicates a request for I/O space. |

## 15.1.12 MMBAR—Mixer Base Address Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 18–1Bh | Attribute: | R/W, RO |
| Default Value: | 00000000h | Size: | 32 bits |
| Lockable: | No | Power Well: | Core |

This BAR creates 512 bytes of memory space to signify the base address of the register space. The lower 256 bytes of this space map to the same registers as the 256-byte I/O space pointed to by NAMBAR. The lower 384 bytes are divided as follows:

- 128 bytes for the primary codec (offsets 00–7Fh)
- 128 bytes for the secondary codec (offsets 80–FFh)
- 128 bytes for the tertiary codec (offsets 100h–17Fh).
- 128 bytes of reserved space (offsets 180h–1FFh), returning all 0.

| Bit | Description |
|---|---|
| 31:9 | **Base Address** — R/W. This field provides the lower 32-bits of the 512-byte memory offset to use for decoding the primary, secondary, and tertiary codec's mixer spaces. |
| 8:3 | Reserved. Read as 0s. |
| 2:1 | Type — RO. Hardwired to 00b to Indicate the base address exists in 32-bit address space |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 0 to indicate a request for memory space. |

## 15.1.13 MBBAR—Bus Master Base Address Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 1C–1Fh | Attribute: | R/W, RO |
| Default Value: | 00000000h | Size: | 32 bits |
| Lockable: | No | Power Well: | Core |

This BAR creates 256-bytes of memory space to signify the base address of the bus master memory space. The lower 64-bytes of the space pointed to by this register point to the same registers as the MBBAR.

| Bit | Description |
|---|---|
| 31:8 | **Base Address** — R/W. This field provides the I/O offset to use for decoding the PCM In, PCM Out, and Microphone 1 DMA engines. |
| 7:3 | Reserved. Read as 0s. |
| 2:1 | Type — RO. Hardwired to 00b to indicate the base address exists in 32-bit address space |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 0 to indicate a request for memory space. |

## 15.1.14 SVID—Subsystem Vendor Identification Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 2C–2Dh | Attribute: | R/WO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

The SVID register, in combination with the Subsystem ID register, 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. |

## 15.1.15 SID—Subsystem Identification Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 2E–2Fh | Attribute: | R/WO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

The SID register, in combination with the Subsystem Vendor ID register 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. |

## 15.1.16 CAP_PTR—Capabilities Pointer Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 34h | Attribute: | RO |
| Default Value: | 50h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

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 |

## 15.1.17 INT_LN—Interrupt Line Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 3Ch | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

This register indicates which PCI interrupt line is used for the AC'97 module interrupt.

| Bit | Description |
|---|---|
| 7:0 | **Interrupt Line (INT_LN)** — R/W. This data is not used by the Intel® ICH5. It is used to communicate to software the interrupt line that the interrupt pin is connected to. |

## 15.1.18 INT_PN—Interrupt Pin Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 3Dh | Attribute: | RO |
| Default Value: | 02h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

This register indicates which PCI interrupt pin is used for the AC '97 module interrupt. The AC '97 interrupt is internally OR’d to the interrupt controller with the PIRQB# signal.

| Bit | Description |
|---|---|
| 7:3 | Reserved. |
| 2:0 | AC '97 Interrupt Routing — RO. Hardwired to 010b to select PIRQB#. |

## 15.1.19 PCID—Programmable Codec Identification Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 40h | Attribute: | R/W |
| Default Value: | 09h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

This register is used to specify the ID for the secondary and tertiary codecs for I/O accesses. This register is not affected by the $D3_{HOT}$ to D0 transition. The value in this register must be modified only before any AC ’97 codec accesses.

| Bit | Description |
|---|---|
| 7:4 | Reserved. |
| 3:2 | **Tertiary Codec ID (TID**) — R/W. These bits define the encoded ID that is used to address the tertiary codec I/O space. Bit 1 is the first bit sent and Bit 0 is the second bit sent on AC_SDOUT during slot 0. |
| 1:0 | **Secondary Codec ID (SCID)** — R/W. These two bits define the encoded ID that is used to address the secondary codec I/O space. The two bits are the ID that will be placed on slot 0, bits 0 and 1, upon an I/O access to the secondary codec. Bit 1 is the first bit sent and bit 0 is the second bit sent on AC_SDOUT during slot 0. |

## 15.1.20 CFG—Configuration Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 41h | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

This register is used to specify the ID for the secondary and tertiary codecs for I/O accesses. This register is not affected by the $D3_{HOT}$ to D0 transition.

| Bit | Description |
|---|---|
| 7:1 | Reserved—RO. |
| 0 | **I/O Space Enable (IOSE)** — R/W.<br>0 = Disable. The IOS bit at offset 04h and the I/O space BARs at offset 10h and 14h become read only registers. Additionally, bit 0 of the I/O BARs at offsets 10h and 14h are hardwired to 0 when this bit is 0. This is the default state for the I/O BARs. BIOS must explicitly set this bit to allow a legacy driver to work.<br>1 = Enable |

## 15.1.21 PID—PCI Power Management Capability Identification Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 50–51h | Attribute: | RO |
| Default Value: | 0001h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:8 | Next Capability (NEXT) — RO. This field indicates that the next item in the list is at offset 00h. |
| 7:0 | Capability ID (CAP) — RO.This field indicates that this pointer is a message signaled interrupt capability. |

## 15.1.22 PC—Power Management Capabilities Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 52–53h | Attribute: | RO |
| Default Value: | C9C2h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

This register is not affected by the $D3_{HOT}$ to D0 transition.

| Bit | Description |
|---|---|
| 15:11 | PME Support — RO. This field indicates PME# can be generated from all D states. |
| 10:9 | Reserved. |
| 8:6 | Auxiliary Current — RO. This field reports 375 mA maximum Suspend well current required when in the D3 cold state. |
| 5 | Device Specific Initialization (DSI)—RO. This field indicates that no device-specific initialization is required. |
| 4 | Reserved — RO. |
| 3 | PME Clock (PMEC) — RO. This field indicates that PCI clock is not required to generate PME#. |
| 2:0 | Version (VER) — RO. This field indicates support for the *PCI Power Management Specification, Revision 1.1* |

## 15.1.23 PCS—Power Management Control and Status Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| Address Offset: | 54–55h | Attribute: | R/W, R/WC |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Resume |

| Bit | Description |
|---|---|
| 15 | **PME Status (PMES)** — R/WC. This bit resides in the resume well. Software clears this bit by writing a 1 to it.<br>0 = PME# signal **not** asserted by AC '97 controller.<br>1 = This bit is set when the AC '97 controller would normally assert the PME# signal independent of the state of the PME_En bit. |
| 14:9 | Reserved — RO. |
| 8 | **Power Management Event Enable (PMEE)** — R/W.<br>0 = Disable<br>1 = Enable. When set, and if corresponding PMES is also set, the AC '97 controller sets the AC97_STS bit in the GPE0_STS register |
| 7:2 | Reserved—RO. |
| 1:0 | **Power State (PS)** — R/W. This field is used both to determine the current power state of the AC'97 controller and to set a new power state. The values are:<br>00 = D0 state<br>01 = not supported<br>10 = not supported<br>11 = $D3_{HOT}$ state<br>When in the $D3_{HOT}$ state, the AC '97 controller's configuration space is available, but the I/O and memory spaces are not. Additionally, interrupts are blocked.<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. |

## 15.2 AC ’97 Audio I/O Space (D31:F5)

The AC ’97 I/O space includes Native Audio Bus Master Registers and Native Mixer Registers. For the ICH5, the offsets are important as they will determine bits 1:0 of the TAG field (codec ID).

Audio Mixer I/O space can be accessed as a 16-bit field only since the data packet length on AC-link is a word. Any S/W access to the codec will be done as a 16-bit access starting from the first active byte. In case no byte enables are active, the access will be done at the first word of the qWord that contains the address of this request.

**Table 169. Intel® ICH5 Audio Mixer Register Configuration (Sheet 1 of 2)**

| Primary Offset (Codec ID =00) | Secondary Offset (Codec ID =01) | Tertiary Offset (Codec ID =10 | NAMBAR Exposed Registers (D31:F5) |
|---|---|---|---|
| 00h | 80h | 100h | Reset |
| 02h | 82h | 102h | Master Volume |
| 04h | 84h | 104h | Aux Out Volume |
| 06h | 86h | 106h | Mono Volume |
| 08h | 88h | 108h | Master Tone (R & L) |
| 0Ah | 8Ah | 10Ah | PC_BEEP Volume |
| 0Ch | 8Ch | 10Ch | Phone Volume |
| 0Eh | 8Eh | 10Eh | Mic Volume |
| 10h | 90h | 110h | Line In Volume |
| 12h | 92h | 112h | CD Volume |
| 14h | 94h | 114h | Video Volume |
| 16h | 96h | 116h | Aux In Volume |
| 18h | 98h | 118h | PCM Out Volume |
| 1Ah | 9Ah | 11Ah | Record Select |
| 1Ch | 9Ch | 11Ch | Record Gain |
| 1Eh | 9Eh | 11Eh | Record Gain Mic |
| 20h | A0h | 120h | General Purpose |
| 22h | A2h | 122h | 3D Control |
| 24h | A4h | 124h | AC’97 RESERVED |
| 26h | A6h | 126h | Powerdown Ctrl/Stat |
| 28h | A8h | 128h | Extended Audio |
| 2Ah | AAh | 12Ah | Extended Audio Ctrl/Stat |
| 2Ch | ACh | 12Ch | PCM Front DAC Rate |
| 2Eh | AEh | 12Eh | PCM Surround DAC Rate |
| 30h | B0h | 130h | PCM LFE DAC Rate |
| 32h | B2h | 132h | PCM LR ADC Rate |
| 34h | B4h | 134h | MIC ADC Rate |
| 36h | B6h | 136h | 6Ch Vol: C, LFE |
| 38h | B8h | 138h | 6Ch Vol: L, R Surround |
| 3Ah | BAh | 13Ah | S/PDIF Control |
| 3C–56h | BC–D6h | 13C–156h | Intel RESERVED |
| 58h | D8h | 158h | AC’97 Reserved |

### Table 169. Intel® ICH5 Audio Mixer Register Configuration (Sheet 2 of 2)

| Primary Offset (Codec ID =00) | Secondary Offset (Codec ID =01) | Tertiary Offset (Codec ID =10 | NAMBAR Exposed Registers (D31:F5) |
|---|---|---|---|
| 5Ah | DAh | 15Ah | Vendor Reserved |
| 7Ch | FCh | 17Ch | Vendor ID1 |
| 7Eh | FEh | 17Eh | Vendor ID2 |

**NOTE:**
1. Software should not try to access reserved registers
2. Primary Codec ID cannot be changed. Secondary codec ID can be changed via bits 1:0 of configuration register 40h. Tertiary codec ID can be changed via bits 3:2 of configuration register 40h.
3. The tertiary offset is only available through the memory space defined by the MMBAR register.

The Bus Master registers are located from offset + 00h to offset + 51h and reside in the AC '97 controller. Accesses to these registers do **not** cause the cycle to be forwarded over the AC-link to the codec. S/W could access these registers as bytes, word, DWord or qWord quantities, but reads must not cross DWord boundaries.

In the case of the split codec implementation accesses to the different codecs are differentiated by the controller by using address offsets 00h–7Fh for the primary codec, address offsets 80h–FFh for the secondary codec and address offsets 100h–17Fh for the tertiary codec.

The Global Control (GLOB_CNT) and Global Status (GLOB_STA) registers are aliased to the same global registers in the audio and modem I/O space. Therefore a read/write to these registers in either audio or modem I/O space affects the same physical register.

Bus Mastering registers exist in I/O space and reside in the AC '97 controller. The six channels, PCM in, PCM in 2, PCM out, Mic in, Mic 2, and S/PDIF out, each have their own set of Bus Mastering registers. The following register descriptions apply to all six channels. The register definition section titles use a generic "x_" in front of the register to indicate that the register applies to all six channels. The naming prefix convention used in Table 170 and in the register description I/O address is as follows:

PI = PCM in channel
PO = PCM out channel
MC = Mic in channel
MC2 = Mic 2 channel
PI2 = PCM in 2 channel
SP = S/PDIF out channel.

### Table 170. Native Audio Bus Master Control Registers (Sheet 1 of 2)

| Offset | Mnemonic | Name | Default | Access |
|---|---|---|---|---|
| 00h | PI_BDBAR | PCM In Buffer Descriptor list Base Address | 00000000h | R/W |
| 04h | PI_CIV | PCM In Current Index Value | 00h | RO |
| 05h | PI_LVI | PCM In Last Valid Index | 00h | R/W |
| 06h | PI_SR | PCM In Status | 0001h | R/WC, RO |
| 08h | PI_PICB | PCM In Position in Current Buffer | 0000h | RO |
| 0Ah | PI_PIV | PCM In Prefetched Index Value | 00h | RO |
| 0Bh | PI_CR | PCM In Control | 00h | R/W, R/W (special) |
| 10h | PO_BDBAR | PCM Out Buffer Descriptor list Base Address | 00000000h | R/W |
| 14h | PO_CIV | PCM Out Current Index Value | 00h | RO |
| 15h | PO_LVI | PCM Out Last Valid Index | 00h | R/W |
| 16h | PO_SR | PCM Out Status | 0001h | R/WC, RO |
| 18h | PO_PICB | PCM In Position In Current Buffer | 0000h | RO |
| 1Ah | PO_PIV | PCM Out Prefetched Index Value | 00h | RO |
| 1Bh | PO_CR | PCM Out Control | 00h | R/W, R/W (special) |
| 20h | MC_BDBAR | Mic. In Buffer Descriptor List Base Address | 00000000h | R/W |
| 24h | MC_CIV | Mic. In Current Index Value | 00h | RO |
| 25h | MC_LVI | Mic. In Last Valid Index | 00h | R/W |
| 26h | MC_SR | Mic. In Status | 0001h | R/WC, RO |
| 28h | MC_PICB | Mic. In Position In Current Buffer | 0000h | RO |
| 2Ah | MC_PIV | Mic. In Prefetched Index Value | 00h | RO |
| 2Bh | MC_CR | Mic. In Control | 00h | R/W, R/W (special) |
| 2Ch | GLOB_CNT | Global Control | 00000000h | R/W, R/W (special) |
| 30h | GLOB_STA | Global Status | 00700000h | R/W, R/WC, RO |
| 34h | CAS | Codec Access Semaphore | 00h | R/W (special) |
| 40h | MC2_BDBAR | Mic. 2 Buffer Descriptor List Base Address | 00000000h | R/W |
| 44h | MC2_CIV | Mic. 2 Current Index Value | 00h | RO |
| 45h | MC2_LVI | Mic. 2 Last Valid Index | 00h | R/W |
| 46h | MC2_SR | Mic. 2 Status | 0001h | RO, RWC |
| 48h | MC2_PICB | Mic 2 Position In Current Buffer | 0000h | RO |
| 4Ah | MC2_PIV | Mic. 2 Prefetched Index Value | 00h | R/W |
| 4Bh | MC2_CR | Mic. 2 Control | 00h | R/W, R/W (special) |
| 50h | PI2_BDBAR | PCM In 2 Buffer Descriptor List Base Address | 00000000h | R/W |
| 54h | PI2_CIV | PCM In 2 Current Index Value | 00h | RO |
| 55h | PI2_LVI | PCM In 2 Last Valid Index | 00h | R/W |
| 56h | PI2_SR | PCM In 2 Status | 0001h | R/WC, RO |
| 58h | PI2_PICB | PCM In 2 Position in Current Buffer | 0000h | RO |
| 5Ah | PI2_PIV | PCM In 2 Prefetched Index Value | 00h | RO |
| 5Bh | PI2_CR | PCM In 2 Control | 00h | R/W, R/W (special) |
| 60h | SP_BAR | S/PDIF Buffer Descriptor List Base Address | 00000000h | R/W |

**Table 170. Native Audio Bus Master Control Registers (Sheet 2 of 2)**

| Offset | Mnemonic | Name | Default | Access |
|---|---|---|---|---|
| 64h | SP_CIV | S/PDIF Current Index Value | 00h | RO |
| 65h | SP_LVI | S/PDIF Last Valid Index | 00h | R/W |
| 66h | SP_SR | S/PDIF Status | 0001h | R/WC, RO |
| 68h | SP_PICB | S/PDIF Position In Current Buffer | 0000h | RO |
| 6Ah | SP_PIV | S/PDIF Prefetched Index Value | 00h | RO |
| 6Bh | SP_CR | S/PDIF Control | 00h | R/W, R/W (special) |
| 80h | SDM | SData_IN Map | 00h | R/W, RO |

***Note:*** Internal reset as a result of $D3_{HOT}$ to D0 transition will reset all the core well registers except the registers shared with the AC '97 Modem (GCR, GSR, CASR). All resume well registers will not be reset by the $D3_{HOT}$ to D0 transition.

Core well registers and bits **not** reset by the $D3_{HOT}$ to D0 transition:

- offset 2Ch–2Fh – bits 6:0 Global Control (GLOB_CNT)
- offset 30h–33h – bits [29,15,11:10,0] Global Status (GLOB_STA)
- offset 34h – Codec Access Semaphore Register (CAS)

Resume well registers and bits **will not** be reset by the $D3_{HOT}$ to D0 transition:

- offset 30h–33h – bits [17:16] Global Status (GLOB_STA)

## 15.2.1 *x*_BDBAR—Buffer Descriptor Base Address Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 00h (PIBDBAR),<br>NABMBAR + 10h (POBDBAR),<br>NABMBAR + 20h (MCBDBAR)<br>MBBAR + 40h (MC2BDBAR)<br>MBBAR + 50h (PI2BDBAR)<br>MBBAR + 60h (SPBAR) | Attribute: | R/W |
| Default Value: | 00000000h | Size: | 32 bits |
| Lockable: | No | Power Well: | Core |

Software can read the register at offset 00h by performing a single 32-bit read from address offset 00h. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 31:3 | **Buffer Descriptor Base Address[31:3]** — R/W. These bits represent address bits 31:3. The data should be aligned on 8-byte boundaries. Each buffer descriptor is 8 bytes long and the list can contain a maximum of 32 entries. |
| 2:0 | Hardwired to 0. |

## 15.2.2 x_CIV—Current Index Value Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 04h (PICIV),<br>NABMBAR + 14h (POCIV),<br>NABMBAR + 24h (MCCIV)<br>MBBAR + 44h (MC2CIV)<br>MBBAR + 54h (PI2CIV)<br>MBBAR + 64h (SPCIV) | Attribute: | RO |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 8-bit read to offset 04h.

| Bit | Description |
|---|---|
| 7:5 | Hardwired to 0 |
| 4:0 | **Current Index Value [4:0]** — RO. These bits represent which buffer descriptor within the list of 32 descriptors is currently being processed. As each descriptor is processed, this value is incremented. The value rolls over after it reaches 31. |

**NOTE:** Reads across DWord boundaries are not supported.

## 15.2.3 x_LVI—Last Valid Index Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 05h (PILVI),<br>NABMBAR + 15h (POLVI),<br>NABMBAR + 25h (MCLVI)<br>MBBAR + 45h (MC2LVI)<br>MBBAR + 55h (PI2LVI)<br>MBBAR + 65h (SPLVI) | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 8-bit read to offset 05h.

| Bit | Description |
|---|---|
| 7:5 | Hardwired to 0. |
| 4:0 | **Last Valid Index [4:0]** — R/W. This value represents the last valid descriptor in the list. This value is updated by the software each time it prepares a new buffer and adds it to the list. |

**NOTE:** Reads across DWord boundaries are not supported.

## 15.2.4 x_SR—Status Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 06h (PISR),<br>NABMBAR + 16h (POSR),<br>NABMBAR + 26h (MCSR)<br>MBBAR + 46h (MC2SR)<br>MBBAR + 56h (PI2SR)<br>MBBAR + 66h (SPSR) | Attribute: | R/WC, RO |
| Default Value: | 0001h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 16-bit read to offset 06h. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 15:5 | Reserved. |
| 4 | **FIFO Error (FIFOE)** — R/WC. Software clears this bit by writing a 1 to it.<br>0 = No FIFO error.<br>1 = FIFO error occurs.<br>**PISR Register:** FIFO error indicates a FIFO overrun. The FIFO pointers don't increment, the incoming data is not written into the FIFO, thus is lost.<br>**POSR Register:** FIFO error indicates a FIFO underrun. The sample transmitted in this case should be the last valid sample.<br>The Intel® ICH5 will set the FIFOE bit if the under-run or overrun occurs when there are more valid buffers to process. |
| 3 | **Buffer Completion Interrupt Status (BCIS)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by the hardware after the last sample of a buffer has been processed, AND if the Interrupt on Completion (IOC) bit is set in the command byte of the buffer descriptor. It remains active until cleared by software. |
| 2 | **Last Valid Buffer Completion Interrupt (LVBCI)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Last valid buffer has been processed. It remains active until cleared by software. This bit indicates the occurrence of the event signified by the last valid buffer being processed. Thus this is an event status bit that can be cleared by software once this event has been recognized. This event will cause an interrupt if the enable bit in the Control Register is set. The interrupt is cleared when the software clears this bit.<br><br>In the case of *Transmits* (PCM out, Modem out) this bit is set, after the last valid buffer has been fetched (not after transmitting it). While in the case of *Receives*, this bit is set after the data for the last buffer has been written to memory. |
| 1 | **Current Equals Last Valid (CELV)** — RO.<br>0 = Cleared by hardware when controller exists state (i.e., until a new value is written to the LVI register.)<br>1 = Current Index is equal to the value in the Last Valid Index Register, and the buffer pointed to by the CIV has been processed (i.e., after the last valid buffer has been processed). This bit is very similar to bit 2, except this bit reflects the state rather than the event. This bit reflects the state of the controller, and remains set until the controller exits this state. |
| 0 | **DMA Controller Halted (DCH)** — RO.<br>0 = Running.<br>1 = Halted. This could happen because of the Start/Stop bit being cleared and the DMA engines are idle, or it could happen once the controller has processed the last valid buffer. |

## 15.2.5 x_PICB—Position In Current Buffer Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 08h (PIPICB),<br>NABMBAR + 18h (POPICB),<br>NABMBAR + 28h (MCPICB)<br>MBBAR + 48h (MC2PICB)<br>MBBAR + 58h (PI2PICB)<br>MBBAR + 68h (SPPICB) | Attribute: | RO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 16-bit read to offset 08h. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 15:0 | **Position In Current Buffer [15:0]** — RO. These bits represent the number of samples left to be processed in the current buffer. Once again, this means, the number of samples not yet read from memory (in the case of reads from memory) or not yet written to memory (in the case of writes to memory), irrespective of the number of samples that have been transmitted/received across AC-link. |

## 15.2.6 x_PIV—Prefetched Index Value Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 0Ah (PIPIV),<br>NABMBAR + 1Ah (POPIV),<br>NABMBAR + 2Ah (MCPIV)<br>MBBAR + 4Ah (MC2PIV)<br>MBBAR + 5Ah (PI2PIV)<br>MBBAR + 6Ah (SPPIV) | Attribute: | RO |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 8-bit read to offset 0Ah. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 7:5 | Hardwired to 0. |
| 4:0 | **Prefetched Index Value [4:0]** — RO. These bits represent which buffer descriptor in the list has been prefetched. The bits in this register are also modulo 32 and roll over after they reach 31. |

## 15.2.7 x_CR—Control Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 0Bh (PICR),<br>NABMBAR + 1Bh (POCR),<br>NABMBAR + 2Bh (MCCR)<br>MBBAR + 4Bh (MC2CR)<br>MBBAR + 5Bh (PI2CR)<br>MBBAR + 6Bh (SPCR) | Attribute: | R/W, R/W (special) |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 8-bit read to offset 0Bh. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 7:5 | Reserved. |
| 4 | **Interrupt on Completion Enable (IOCE)** — R/W. This bit controls whether or not an interrupt occurs when a buffer completes with the IOC bit set in its descriptor.<br>0 = Disable. Interrupt will not occur.<br>1 = Enable |
| 3 | **FIFO Error Interrupt Enable (FEIE)** — R/W. This bit controls whether the occurrence of a FIFO error will cause an interrupt or not.<br>0 = Disable. Bit 4 in the Status Register will be set, but the interrupt will not occur.<br>1 = Enable. Interrupt will occur. |
| 2 | **Last Valid Buffer Interrupt Enable (LVBIE)** — R/W. This bit controls whether the completion of the last valid buffer will cause an interrupt or not.<br>0 = Disable. Bit 2 in the Status register will still be set, but the interrupt will not occur.<br>1 = Enable |
| 1 | **Reset Registers (RR)** — R/W (special).<br>0 = Removes reset condition.<br>1 = Contents of all Bus master related registers to be reset, except the interrupt enable bits (bit 4,3,2 of this register). Software needs to set this bit but need not clear it since the bit is self clearing. This bit must be set only when the Run/Pause bit is cleared. Setting it when the Run bit is set will cause undefined consequences. |
| 0 | **Run/Pause Bus Master (RPBM)** — R/W.<br>0 = Pause bus master operation. This results in all state information being retained (i.e., master mode operation can be stopped and then resumed).<br>1 = Run. Bus master operation starts. |

## 15.2.8 GLOB_CNT—Global Control Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 2Ch | Attribute: | R/W, R/W (special) |
| Default Value: | 00000000h | Size: | 32 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 31:30 | **S/PDIF Slot Map (SSM)** — R/W. If the run/pause bus master bit (bit 0 of offset 2Bh) is set, then the value in these bits indicate which slots S/PDIF data is transmitted on. Software must ensure that the programming here does not conflict with the PCM channels being used. If there is a conflict, unpredictable behavior will result — the hardware will not check for a conflict.<br>00 = Reserved<br>01 = Slots 7 and 8<br>10 = Slots 6 and 9<br>11 = Slots 10 and 11 |
| 29:24 | Reserved. |
| 23:22 | **PCM Out Mode (POM)** — R/W. Enables the PCM out channel to use 16 or 20-bit audio on PCM out. This does not affect the microphone of S/PDIF DMA. When greater than 16 bit audio is used, the data structures are aligned as 32-bits per sample, with the highest order bits representing the data, and the lower order bits as don't care.<br>00 = 16 bit audio (default)<br>01 = 20 bit audio<br>10 = Reserved. If set, indeterminate behavior will result.<br>11 = Reserved. If set, indeterminate behavior will result. |
| 21:20 | **PCM 4/6 Enable** — R/W. This field configures PCM Output for 2, 4 or 6 channel mode.<br>00 = 2-channel mode (default)<br>01 = 4-channel mode<br>10 = 6-channel mode<br>11 = Reserved |
| 19:7 | Reserved. |
| 6 | **AC_SDIN2 Interrupt Enable** — R/W.<br>0 = Disable<br>1 = Enable an interrupt to occur when the codec on the AC_SDIN2 causes a resume event on the AC-link. |
| 5 | **AC_SDIN1 Interrupt Enable** — R/W.<br>0 = Disable<br>1 = Enable an interrupt to occur when the codec on the AC_SDIN1 causes a resume event on the AC-link. |
| 4 | **AC_SDIN0 Interrupt Enable** — R/W.<br>0 = Disable<br>1 = Enable an interrupt to occur when the codec on AC_SDIN0 causes a resume event on the AC-link. |
| 3 | **ACLINK Shut Off (LSO)** — R/W.<br>0 = Normal operation.<br>1 = Controller disables all outputs which will be pulled low by internal pull down resistors. |

| Bit | Description |
|---|---|
| 2 | **AC ’97 Warm Reset** — R/W (special).<br>0 = Normal operation.<br>1 = Writing a 1 to this bit causes a warm reset to occur on the AC-link. The warm reset will awaken a suspended codec without clearing its internal registers. If software attempts to perform a warm reset while AC_BIT_CLK is running, the write will be ignored and the bit will not change. This bit is self-clearing (it remains set until the reset completes and AC_BIT_CLK is seen on the AC-link, after which it clears itself). |
| 1 | **AC ’97 Cold Reset#** — R/W.<br>0 = Writing a 0 to this bit causes a cold reset to occur throughout the AC ‘97 circuitry. All data in the controller and the codec will be lost. Software needs to clear this bit no sooner than the minimum number of ms have elapsed.<br>1 = This bit defaults to 0 and hence after reset, the driver needs to set this bit to a 1. The value of this bit is retained after suspends; hence, if this bit is set to a 1 prior to suspending, a cold reset is not generated automatically upon resuming.<br>Note: This bit is in the Core well. |
| 0 | **GPI Interrupt Enable (GIE)** — R/W. This bit controls whether the change in status of any GPI causes an interrupt.<br>0 = Bit 0 of the Global Status Register is set, but no interrupt is generated.<br>1 = The change on value of a GPI causes an interrupt and sets bit 0 of the Global Status Register. |

**NOTE:** Reads across DWord boundaries are not supported.

## 15.2.9 GLOB_STA—Global Status Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 30h | Attribute: | RO, R/W, R/WC |
| Default Value: | 00700000h | Size: | 32 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 31:30 | Reserved. |
| 29 | **AC_SDIN2 Resume Interrupt (S2RI)** — R/WC. This bit indicates a resume event occurred on AC_SDIN2. Software clears this bit by writing a 1 to it.<br>0 = Resume event did **not** occur.<br>1 = Resume event occurred.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |
| 28 | **AC_SDIN2 Codec Ready (S2CR)** — RO. Reflects the state of the codec ready bit on AC_SDIN2. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously.<br>0 = Not Ready.<br>1 = Ready. |
| 27 | **Bit Clock Stopped (BCS)** — RO. This bit indicates that the bit clock is not running.<br>0 = Transition is found on AC_BIT_CLK.<br>1 = Intel® ICH5 detected that there has been no transition on AC_BIT_CLK for four consecutive PCI clocks. |
| 26 | **S/PDIF Interrupt (SPINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = S/PDIF out channel interrupt status bits have been set. |
| 25 | **PCM In 2 Interrupt (P2INT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the PCM In 2 channel status bits have been set. |
| 24 | **Microphone 2 In Interrupt (M2INT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the Mic in channel interrupts status bits has been set. |
| 23:22 | **Sample Capabilities** — RO. This field indicates the capability to support more greater than 16-bit audio.<br>00 = Reserved<br>01 = 16 and 20-bit Audio supported<br>10 = Reserved<br>11 = Reserved |
| 21:20 | **Multichannel Capabilities**— RO. This field indicates the capability to support more 4 and 6 channels on PCM Out. |
| 19:18 | Reserved. |
| 17 | **MD3** — R/W. Power down semaphore for Modem. This bit exists in the suspend well and maintains context across power states (except G3). The bit has no hardware function. It is used by software in conjunction with the AD3 bit to coordinate the entry of the two codecs into D3 state.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |
| 16 | **AD3** — R/W. Power down semaphore for Audio. This bit exists in the suspend well and maintains context across power states (except G3). The bit has no hardware function. It is used by software in conjunction with the MD3 bit to coordinate the entry of the two codecs into D3 state.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |

| Bit | Description |
|---|---|
| 15 | **Read Completion Status (RCS)** — R/WC. This bit indicates the status of codec read completions.<br>0 = A codec read completes normally.<br>1 = A codec read results in a time-out. The bit remains set until being cleared by software writing a 1 to the bit location.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |
| 14 | **Bit 3 of Slot 12** — RO. Display bit 3 of the most recent slot 12. |
| 13 | **Bit 2 of Slot 12** — RO. Display bit 2 of the most recent slot 12. |
| 12 | **Bit 1 of slot 12** — RO. Display bit 1 of the most recent slot 12. |
| 11 | **AC_SDIN1 Resume Interrupt (S1R1)** — R/WC. This bit indicates that a resume event occurred on AC_SDIN1. Software clears this bit by writing a 1 to it.<br>0 = Resume event did **not** occur<br>1 = Resume event occurred.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |
| 10 | **AC_SDIN0 Resume Interrupt (S0R1)** — R/WC. This bit indicates that a resume event occurred on AC_SDIN0. Software clears this bit by writing a 1 to it.<br>0 = Resume event did **not** occur<br>1 = Resume event occurred.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |
| 9 | **AC_SDIN1 Codec Ready (S1CR)** — RO. Reflects the state of the codec ready bit in AC_SDIN1. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously.<br>0 = Not Ready.<br>1 = Ready. |
| 8 | **AC_SDIN0 Codec Ready (S0CR)** — RO. Reflects the state of the codec ready bit in AC_SDIN 0. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously.<br>0 = Not Ready.<br>1 = Ready. |
| 7 | **Microphone In Interrupt (MINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the Mic in channel interrupts status bits has been set. |
| 6 | **PCM Out Interrupt (POINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the PCM out channel interrupts status bits has been set. |
| 5 | **PCM In Interrupt (PIINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the PCM in channel interrupts status bits has been set. |
| 4:3 | Reserved |
| 2 | **Modem Out Interrupt (MOINT**) — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the modem out channel interrupts status bits has been set. |
| 1 | **Modem In Interrupt (MIINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the modem in channel interrupts status bits has been set. |
| 0 | **GPI Status Change Interrupt (GSCI)** — RWC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit reflects the state of bit 0 in slot 12, and is set when bit 0 of slot 12 is set. This indicates that one of the GPIs changed state, and that the new values are available in slot 12.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |

**NOTE:** Reads across DWord boundaries are not supported.

## 15.2.10 CAS—Codec Access Semaphore Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 34h | Attribute: | R/W (special) |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:1 | Reserved. |
| 0 | **Codec Access Semaphore (CAS)** — R/W (special). This bit is read by software to check whether a codec access is currently in progress.<br>0 = No access in progress.<br>1 = The act of reading this register sets this bit to 1. The driver that read this bit can then perform an I/O access. Once the access is completed, hardware automatically clears this bit. |

**NOTE:** Reads across DWord boundaries are not supported.

## 15.2.11 SDM—SDATA_IN Map Register (Audio—D31:F5)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 80h | Attribute: | R/W, RO |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:6 | **PCM In 2, Microphone In 2 Data In Line (DI2L)**— R/W. When the SE bit is set, these bits indicates which AC_SDIN line should be used by the hardware for decoding the input slots for PCM In 2 and Microphone In 2. When the SE bit is cleared, the value of these bits are irrelevant, and PCM In 2 and Mic In 2 DMA engines are not available.<br>00 = AC_SDIN0<br>01 = AC_SDIN1<br>10 = AC_SDIN2<br>11 = Reserved |
| 5:4 | **PCM In 1, Microphone In 1 Data In Line (DI1L)**— R/W. When the SE bit is set, these bits indicates which AC_SDIN line should be used by the hardware for decoding the input slots for PCM In 1 and Microphone In 1. When the SE bit is cleared, the value of these bits are irrelevant, and the PCM In 1 and Mic In 1 engines use the OR'd AC_SDIN lines.<br>00 = AC_SDIN0<br>01 = AC_SDIN1<br>10 = AC_SDIN2<br>11 = Reserved |
| 3 | **Steer Enable (SE)** — R/W. When set, the AC_SDIN lines are treated separately and not OR'd together before being sent to the DMA engines. When cleared, the AC_SDIN lines are OR'd together, and the "Microphone In 2" and "PCM In 2" DMA engines are not available. |
| 2 | Reserved — RO. |
| 1:0 | **Last Codec Read Data Input (LDI)** — RO. When a codec register is read, this indicates which AC_SDIN the read data returned on. Software can use this to determine how the codecs are mapped. The values are:<br>00 = AC_SDIN0<br>01 = AC_SDIN1<br>10 = AC_SDIN2<br>11 = Reserved |

**NOTE:** Reads across DWord boundaries are not supported.

# AC '97 Modem Controller Registers (D31:F6) 16

## 16.1 AC '97 Modem PCI Configuration Space (D31:F6)

*Note:* Address locations that are not shown in Table 171 should be treated as Reserved (see Section 6.2 for details).

**Table 171. AC '97 Modem PCI Register Address Map (Modem—D31:F6)**

| Offset | Mnemonic | Register | Default | Access |
|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086 | RO |
| 02–03h | DID | Device Identification | 24D6 | RO |
| 04–05h | PCICMD | PCI Command | 0000h | R/W, RO |
| 06–07h | PCISTS | PCI Status | 0290h | 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 | 07h | RO |
| 0Eh | HEADTYP | Header Type | 00h | RO |
| 10–13h | MMBAR | Modem Mixer Base Address | 00000001h | R/W, RO |
| 14–17h | MBAR | Modem Base Address | 00000001h | R/W, RO |
| 2C–2Dh | SVID | Subsystem Vendor Identification | 0000h | R/WO |
| 2E–2Fh | SID | Subsystem Identification | 0000h | R/WO |
| 34h | CAP_PTR | Capabilities Pointer | 50h | RO |
| 3Ch | INT_LN | Interrupt Line | 00h | R/W |
| 3Dh | INT_PN | Interrupt Pin | 02h | RO |
| 50–51h | PID | PCI Power Management Capability ID | 0001h | RO |
| 52–53h | PC | Power Management Capabilities | C9C2h | RO |
| 54–55h | PCS | Power Management Control and Status | 0000h | R/W, R/WC |

*Note:* Internal reset as a result of $D3_{HOT}$ to D0 transition will reset all the core well registers except the following BIOS programmed registers as BIOS may not be invoked following the D3-to-D0 transition. All resume well registers will not be reset by the $D3_{HOT}$ to D0 transition.

Core well registers **not** reset by the $D3_{HOT}$ to D0 transition:

- offset 2Ch–2Dh – Subsystem Vendor ID (SVID)
- offset 2Eh–2Fh – Subsystem ID (SID)

Resume well registers **will not** be reset by the $D3_{HOT}$ to D0 transition:

- offset 54h–55h – Power Management Control and Status (PCS)

## 16.1.1 VID—Vendor Identification Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 00–01h | Attribute: | RO |
| Default Value: | 8086 | Size: | 16 Bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | Vendor ID. |

## 16.1.2 DID—Device Identification Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 02–03h | Attribute: | RO |
| Default Value: | 24D6h | Size: | 16 Bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:0 | Device ID. |

## 16.1.3 PCICMD—PCI Command Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 04–05h | Attribute: | R/W, RO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

PCICMD is a 16-bit control register. Refer to the *PCI Local Bus Specification, Revision 2.3* for complete details on each bit.

| Bit | Description |
|---|---|
| 15:11 | Reserved. Read 0. |
| 10 | **Interrupt Disable (ID)**— R/W.<br>0 = The INTx# signals may be asserted and MSIs may be generated.<br>1 = The AC '97 controller's INTx# signal will be de-asserted and it may not generate MSIs. |
| 9 | Fast Back to Back Enable (FBE) — RO. Not implemented. Hardwired to 0. |
| 8 | SERR# Enable (SERR_EN) — RO. Not implemented. Hardwired to 0. |
| 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) — RO. Not implemented. Hardwired to 0. |
| 4 | Memory Write and Invalidate Enable (MWIE) — RO. Not implemented. Hardwired to 0. |
| 3 | Special Cycle Enable (SCE) — RO. Not implemented. Hardwired to 0. |
| 2 | **Bus Master Enable (BME)** — R/W. This bit controls standard PCI bus mastering capabilities.<br>0 = Disable<br>1 = Enable |
| 1 | Memory Space Enable (MSE) — RO. Hardwired to 0, AC '97 does not respond to memory accesses. |
| 0 | **I/O Space Enable (IOSE)** — R/W. This bit controls access to the I/O space registers.<br>0 = Disable access. (default = 0).<br>1 = Enable access to I/O space. The Native PCI Mode Base Address register should be programmed prior to setting this bit. |

## 16.1.4 PCISTS—PCI Status Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 06–07h | Attribute: | R/WC, RO |
| Default Value: | 0290h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

PCISTA is a 16-bit status register. Refer to the *PCI Local Bus Specification, Revision 2.3* for complete details on each bit.

***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. Not implemented. Hardwired to 0. |
| 14 | Signaled System Error (SSE) —RO. Not implemented. Hardwired to 0. |
| 13 | **Master Abort Status (MAS)** — R/WC.<br>0 = Master abort **not** generated by bus master AC '97 function.<br>1 = Bus Master AC '97 interface function, as a master, generates a master abort. |
| 12 | Reserved. Read as 0. |
| 11 | Signaled Target Abort (STA) — RO. Not implemented. Hardwired to 0. |
| 10:9 | DEVSEL# Timing Status (DEV_STS) — RO. This 2-bit field reflects the Intel® ICH5's DEVSEL# timing parameter. These read only bits indicate the ICH5's DEVSEL# timing when performing a positive decode. |
| 8 | Data Parity Error Detected (DPED) — RO. Not implemented. Hardwired to 0. |
| 7 | Fast Back to Back Capable (FB2BC) — RO. Hardwired to 1. This bit indicates that the ICH5 as a target is capable of fast back-to-back transactions. |
| 6 | User Definable Features (UDF) — RO. Not implemented. Hardwired to 0. |
| 5 | 66 MHz Capable (66MHZ_CAP) — RO. Hardwired to 0. |
| 4 | Capabilities List (CAP_LIST) — RO. Indicates that the controller contains a capabilities pointer list. The first item is pointed to by looking at configuration offset 34h. |
| 3 | **Interrupt Status (INTS)** — RO.<br>0 = This bit is 0 after the interrupt is cleared.<br>1 = This bit is 1 when the INTx# is asserted. |
| 2:0 | Reserved |

## 16.1.5 RID—Revision Identification Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 08h | Attribute: | RO |
| Default Value: | See bit description | Size: | 8 Bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Revision ID — RO. Refer to the latest Intel® ICH5 / ICH5R Specification Update for the value of the Revision Identification register. |

## 16.1.6 PI—Programming Interface Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 09h | Attribute: | RO |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Programming Interface — RO. |

## 16.1.7 SCC—Sub Class Code Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 0Ah | Attribute: | RO |
| Default Value: | 03h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Sub Class Code — RO.<br>03h = Generic Modem. |

## 16.1.8 BCC—Base Class Code Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 0Bh | Attribute: | RO |
| Default Value: | 07h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Base Class Code — RO.<br>07h = Simple Communications Controller. |

## 16.1.9 HEADTYP—Header Type Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 0Eh | Attribute: | RO |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:0 | Header Type — RO. |

## 16.1.10 MMBAR—Modem Mixer Base Address Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 10–13h | Attribute: | R/W, RO |
| Default Value: | 00000001h | Size: | 32 bits |

The Native PCI Mode Modem uses PCI Base Address register #1 to request a contiguous block of I/O space that is to be used for the Modem Mixer software interface. The mixer requires 256 bytes of I/O space. All accesses to the mixer registers are forwarded over the AC-link to the codec where the registers reside.

In the case of the split codec implementation accesses to the different codecs are differentiated by the controller by using address offsets 00h–7Fh for the primary codec and address offsets 80h–FEh for the secondary codec.

| Bit | Description |
|---|---|
| 31:16 | Hardwired to 0s. |
| 15:8 | **Base Address** — R/W. These bits are used in the I/O space decode of the Modem interface registers. The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. For the AC '97 Modem, the upper 16 bits are hardwired to 0, while bits 15:8 are programmable. This configuration yields a maximum I/O block size of 256 bytes for this base address. |
| 7:1 | Reserved. Read as 0. |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1indicating a request for I/O space. |

## 16.1.11 MBAR—Modem Base Address Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 14–17h | Attribute: | R/W, RO |
| Default Value: | 00000001h | Size: | 32 bits |

The Modem function uses PCI Base Address register #1 to request a contiguous block of I/O space that is to be used for the Modem software interface. The Modem Bus Mastering register space requires 128 bytes of I/O space. All Modem registers reside in the controller, therefore cycles are **not** forwarded over the AC-link to the codec.

| Bit | Description |
|---|---|
| 31:16 | Hardwired to 0s. |
| 15:7 | **Base Address** — R/W. These bits are used in the I/O space decode of the Modem interface registers. The number of upper bits that a device actually implements depends on how much of the address space the device will respond to. For the AC '97 Modem, the upper 16 bits are hardwired to 0, while bits 15:7 are programmable. This configuration yields a maximum I/O block size of 128 bytes for this base address. |
| 6:1 | Reserved. Read as 0. |
| 0 | Resource Type Indicator (RTE) — RO. Hardwired to 1 indicating a request for I/O space. |

## 16.1.12 SVID—Subsystem Vendor Identification Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 2C–2Dh | Attribute: | R/WO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

The SVID register, in combination with the Subsystem ID register, 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. |

## 16.1.13 SID—Subsystem Identification Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 2E–2Fh | Attribute: | R/WO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

The SID register, in combination with the Subsystem Vendor ID register make it possible for the operating environment to distinguish one audio subsystem from another. 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. |

## 16.1.14 CAP_PTR—Capabilities Pointer Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 34h | Attribute: | RO |
| Default Value: | 50h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

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. |

## 16.1.15 INT_LN—Interrupt Line Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 3Ch | Attribute: | R/W |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

This register indicates which PCI interrupt line is used for the AC ’97 module interrupt.

| Bit | Description |
|---|---|
| 7:0 | Interrupt Line (INT_LN) — R/W. This data is not used by the Intel® ICH5. It is used to communicate to software the interrupt line that the interrupt pin is connected to. |

## 16.1.16 INT_PIN—Interrupt Pin Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 3Dh | Attribute: | RO |
| Default Value: | 02h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

This register indicates which PCI interrupt pin is used for the AC ’97 modem interrupt. The AC ’97 interrupt is internally OR’d to the interrupt controller with the PIRQB# signal.

| Bit | Description |
|---|---|
| 7:3 | Reserved |
| 2:0 | Interrupt Pin (INT_PN) — RO. Hardwired to 010b to select PIRQB#. |

## 16.1.17 PID—PCI Power Management Capability Identification Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 50h | Attribute: | RO |
| Default Value: | 0001h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:8 | Next Capability (NEXT) — RO. This field indicates that this is the last item in the list. |
| 7:0 | Capability ID (CAP) — RO. This field indicates that this pointer is a message signaled interrupt capability. |

## 16.1.18 PC—Power Management Capabilities Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 52h | Attribute: | RO |
| Default Value: | C9C2h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 15:11 | PME Support — RO. This field indicates PME# can be generated from all D states. |
| 10:9 | Reserved. |
| 8:6 | Auxiliary Current — RO. This field reports 375 mA maximum Suspend well current required when in the D3cold state. |
| 5 | Device Specific Initialization (DSI) — RO. This bit indicates that no device-specific initialization is required. |
| 4 | Reserved — RO. |
| 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 the *PCI Power Management Specification, Revision 1.1*. |

## 16.1.19 PCS—Power Management Control and Status Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| Address Offset: | 54h | Attribute: | R/W, R/WC |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Resume |

This register is not affected by the $D3_{HOT}$ to D0 transition.

| Bit | Description |
|---|---|
| 15 | **PME Status (PMES)** — RW/C.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit is set when the AC '97 controller would normally assert the PME# signal independent of the state of the PME_En bit. This bit resides in the resume well. |
| 14:9 | Reserved — RO. |
| 8 | **PME Enable (PMEE)** — R/W.<br>0 = Disable<br>1 = Enable. When set, and if corresponding PMES is also set, the AC '97 controller sets the AC97_STS bit in the GPE0_STS register. |
| 7:2 | Reserved — RO. |
| 1:0 | **Power State (PS)** — R/W. This field is used both to determine the current power state of the AC '97 controller and to set a new power state. The values are:<br>00 = D0 state<br>01 = not supported<br>10 = not supported<br>11 = $D3_{HOT}$ state<br>When in the $D3_{HOT}$ state, the AC '97 controller's configuration space is available, but the I/O and memory spaces are not. Additionally, interrupts are blocked.<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. |

## 16.2 AC '97 Modem I/O Space (D31:F6)

In the case of the split codec implementation accesses to the modem mixer registers in different codecs are differentiated by the controller by using address offsets 00h–7Fh for the primary codec and address offsets 80h–FEh for the secondary codec. Table 172 shows the register addresses for the modem mixer registers.

**Table 172. Intel® ICH5 Modem Mixer Register Configuration**

| Register | | MMBAR Exposed Registers (D31:F6) |
|---|---|---|
| **Primary** | **Secondary** | **Name** |
| 00h:38h | 80h:B8h | Intel RESERVED |
| 3Ch | BCh | Extended Modem ID |
| 3Eh | BEh | Extended Modem Stat/Ctrl |
| 40h | C0h | Line 1 DAC/ADC Rate |
| *42h* | *C2h* | *Line 2 DAC/ADC Rate* |
| *44h* | *C4h* | *Handset DAC/ADC Rate* |
| 46h | C6h | Line 1 DAC/ADC Level Mute |
| *48h* | *C8h* | *Line 2 DAC/ADC Level Mute* |
| *4Ah* | *CAh* | *Handset DAC/ADC Level Mute* |
| 4Ch | CCh | GPIO Pin Config |
| 4Eh | CEh | GPIO Polarity/Type |
| 50h | D0h | GPIO Pin Sticky |
| 52h | D2h | GPIO Pin Wake Up |
| 54h | D4h | GPIO Pin Status |
| 56h | D6h | Misc. Modem AFE Stat/Ctrl |
| 58h | D8h | AC '97 Reserved |
| 5Ah | DAh | Vendor Reserved |
| 7Ch | FCh | Vendor ID1 |
| 7Eh | FEh | Vendor ID2 |

**NOTES:**
1. Registers in italics are for functions not supported by the ICH5
2. Software should not try to access reserved registers.
3. The ICH5 supports a modem codec connected to AC_SDIN[2:0], as long as the Codec ID is 00 or 01. However, the ICH5 does not support more than one modem codec. For a complete list of topologies, see your ICH5 enabled Platform Design Guide.

The Global Control (GLOB_CNT) and Global Status (GLOB_STA) registers are aliased to the same global registers in the audio and modem I/O space. Therefore a read/write to these registers in either audio or modem I/O space affects the same physical register. Software could access these registers as bytes, word, DWord quantities, but reads must not cross DWord boundaries.

These registers exist in I/O space and reside in the AC '97 controller. The two channels, Modem in and Modem out, each have their own set of Bus Mastering registers. The following register descriptions apply to both channels. The naming prefix convention used is as follows:
MI = Modem in channel
MO = Modem out channel

### Table 173. Modem Registers

| Offset | Mnemonic | Name | Default | Access |
|---|---|---|---|---|
| 00h–03h | MI_BDBAR | Modem In Buffer Descriptor List Base Address | 00000000h | R/W |
| 04h | MI_CIV | Modem In Current Index Value | 00h | RO |
| 05h | MI_LVI | Modem In Last Valid Index | 00h | R/W |
| 06h–07h | MI_SR | Modem In Status | 0001h | R/WC, RO |
| 08h–09h | MI_PICB | Modem In Position In Current Buffer | 0000h | RO |
| 0Ah | MI_PIV | Modem In Prefetch Index Value | 00h | RO |
| 0Bh | MI_CR | Modem In Control | 00h | R/W, R/W (special) |
| 10h–13h | MO_BDBAR | Modem Out Buffer Descriptor List Base Address | 00000000h | R/W |
| 14h | MO_CIV | Modem Out Current Index Value | 00h | RO |
| 15h | MO_LVI | Modem Out Last Valid | 00h | R/W |
| 16h–17h | MO_SR | Modem Out Status | 0001h | R/WC, RO |
| 18h–19h | MI_PICB | Modem In Position In Current Buffer | 0000h | RO |
| 1Ah | MO_PIV | Modem Out Prefetched Index | 00h | RO |
| 1Bh | MO_CR | Modem Out Control | 00h | R/W, R/W (special) |
| 3Ch–3Fh | GLOB_CNT | Global Control | 00000000h | R/W, R/W (special) |
| 40h–43h | GLOB_STA | Global Status | 00300000h | RO, R/W, R/WC |
| 44h | CAS | Codec Access Semaphore | 00h | R/W (special) |

**NOTE:**
1. MI = Modem in channel; MO = Modem out channel

***Note:*** Internal reset as a result of $D3_{HOT}$ to D0 transition will reset all the core well registers except the registers shared with the AC '97 audio controller (GCR, GSR, CASR). All resume well registers will not be reset by the $D3_{HOT}$ to D0 transition.

Core well registers and bits **not** reset by the $D3_{HOT}$ to D0 transition:

- offset 3Ch–3Fh – bits [6:0] Global Control (GLOB_CNT)
- offset 40h–43h – bits [29,15,11:10] Global Status (GLOB_STA)
- offset 44h – Codec Access Semaphore Register (CAS)

Resume well registers and bits **will not** be reset by the $D3_{HOT}$ to D0 transition:

- offset 40h–43h – bits [17:16] Global Status (GLOB_STA)

## 16.2.1 x_BDBAR—Buffer Descriptor List Base Address Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| I/O Address: | MBAR + 00h (MIBDBAR), MBAR + 10h (MOBDBAR) | Attribute: | R/W |
| Default Value: | 00000000h | Size: | 32bits |
| Lockable: | No | Power Well: | Core |

Software can read the register at offset 00h by performing a single, 32-bit read from address offset 00h. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 31:3 | **Buffer Descriptor List Base Address [31:3]** — R/W. These bits represent address bits 31:3. The entries should be aligned on 8-byte boundaries. |
| 2:0 | Hardwired to 0. |

## 16.2.2 x_CIV—Current Index Value Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| I/O Address: | MBAR + 04h (MICIV), MBAR + 14h (MOCIV), | Attribute: | RO |
| Default Value: | 00h | Size: | 8bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 8-bit read to offset 04h. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 7:5 | Hardwired to 0. |
| 4:0 | **Current Index Value [4:0]** — RO. These bits represent which buffer descriptor within the list of 16 descriptors is being processed currently. As each descriptor is processed, this value is incremented. |

## 16.2.3 x_LVI—Last Valid Index Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| I/O Address: | MBAR + 05h (MILVI), MBAR + 15h (MOLVI) | Attribute: | R/W |
| Default Value: | 00h | Power Well: | Core |

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 8-bit read to offset 05h. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 7:5 | Hardwired to 0 |
| 4:0 | **Last Valid Index [4:0]** — R/W. These bits indicate the last valid descriptor in the list. This value is updated by the software as it prepares new buffers and adds to the list. |

## 16.2.4 x_SR—Status Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| I/O Address: | MBAR + 06h (MISR), MBAR + 16h (MOSR) | Attribute: | R/WC, RO |
| Default Value: | 0001h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at offsets 04h, 05h and 06h simultaneously by performing a single, 32-bit read from address offset 04h. Software can also read this register individually by doing a single, 16-bit read to offset 06h. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 15:5 | Reserved |
| 4 | **FIFO Error (FIFOE)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = FIFO error occurs.<br>**Modem in:** FIFO error indicates a FIFO overrun. The FIFO pointers don't increment, the incoming data is not written into the FIFO, thereby being lost.<br>**Modem out:** FIFO error indicates a FIFO underrun. The sample transmitted in this case should be the last valid sample.<br>The Intel® ICH5 will set the FIFOE bit if the under-run or overrun occurs when there are more valid buffers to process. |
| 3 | **Buffer Completion Interrupt Status (BCIS)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by the hardware after the last sample of a buffer has been processed, AND if the Interrupt on Completion (IOC) bit is set in the command byte of the buffer descriptor. Remains active until software clears bit. |
| 2 | **Last Valid Buffer Completion Interrupt (LVBCI)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Set by hardware when last valid buffer has been processed. It remains active until cleared by software. This bit indicates the occurrence of the event signified by the last valid buffer being processed. Thus, this is an event status bit that can be cleared by software once this event has been recognized. This event will cause an interrupt if the enable bit in the Control Register is set. The interrupt is cleared when the software clears this bit.<br><br>In the case of transmits (PCM out, Modem out) this bit is set, after the last valid buffer has been fetched (not after transmitting it). While in the case of Receives, this bit is set after the data for the last buffer has been written to memory. |
| 1 | **Current Equals Last Valid (CELV)** — RO.<br>0 = Hardware clears when controller exists state (i.e., until a new value is written to the LVI register).<br>1 = Current Index is equal to the value in the Last Valid Index Register, AND the buffer pointed to by the CIV has been processed (i.e., after the last valid buffer has been processed). This bit is very similar to bit 2, except, this bit reflects the state rather than the event. This bit reflects the state of the controller, and remains set until the controller exits this state. |
| 0 | **DMA Controller Halted (DCH)** — RO.<br>0 = Running.<br>1 = Halted. This could happen because of the Start/Stop bit being cleared and the DMA engines are idle, or it could happen once the controller has processed the last valid buffer. |

## 16.2.5 x_PICB—Position in Current Buffer Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| I/O Address: | MBAR + 08h (MIPICB), MBAR + 18h (MOPICB), | Attribute: | RO |
| Default Value: | 0000h | Size: | 16 bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 16-bit read to offset 08h. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 15:0 | **Position In Current Buffer[15:0]** — RO. These bits represent the number of samples left to be processed in the current buffer. |

## 16.2.6 x_PIV—Prefetch Index Value Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| I/O Address: | MBAR + 0Ah (MIPIV), MBAR + 1Ah (MOPIV) | Attribute: | RO |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 8-bit read to offset 0Ah. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 7:5 | Hardwired to 0. |
| 4:0 | **Prefetched Index Value [4:0]** — RO. These bits represent which buffer descriptor in the list has been prefetched. |

## 16.2.7 x_CR—Control Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| I/O Address: | MBAR + 0Bh (MICR), MBAR + 1Bh (MOCR) | Attribute: | R/W, R/W (special) |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

Software can read the registers at the offsets 08h, 0Ah, and 0Bh by performing a 32-bit read from the address offset 08h. Software can also read this register individually by doing a single, 8-bit read to offset 0Bh. Reads across DWord boundaries are not supported.

| Bit | Description |
|---|---|
| 7:5 | Reserved |
| 4 | **Interrupt on Completion Enable (IOCE)** — R/W. This bit controls whether or not an interrupt occurs when a buffer completes with the IOC bit set in its descriptor.<br>0 = Disable<br>1 = Enable |
| 3 | **FIFO Error Interrupt Enable (FEIE)** — R/W. This bit controls whether the occurrence of a FIFO error will cause an interrupt or not.<br>0 = Disable. Bit 4 in the Status Register will be set, but the interrupt will not occur.<br>1 = Enable. Interrupt will occur |
| 2 | **Last Valid Buffer Interrupt Enable (LVBIE)** — R/W. This bit controls whether the completion of the last valid buffer will cause an interrupt or not.<br>0 = Disable. Bit 2 in the Status register will still be set, but the interrupt will not occur.<br>1 = Enable |
| 1 | **Reset Registers (RR)** — R/W (special).<br>0 = Removes reset condition.<br>1 = Contents of all registers to be reset, except the interrupt enable bits (bit 4,3,2 of this register). Software needs to set this bit. It must be set only when the Run/Pause bit is cleared. Setting it when the Run bit is set will cause undefined consequences. This bit is self-clearing (software needs not clear it). |
| 0 | **Run/Pause Bus Master (RPBM)** — R/W.<br>0 = Pause bus master operation. This results in all state information being retained (i.e., master mode operation can be stopped and then resumed).<br>1 = Run. Bus master operation starts. |

## 16.2.8 GLOB_CNT—Global Control Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| I/O Address: | MBAR + 3Ch | Attribute: | R/W, R/W (special) |
| Default Value: | 00000000h | Size: | 32 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 31:6 | Reserved. |
| 6 | **AC_SDIN2 Interrupt Enable (S2RE)** — R/W.<br>0 = Disable<br>1 = Enable an interrupt to occur when the codec on the AC_SDIN2 causes a resume event on the AC-link. |
| 5 | **AC_SDIN1 Resume Interrupt Enable (S1RE)** — R/W.<br>0 = Disable<br>1 = Enable an interrupt to occur when the codec on the AC_SDIN1 causes a resume event on the AC-link. |
| 4 | **AC_SDIN0 Resume Interrupt Enable (S0RE)** — R/W.<br>0 = Disable<br>1 = Enable an interrupt to occur when the codec on AC_SDIN0 causes a resume event on the AC-link. |
| 3 | **ACLINK Shut Off (LSO)** — R/W.<br>0 = Normal operation.<br>1 = Controller disables all outputs which will be pulled low by internal pull down resistors. |
| 2 | **AC '97 Warm Reset** — R/W (special).<br>0 = Normal operation.<br>1 = Writing a 1 to this bit causes a warm reset to occur on the AC-link. The warm reset will awaken a suspended codec without clearing its internal registers. If software attempts to perform a warm reset while AC_BIT_CLK is running, the write will be ignored and the bit will not change. This bit is self-clearing (it remains set until the reset completes and AC_BIT_CLK is seen on the AC-link, after which it clears itself). |
| 1 | **AC '97 Cold Reset#** — R/W.<br>0 = Writing a 0 to this bit causes a cold reset to occur throughout the AC '97 circuitry. All data in the controller and the codec will be lost. Software needs to clear this bit no sooner than the minimum number of ms have elapsed.<br>1 = This bit defaults to 0 and hence after reset, the driver needs to set this bit to a 1. The value of this bit is retained after suspends; hence, if this bit is set to a 1 prior to suspending, a cold reset is not generated automatically upon resuming.<br>Note: This bit is in the Core well. |
| 0 | **GPI Interrupt Enable (GIE)** — R/W. This bit controls whether the change in status of any GPI causes an interrupt.<br>0 = Bit 0 of the Global Status Register is set, but no interrupt is generated.<br>1 = The change on value of a GPI causes an interrupt and sets bit 0 of the Global Status Register. |

***Note:*** Reads across DWord boundaries are not supported.

## 16.2.9 GLOB_STA—Global Status Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| I/O Address: | MBAR + 40h | Attribute: | RO, R/W, R/WC |
| Default Value: | 00300000h | Size: | 32 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 31:30 | Reserved. |
| 29 | **AC_SDIN2 Resume Interrupt (S2RI)** — R/WC. This bit indicates a resume event occurred on AC_SDIN2.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = Resume event occurred.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |
| 28 | **AC_SDIN2 Codec Ready (S2CR)** — RO. This bit reflects the state of the codec ready bit on AC_SDIN2. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously.<br>0 = Not Ready.<br>1 = Ready. |
| 27 | **Bit Clock Stopped (BCS)** — RO. This bit indicates that the bit clock is not running.<br>0 = Transition is found on AC_BIT_CLK.<br>1 = Intel® ICH5 detects that there has been no transition on AC_BIT_CLK for four consecutive PCI clocks. |
| 26 | **S/PDIF Interrupt (SPINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = S/PDIF out channel interrupt status bits have been set. |
| 25 | **PCM In 2 Interrupt (P2INT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the PCM In 2 channel status bits have been set. |
| 24 | **Microphone 2 In Interrupt (M2INT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the Mic in channel interrupts status bits has been set. |
| 23:22 | **Sample Capabilities** — RO. This field indicates the capability to support more greater than 16-bit audio.<br>00 = Reserved<br>01 = 16 and 20-bit Audio supported (ICH5 value)<br>10 = Reserved<br>11 = Reserved |
| 21:20 | **Multichannel Capabilities** — RO. This field indicates the capability to support 4 and 6 channels on PCM Out. |
| 19:18 | Reserved. |
| 17 | **MD3** — R/W. Power down semaphore for Modem. This bit exists in the suspend well and maintains context across power states (except G3). The bit has no hardware function. It is used by software in conjunction with the AD3 bit to coordinate the entry of the two codecs into D3 state.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |
| 16 | **AD3** — R/W. Power down semaphore for Audio. This bit exists in the suspend well and maintains context across power states (except G3). The bit has no hardware function. It is used by software in conjunction with the MD3 bit to coordinate the entry of the two codecs into D3 state.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |

| Bit | Description |
|---|---|
| 15 | **Read Completion Status (RCS)** — R/WC. This bit indicates the status of codec read completions. Software clears this bit by writing a 1 to it.<br>0 = A codec read completes normally.<br>1 = A codec read results in a time-out.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |
| 14 | **Bit 3 of Slot 12** — RO. Display bit 3 of the most recent slot 12. |
| 13 | **Bit 2 of Slot 12** — RO. Display bit 2 of the most recent slot 12. |
| 12 | **Bit 1 of Slot 12** — RO. Display bit 1 of the most recent slot 12. |
| 11 | **AC_SDIN1 Resume Interrupt (S1RI)** — R/WC. This bit indicates that a resume event occurred on AC_SDIN1. Software clears this bit by writing a 1 to it.<br>0 = Resume event did **not** occur.<br>1 = Resume event occurred.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |
| 10 | **AC_SDIN0 Resume Interrupt (S0RI)** — R/WC. This bit indicates that a resume event occurred on AC_SDIN0. Software clears this bit by writing a 1 to it.<br>0 = Resume event did **not** occur.<br>1 = Resume event occurred.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |
| 9 | **AC_SDIN1 Codec Ready (S1CR)** — RO. This bit reflects the state of the codec ready bit in AC_SDIN1. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously.<br>0 = Not Ready.<br>1 = Ready. |
| 8 | **AC_SDIN0 Codec Ready (S0CR)** — RO. This bit reflects the state of the codec ready bit in AC_SDIN 0. Bus masters ignore the condition of the codec ready bits, so software must check this bit before starting the bus masters. Once the codec is "ready", it must never go "not ready" spontaneously.<br>0 = Not Ready.<br>1 = Ready. |
| 7 | **Microphone In Interrupt (MINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the Mic in channel interrupts status bits has been set. |
| 6 | **PCM Out Interrupt (POINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the PCM out channel interrupts status bits has been set. |
| 5 | **PCM In Interrupt (PIINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the PCM in channel interrupts status bits has been set. |
| 4:3 | Reserved |
| 2 | **Modem Out Interrupt (MOINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the modem out channel interrupts status bits has been set. |
| 1 | **Modem In Interrupt (MIINT)** — RO.<br>0 = When the specific status bit is cleared, this bit will be cleared.<br>1 = One of the modem in channel interrupts status bits has been set. |
| 0 | **GPI Status Change Interrupt (GSCI)** — R/WC.<br>0 = Software clears this bit by writing a 1 to it.<br>1 = This bit reflects the state of bit 0 in slot 12, and is set when bit 0 of slot 12 is set. This indicates that one of the GPI's changed state, and that the new values are available in slot 12.<br>This bit is not affected by $D3_{HOT}$ to D0 Reset. |

*Note:* On reads from a codec, the controller will give the codec a maximum of four frames to respond, after which if no response is received, it will return a dummy read completion to the processor (with all F's on the data) and also set the Read Completion Status bit in the GSR.

*Note:* Reads across DWord boundaries are not supported.

## 16.2.10 CAS—Codec Access Semaphore Register (Modem—D31:F6)

| | | | |
|---|---|---|---|
| I/O Address: | NABMBAR + 44h | Attribute: | R/W (special) |
| Default Value: | 00h | Size: | 8 bits |
| Lockable: | No | Power Well: | Core |

| Bit | Description |
|---|---|
| 7:1 | Reserved |
| 0 | **Codec Access Semaphore (CAS)** — R/W (special). This bit is read by software to check whether a codec access is currently in progress.<br>0 = No access in progress.<br>1 = The act of reading this register sets this bit to 1. The driver that read this bit can then perform an I/O access. Once the access is completed, hardware automatically clears this bit. |

*Note:* Reads across DWord boundaries are not supported.

# High-Precision Event Timer Registers17

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 General Control register (offset D0h) in Device 31, Function 0.

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.

**Table 174. Memory-Mapped Registers**

| Offset | Mnemonic | Register | Default | Type |
|---|---|---|---|---|
| 000–007h | GCAP_ID | General Capabilities and Identification | 0429B17F8086A201h | RO |
| 008–00Fh | — | Reserved | — | — |
| 010–017h | GEN_CONF | General Configuration | 0000h | R/W |
| 018–01Fh | — | Reserved | — | — |
| 020–027h | GINTR_STA | General Interrupt Status | 0000h | R/WC |
| 028–0EFh | — | Reserved | — | — |
| 0F0–0F7h | MAIN_CNT | Main Counter Value | N/A | R/W |
| 0F8–0FFh | — | Reserved | — | — |
| 100–107h | TIM1_CONF | Timer 0 Configuration and Capabilities | N/A | R/W |
| 108–10Fh | TIM1_COMP | Timer 0 Comparator Value | N/A | R/W |
| 110–11Fh | — | Reserved | — | — |
| 120–127h | TIM2_CONF | Timer 1 Configuration and Capabilities | N/A | R/W |
| 128–12Fh | TIM2_COMP | Timer 1 Comparator Value | N/A | R/W |
| 130–13Fh | — | Reserved | — | — |
| 140–147h | TIM3_CONF | Timer 2 Configuration and Capabilities | N/A | R/W |
| 148–14Fh | TIM3_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 Multimedia Timers. If attempted, the lock is not honored, which means potential deadlock conditions may occur.

## 17.1 GCAP_ID—General Capabilities and Identification Register

Address Offset: 00h
Default Value: 0429B17F8086A201h
Attribute: RO
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 0429B17F when read. This indicates a period of 69841279 fs (69.841279 ns). |
| 31:16 | Vendor ID Capability (VENDOR_ID_CAP) — RO. This is a 16-bit value assigned to Intel. |
| 15 | Legacy Rout Capable (LEG_RT_CAP) — RO. Hardwired to 1. Legacy 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. |

## 17.2 GEN_CONF—General Configuration Register

Address Offset: 010h
Default Value: 0000h
Attribute: R/W
Size: 64 bits

| Bit | Description |
|---|---|
| 63:2 | Reserved. These bits return 0 when read. |
| 1 | **Legacy 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 Rout bit is set, the individual routing bits for Timers 0 and 1 (APIC) will have no impact.<br>• If the Legacy 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 routing, or 0 to disable the legacy routing. |
| 0 | **Overall Enable (ENABLE_CNF)** — R/W. This bit must be set to enable any of the timers to generate interrupts. If this bit is 0, then 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>**NOTE:** This bit will default to 0. BIOS can set it to 1 or 0. |

## 17.3 GINTR_STA—General Interrupt Status Register

Address Offset: 020h
Default Value: 0000h
Attribute: R/W, R/WC
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>**NOTE:** Defaults to 0. In edge triggered mode, this bit will always read as 0 and writes will have no effect. |

## 17.4 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>**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). |

## 17.5 TIMn_CONF—Timer n Configuration and Capabilities Register

| | | | |
|---|---|---|---|
| Address Offset: | Timer 0: 100–107h,<br>Timer 1: 120–127h,<br>Timer 2: 140–147h | Attribute: | RO, R/W |
| Default Value: | N/A | Size: | 64 bits |

***Note:*** The letter n can be 0, 1, or 2, referring to Timer 0, 1 or 2.

| Bit | Description |
|---|---|
| 64:56 | 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>**NOTE:** If IRQ 11 is used for HPET #2, software should ensure IRQ 11 is not shared with any other devices to guarantee the proper operation of HPET #2. |
| 51:44, 42:14 | Reserved**.** These bits return 0 when read. |
| 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>**NOTES:**<br>1. If the Legacy 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 Intel® ICH5 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 ICH5 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.<br>Timer 0:Bit is read/write (default to 0). 1 = 64 bit; 0 = 32 bit<br>Timers 1, 2:Hardwired to 0. Writes have no effect (since these two timers are 32-bits). |
| 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>**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). |

| Bit | Description |
|---|---|
| 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 affect. |
| 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>1 = Enable<br>0 = Disable (Default). The timer can still count and generate appropriate status bits, but will not cause an interrupt. |
| 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.

## 17.6 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>For example, if the value written to the register is 00000123h, then<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>• 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. |

This page is intentionally left blank.

# Ballout Definition 18

This section contains the ICH5 ballout information. Figure 1 and Figure 2 are the ballout map of the 460 mBGA package. Table 175 is an mBGA ball list, sorted alphabetically by signal name. Table 176 is an mBGA ball list, sorted alphabetically by ball number.

## Figure 1. Intel® ICH5 Ballout (Topview–Left Side)

| Begin | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| A | VSS | PIRQC# | GNT1# | GNT4# / GPIO48 | REQA# / GPIO0 | PIRQF# / GPIO3 | VSS | V5REF | AC_SDOUT | VSS | NO CONNECT | EE_SHCLK |
| B | PIRQH# / GPIO5 | AD18 | PIRQA# | GNTB# / GNT5# / GPIO17 | Vcc3_3 | REQ3# | GNT2# | AC_SYNC | EE_DOUT | EE_CS | EE_DIN | LAN_TXD2 |
| C | REQ1# | PIRQD# | VSS | AD22 | REQ2# | REQ4# / GPIO40 | GNT3# | VSS | LAN_RXD1 | LAN_RXD0 | LAN_RXD2 | AC_RST# |
| D | VSS | FRAME# | AD26 | GNT0# | REQ0# | VSS | PIRQE# / GPIO2 | AC_BIT_CLK | LAN_TXD0 | LAN _RSTSYNC | VSS | AC_SDIN1 |
| E | PIRQB# | PIRQG# / GPIO4 | C/BE0# | TRDY# | STOP# | AD24 | REQB# / REQ5#/ GPIO1 | GNTA# / GPIO16 | LAN_TXD1 | LAN_CLK | VCCSUS3_3 | AC_SDIN0 |
| F | PAR | AD9 | VSS | AD30 | AD28 | VCC3_3 | VCCSUS1_5 _C | VCCSUS1_5 _C | VSS | VCCSUS3_3 | VCCSUs3_3 | VOID |
| G | VCC3_3 | AD13 | AD2 | AD16 | AD15 | VSS | VOID | VOID | VOID | VOID | VOID | VOID |
| H | VSS | AD5 | AD20 | AD11 | AD4 | VCC3_3 | VOID | VOID | VOID | VOID | VOID | VOID |
| J | C/BE1# | AD7 | AD6 | AD0 | AD1 | VSS | VOID | VOID | VOID | VOID | VOID | VOID |
| K | AD14 | PERR# | VSS | AD3 | AD8 | VCC3_3 | VOID | VOID | VOID | Vcc1_5 | VSS | Vcc1_5 |
| L | AD17 | PLOCK# | DEVSEL# | SERR# | AD12 | VCC3_3 | VOID | VOID | VOID | VSS | VSS | VSS |
| M | VSS | C/BE3# | IRDY# | AD10 | VSS | VOID | VOID | VOID | VOID | Vcc3_3 | VSS | VSS |
| N | PCICLK | AD27 | C/BE2# | AD23 | AD21 | VOID | VOID | VOID | VOID | Vcc3_3 | VSS | VSS |
| P | VSS | AD31 | AD25 | AD29 | AD19 | VCC3_3 | VOID | VOID | VOID | VSS | VSS | VSS |
| R | GPIO21 | LDRQ1# / GPIO41 | LAD2 / FB2 | LAD1 / FB1 | GPIO6 | VCC1_5 | VOID | VOID | VOID | Vcc1_5 | VSS | Vcc1_5 |
| T | GPIO32 | THRM# | VSS | LFRAME# / FB4 | LAD0 / FB0 | VSS | VOID | VOID | VOID | VOID | VOID | VOID |
| U | SYS_RESET# | SLP_S4# | GPIO7 | LAD3 / FB3 | LDRQ0# | VCCSUS3_3 | VOID | VOID | VOID | VOID | VOID | VOID |
| V | VSS | PME# | GPIO27 | PCIRST# | LINKALERT# | VCCSUS3_3 | VOID | VOID | VOID | VOID | VOID | VOID |
| W | SLP_S3# | GPIO28 | GPIO25 | GPIO12 | GPIO13 | VCC1_5 | VCC1_5 | VCC1_5 | VCC1_5 | VCC1_5 | VCC1_5 | VOID |
| Y | SUSCLK | GPIO8 | VSS | PWRBTN# | VCCSUS1_5 _B | VSS | VSS | VSS | SATARBIAS# | VSS | SATARBIAS | INTRUDER# |
| AA | LAN_RST# | SMLINK1 | SLP_S5# | VCCSUS1_5 _B | VSS | VCCSATA PLL | VSS | SATA0TXP | VSS | SATA1TXP | VSS | RTCRST# |
| AB | SUS_STAT# | TP0 | RI# | VCCSUS1_5 _B | VSS | VCCSATA PLL | VSS | SATA0TXN | VSS | SATA1TXN | VSS | RTCX2 |
| AC | GPIO24 | VSS | SMBALERT# / GPIO11 | VSS | CLK100P | VSS | SATA0RXP | VSS | SATA1RXP | VSS | RTCX1 | PWROK |
| AD | SMBDATA | SMBCLK | SMLINK0 | VSS | CLK100N | VSS | SATA0RXN | VSS | SATA1RXN | INTVRMEN | VCCRTC | VSS |
| end | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |

## Figure 2. Intel® ICH5 Ballout (Topview–Right Side)

| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AC_SDIN2 | OC5# / GPIO10 | VSS | USBP7P | VSS | USBP5P | VSS | USBP3P | VSS | USBP1P | VSS | USBRBIAS | A |
| VSS | OC4# / GPIO9 | VCCSUS3_3 | USBP7N | VSS | USBP5N | VSS | USBP3N | VSS | USBP1N | VSS | USBRBIAS# | B |
| OC7# / GPIO15 | OC3# | OC0# | VSS | USBP6P | VSS | USBP4P | VSS | USBP2P | VSS | USBP0P | VCCUSB PLL | C |
| OC6# / GPIO14 | OC2# | OC1# | VSS | USBP6N | VSS | USBP4N | VSS | USBP2N | VSS | USBP0N | VSS | D |
| VCCSUS3_3 | VCCSUS3_3 | VCC1_5 | V5REF_SUS | VSS | VCCSUS3_3 | VSS | VSS | VSS | VCC1_5 | VSS | SPKR | E |
| VOID | VCC1_5 | VCC1_5 | VCCSUS3_3 | VCCSUS3_3 | VCCSUS3_3 | VCCSUS1_5_A | CLK14 | GPIO34 | GPIO23 | SERIRQ | CLK48 | F |
| VOID | VOID | VOID | VOID | VOID | VOID | VCC3_3 | VSS | VCC3_3 | HI11 | SATALED# | VSS | G |
| VOID | VOID | VOID | VOID | VOID | VOID | VSS | HI0 | HI1 | VSS | HI3 | VCC1_5 | H |
| VOID | VOID | VOID | VOID | VOID | VOID | VCC1_5 | HI2 | VSS | HI9 | VSS | HI_STBS | J |
| VCC1_5 | VSS | VCCSUS3_3 | VOID | VOID | VOID | VCC1_5 | VSS | HI10 | VSS | HI_STBF | VSS | K |
| VSS | VSS | VSS | VOID | VOID | VOID | VCC1_5 | HI_VSWING | VSS | HI8 | VSS | HIREF | L |
| VSS | VSS | VCC1_5 | VOID | VOID | VOID | VOID | HI7 | HI5 | VSS | HI4 | VSS | M |
| VSS | VSS | VCC1_5 | VOID | VOID | VOID | VOID | VSS | HI6 | CLK66 | VCC1_5 | HIRCOMP | N |
| VSS | VSS | VSS | VOID | VOID | VOID | VCC1_5 | TP1 | VSS | CPUSLP# | RCIN# | CPUPWRGD / GPIO49 | P |
| VCC3_3 | VSS | V_CPU_IO | VOID | VOID | VOID | V_CPU_IO | VRMPWRGD | IGNNE# | NMI | INIT# | TP2 | R |
| VOID | VOID | VOID | VOID | VOID | VOID | V_CPU_IO | GPIO19 | THRMTRIP# | A20GATE | VSS | STPCLK# | T |
| VOID | VOID | VOID | VOID | VOID | VOID | VSS | GPIO22 | GPIO18 | GPIO20 | INTR | FERR# | U |
| VOID | VOID | VOID | VOID | VOID | VOID | VCC3_3 | SDCS3# | VSS | SDCS1# | A20M# | SMI# | V |
| VOID | V5REF | VCC3_3 | VSS | VCC3_3 | VSS | VCC1_5 | SDDACK# | SDA2 | SDA0 | SDA1 | VCC3_3 | W |
| PDD1 | PDD2 | PDD9 | PDD13 | IRQ14 | PDCS3# | SDD8 | SDDREQ | SIORDY | SDIOW# | SDIOR# | IRQ15 | Y |
| VSS | PDD4 | PDD11 | PDD14 | PDIOW# | PIORDY | PDA0 | SDD11 | VSS | SDD0 | SDD15 | VSS | AA |
| RSMRST# | PDD7 | VSS | PDD0 | PDD15 | VSS | PDCS1# | SDD6 | SDD4 | SDD12 | SDD1 | SDD14 | AB |
| VSS | PDD3 | PDD5 | PDD12 | PDDREQ | PDDACK# | PDA2 | SDD7 | SDD5 | SDD10 | VSS | SDD13 | AC |
| VCC3_3 | PDD6 | PDD8 | PDD10 | VSS | PDIOR# | PDA1 | VCC3_3 | VSS | SDD9 | SDD2 | SDD3 | AD |
| 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| A20GATE | T22 |
| A20M# | V23 |
| AC_BIT_CLK | D8 |
| AC_RST# | C12 |
| AC_SDIN0 | E12 |
| AC_SDIN1 | D12 |
| AC_SDIN2 | A13 |
| AC_SDOUT | A9 |
| AC_SYNC | B8 |
| AD0 | J4 |
| AD1 | J5 |
| AD2 | G3 |
| AD3 | K4 |
| AD4 | H5 |
| AD5 | H2 |
| AD6 | J3 |
| AD7 | J2 |
| AD8 | K5 |
| AD9 | F2 |
| AD10 | M4 |
| AD11 | H4 |
| AD12 | L5 |
| AD13 | G2 |
| AD14 | K1 |
| AD15 | G5 |
| AD16 | G4 |
| AD17 | L1 |
| AD18 | B2 |
| AD19 | P5 |
| AD20 | H3 |
| AD21 | N5 |
| AD22 | C4 |
| AD23 | N4 |
| AD24 | E6 |
| AD25 | P3 |
| AD26 | D3 |
| AD27 | N2 |
| AD28 | F5 |
| AD29 | P4 |
| AD30 | F4 |
| AD31 | P2 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| GPIO6 | R5 |
| TP0 | AB2 |
| C/BE0# | E3 |
| C/BE1# | J1 |
| C/BE2# | N3 |
| C/BE3# | M2 |
| GPIO21 | R1 |
| CLK14 | F20 |
| CLK48 | F24 |
| CLK66 | N22 |
| CLK100N | AD5 |
| CLK100P | AC5 |
| GPIO24 | AC1 |
| GPIO22 | U20 |
| CPUPWRGD/GPIO49 | P24 |
| CPUSLP# | P22 |
| DEVSEL# | L3 |
| TP1 | P20 |
| TP2 | R24 |
| EE_CS | B10 |
| EE_DIN | B11 |
| EE_DOUT | B9 |
| EE_SHCLK | A12 |
| FERR# | U24 |
| FRAME# | D2 |
| GNT0# | D4 |
| GNT1# | A3 |
| GNT2# | B7 |
| GNT3# | C7 |
| GNT4#/GPIO48 | A4 |
| GNTA#/GPIO16 | E8 |
| GNTB#/ GNT5#GPIO17 | B4 |
| GPIO7 | U3 |
| GPIO8 | Y2 |
| GPIO12 | W4 |
| GPIO13 | W5 |
| GPIO25 | W3 |
| GPIO27 | V3 |
| GPIO28 | W2 |
| GPIO32 | T1 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| GPIO34 | F21 |
| HI_STBF | K23 |
| HI_STBS | J24 |
| HI0 | H20 |
| HI1 | H21 |
| HI2 | J20 |
| HI3 | H23 |
| HI4 | M23 |
| HI5 | M21 |
| HI6 | N21 |
| HI7 | M20 |
| HI8 | L22 |
| HI9 | J22 |
| HI10 | K21 |
| HI11 | G22 |
| HIRCOMP | N24 |
| HIREF | L24 |
| HI_VSWING | L20 |
| IGNNE# | R21 |
| INIT# | R23 |
| INTR | U23 |
| INTRUDER# | Y12 |
| INTVRMEN | AD10 |
| IRDY# | M3 |
| IRQ14 | Y17 |
| IRQ15 | Y24 |
| LAD0 | T5 |
| LAD1 | R4 |
| LAD2 | R3 |
| LAD3 | U4 |
| LAN_CLK | E10 |
| LAN_RST# | AA1 |
| LAN_RSTSYNC | D10 |
| LAN_RXD0 | C10 |
| LAN_RXD1 | C9 |
| LAN_RXD2 | C11 |
| LAN_TXD0 | D9 |
| LAN_TXD1 | E9 |
| LAN_TXD2 | B12 |
| LDRQ0# | U5 |
| LDRQ1# / GPIO41 | R2 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| LFRAME# | T4 |
| LINKALERT# | V5 |
| NMI | R22 |
| No Connect | A11 |
| OC0# | C15 |
| OC1# | D15 |
| OC2# | D14 |
| OC3# | C14 |
| OC4# / GPIO9 | B14 |
| OC5# / GPIO10 | A14 |
| OC6# / GPIO14 | D13 |
| OC7# / GPIO15 | C13 |
| PAR | F1 |
| PCICLK | N1 |
| PCIRST# | V4 |
| PDA0 | AA19 |
| PDA1 | AD19 |
| PDA2 | AC19 |
| PDCS1# | AB19 |
| PDCS3# | Y18 |
| PDD0 | AB16 |
| PDD1 | Y13 |
| PDD2 | Y14 |
| PDD3 | AC14 |
| PDD4 | AA14 |
| PDD5 | AC15 |
| PDD6 | AD14 |
| PDD7 | AB14 |
| PDD8 | AD15 |
| PDD9 | Y15 |
| PDD10 | AD16 |
| PDD11 | AA15 |
| PDD12 | AC16 |
| PDD13 | Y16 |
| PDD14 | AA16 |
| PDD15 | AB17 |
| PDDACK# | AC18 |
| PDDREQ | AC17 |
| PDIOR# | AD18 |
| PDIOW# | AA17 |
| PERR# | K2 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| PIORDY | AA18 |
| PIRQA# | B3 |
| PIRQB# | E1 |
| PIRQC# | A2 |
| PIRQD# | C2 |
| PIRQE#/GPIO2 | D7 |
| PIRQF#/GPIO3 | A6 |
| PIRQG#/GPIO4 | E2 |
| PIRQH#/GPIO5 | B1 |
| PLOCK# | L2 |
| PME# | V2 |
| PWRBTN# | Y4 |
| PWROK | AC12 |
| RCIN# | P23 |
| REQ0# | D5 |
| REQ1# | C1 |
| REQ2# | C5 |
| REQ3# | B6 |
| REQ4#/GPIO40 | C6 |
| REQA#/GPIO0 | A5 |
| REQB#/REQ5#/ GPIO1 | E7 |
| RI# | AB3 |
| RSMRST# | AB13 |
| RTCRST# | AA12 |
| RTCX1 | AC11 |
| RTCX2 | AB12 |
| SATALED# | G23 |
| SATARBIAS# | Y9 |
| SATARBIAS | Y11 |
| SATA0RXN | AD7 |
| SATA0RXP | AC7 |
| SATA1RXN | AD9 |
| SATA1RXP | AC9 |
| SATA0TXN | AB8 |
| SATA0TXP | AA8 |
| SATA1TXN | AB10 |
| SATA1TXP | AA10 |
| SDA0 | W22 |
| SDA1 | W23 |
| SDA2 | W21 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| SDCS1# | V22 |
| SDCS3# | V20 |
| SDD0 | AA22 |
| SDD1 | AB23 |
| SDD2 | AD23 |
| SDD3 | AD24 |
| SDD4 | AB21 |
| SDD5 | AC21 |
| SDD6 | AB20 |
| SDD7 | AC20 |
| SDD8 | Y19 |
| SDD9 | AD22 |
| SDD10 | AC22 |
| SDD11 | AA20 |
| SDD12 | AB22 |
| SDD13 | AC24 |
| SDD14 | AB24 |
| SDD15 | AA23 |
| SDDACK# | W20 |
| SDDREQ | Y20 |
| SDIOR# | Y23 |
| SDIOW# | Y22 |
| SERIRQ | F23 |
| SERR# | L4 |
| SIORDY | Y21 |
| GPIO19 | T20 |
| SLP_S3# | W1 |
| SLP_S4# | U2 |
| SLP_S5# | AA3 |
| SMBALERT#/GPIO11 | AC3 |
| SMBCLK | AD2 |
| SMBDATA | AD1 |
| SMI# | V24 |
| SMLINK0 | AD3 |
| SMLINK1 | AA2 |
| SPKR | E24 |
| GPIO23 | F22 |
| STOP# | E5 |
| GPIO20 | U22 |
| GPIO18 | U21 |
| STPCLK# | T24 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| SUS_STAT# | AB1 |
| SUSCLK | Y1 |
| SYS_RESET# | U1 |
| THRM# | T2 |
| THRMTRIP# | T21 |
| TRDY# | E4 |
| USBP0N | D23 |
| USBP0P | C23 |
| USBP1N | B22 |
| USBP1P | A22 |
| USBP2N | D21 |
| USBP2P | C21 |
| USBP3N | B20 |
| USBP3P | A20 |
| USBP4N | D19 |
| USBP4P | C19 |
| USBP5N | B18 |
| USBP5P | A18 |
| USBP6N | D17 |
| USBP6P | C17 |
| USBP7N | B16 |
| USBP7P | A16 |
| USBRBIAS | A24 |
| USBRBIAS# | B24 |
| V_CPU_IO | R15 |
| V_CPU_IO | R19 |
| V_CPU_IO | T19 |
| V5REF | A8 |
| V5REF | W14 |
| V5REF_SUS | E16 |
| VCC1_5 | E15 |
| VCC1_5 | F14 |
| VCC1_5 | F15 |
| VCC1_5 | H24 |
| VCC1_5 | J19 |
| VCC1_5 | K19 |
| VCC1_5 | K10 |
| VCC1_5 | K12 |
| VCC1_5 | K13 |
| VCC1_5 | L19 |
| VCC1_5 | M15 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| VCC1_5 | N15 |
| VCC1_5 | N23 |
| VCC1_5 | P19 |
| VCC1_5 | R10 |
| VCC1_5 | R12 |
| VCC1_5 | R6 |
| VCC1_5 | W10 |
| VCC1_5 | W11 |
| VCC1_5 | W19 |
| VCC1_5 | W6 |
| VCC1_5 | W7 |
| VCC1_5 | W8 |
| VCC1_5 | W9 |
| VCC3_3 | AD13 |
| VCC3_3 | AD20 |
| VCC3_3 | B5 |
| VCC3_3 | F6 |
| VCC3_3 | G1 |
| VCC3_3 | G19 |
| VCC3_3 | G21 |
| VCC3_3 | H6 |
| VCC3_3 | K6 |
| VCC3_3 | L6 |
| VCC3_3 | M10 |
| VCC3_3 | N10 |
| VCC3_3 | P6 |
| VCC3_3 | R13 |
| VCC3_3 | V19 |
| VCC3_3 | W15 |
| VCC3_3 | W17 |
| VCC3_3 | W24 |
| VCCSUS1_5_C | F7 |
| VCCSUS1_5_C | F8 |
| VCCSUS3_3 | E11 |
| VCCSUS3_3 | F10 |
| VCCSUS3_3 | F11 |
| VCCRTC | AD11 |
| VCCSATAPLL | AA6 |
| VCCSATAPLL | AB6 |
| VCCSUS1_5_B | AA4 |
| VCCSUS1_5_B | AB4 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| VCC1_5 | E22 |
| VCCSUS1_5_A | F19 |
| VCCSUS1_5_B | Y5 |
| VCCSUS3_3 | B15 |
| VCCSUS3_3 | K15 |
| VCCSUS3_3 | E13 |
| VCCSUS3_3 | E14 |
| VCCSUS3_3 | E18 |
| VCCSUS3_3 | F16 |
| VCCSUS3_3 | F17 |
| VCCSUS3_3 | F18 |
| VCCSUS3_3 | U6 |
| VCCSUS3_3 | V6 |
| VCCUSBPLL | C24 |
| VRMPWRGD | R20 |
| VSS | A1 |
| VSS | A10 |
| VSS | A15 |
| VSS | A17 |
| VSS | A19 |
| VSS | A21 |
| VSS | A23 |
| VSS | A7 |
| VSS | AA11 |
| VSS | AA13 |
| VSS | AA21 |
| VSS | AA24 |
| VSS | AA5 |
| VSS | AA7 |
| VSS | AA9 |
| VSS | AB11 |
| VSS | AB15 |
| VSS | AB18 |
| VSS | AB5 |
| VSS | AB7 |
| VSS | AB9 |
| VSS | AC10 |
| VSS | AC13 |
| VSS | AC2 |
| VSS | AC23 |
| VSS | AC4 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| VSS | AC6 |
| VSS | AC8 |
| VSS | AD4 |
| VSS | AD6 |
| VSS | AD8 |
| VSS | AD12 |
| VSS | AD17 |
| VSS | AD21 |
| VSS | B13 |
| VSS | B17 |
| VSS | B19 |
| VSS | B21 |
| VSS | B23 |
| VSS | C16 |
| VSS | C18 |
| VSS | C20 |
| VSS | C22 |
| VSS | C3 |
| VSS | C8 |
| VSS | D1 |
| VSS | D11 |
| VSS | D16 |
| VSS | D18 |
| VSS | D20 |
| VSS | D22 |
| VSS | D24 |
| VSS | D6 |
| VSS | E17 |
| VSS | E19 |
| VSS | E20 |
| VSS | E21 |
| VSS | E23 |
| VSS | F3 |
| VSS | F9 |
| VSS | G20 |
| VSS | G24 |
| VSS | G6 |
| VSS | H1 |
| VSS | H19 |
| VSS | H22 |
| VSS | J21 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| VSS | J23 |
| VSS | J6 |
| VSS | K11 |
| VSS | K14 |
| VSS | K20 |
| VSS | K22 |
| VSS | K24 |
| VSS | K3 |
| VSS | L10 |
| VSS | L11 |
| VSS | L12 |
| VSS | L13 |
| VSS | L14 |
| VSS | L15 |
| VSS | L21 |
| VSS | L23 |
| VSS | M1 |
| VSS | M11 |
| VSS | M12 |
| VSS | M13 |
| VSS | M14 |
| VSS | M22 |
| VSS | M24 |
| VSS | M5 |
| VSS | N11 |
| VSS | N12 |
| VSS | N13 |
| VSS | N14 |
| VSS | N20 |
| VSS | P1 |
| VSS | P10 |
| VSS | P11 |
| VSS | P12 |
| VSS | P13 |
| VSS | P14 |
| VSS | P15 |
| VSS | P21 |
| VSS | R11 |
| VSS | R14 |
| VSS | T23 |
| VSS | T3 |

### Table 175. Intel® ICH5 Ballout by Signal Name

| Signal Name | Ball # |
|---|---|
| VSS | T6 |
| VSS | U19 |
| VSS | V1 |
| VSS | V21 |
| VSS | W16 |
| VSS | W18 |
| VSS | Y10 |
| VSS | Y3 |
| VSS | Y6 |
| VSS | Y7 |
| VSS | Y8 |

## Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| A1 | VSS |
| A2 | PIRQC# |
| A3 | GNT1# |
| A4 | GNT4#/GPIO48 |
| A5 | REQA#/GPIO0 |
| A6 | PIRQF#/GPIO3 |
| A7 | VSS |
| A8 | V5REF |
| A9 | AC_SDOUT |
| A10 | VSS |
| A11 | No Connect |
| A12 | EE_SHCLK |
| A13 | AC_SDIN2 |
| A14 | OC5# / GPIO10 |
| A15 | VSS |
| A16 | USBP7P |
| A17 | VSS |
| A18 | USBP5P |
| A19 | VSS |
| A20 | USBP3P |
| A21 | VSS |
| A22 | USBP1P |
| A23 | VSS |
| A24 | USBRBIAS |
| AA1 | LAN_RST# |
| AA2 | SMLINK1 |
| AA3 | SLP_S5# |
| AA4 | VCCSUS1_5_B |
| AA5 | VSS |
| AA6 | VCCSATAPLL |
| AA7 | VSS |
| AA8 | SATA0TXP |
| AA9 | VSS |
| AA10 | SATA1TXP |
| AA11 | VSS |
| AA12 | RTCRST# |
| AA13 | VSS |
| AA14 | PDD4 |
| AA15 | PDD11 |
| AA16 | PDD14 |
| AA17 | PDIOW# |

## Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| AA18 | PIORDY |
| AA19 | PDA0 |
| AA20 | SDD11 |
| AA21 | VSS |
| AA22 | SDD0 |
| AA23 | SDD15 |
| AA24 | VSS |
| AB1 | SUS_STAT# |
| AB2 | TP0 |
| AB3 | RI# |
| AB4 | VCCSUS1_5_B |
| AB5 | VSS |
| AB6 | VCCSATAPLL |
| AB7 | VSS |
| AB8 | SATA0TXN |
| AB9 | VSS |
| AB10 | SATA1TXN |
| AB11 | VSS |
| AB12 | RTCX2 |
| AB13 | RSMRST# |
| AB14 | PDD7 |
| AB15 | VSS |
| AB16 | PDD0 |
| AB17 | PDD15 |
| AB18 | VSS |
| AB19 | PDCS1# |
| AB20 | SDD6 |
| AB21 | SDD4 |
| AB22 | SDD12 |
| AB23 | SDD1 |
| AB24 | SDD14 |
| AC1 | GPIO24 |
| AC2 | VSS |
| AC3 | SMBALERT#/GPIO11 |
| AC4 | VSS |
| AC5 | CLK100P |
| AC6 | VSS |
| AC7 | SATA0RXP |
| AC8 | VSS |
| AC9 | SATA1RXP |
| AC10 | VSS |

## Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| AC11 | RTCX1 |
| AC12 | PWROK |
| AC13 | VSS |
| AC14 | PDD3 |
| AC15 | PDD5 |
| AC16 | PDD12 |
| AC17 | PDDREQ |
| AC18 | PDDACK# |
| AC19 | PDA2 |
| AC20 | SDD7 |
| AC21 | SDD5 |
| AC22 | SDD10 |
| AC23 | VSS |
| AC24 | SDD13 |
| AD1 | SMBDATA |
| AD2 | SMBCLK |
| AD3 | SMLINK0 |
| AD4 | VSS |
| AD5 | CLK100N |
| AD6 | VSS |
| AD7 | SATA0RXN |
| AD8 | VSS |
| AD9 | SATA1RXN |
| AD10 | INTVRMEN |
| AD11 | VCCRTC |
| AD12 | VSS |
| AD13 | VCC3_3 |
| AD14 | PDD6 |
| AD15 | PDD8 |
| AD16 | PDD10 |
| AD17 | VSS |
| AD18 | PDIOR# |
| AD19 | PDA1 |
| AD20 | VCC3_3 |
| AD21 | VSS |
| AD22 | SDD9 |
| AD23 | SDD2 |
| AD24 | SDD3 |
| B1 | PIRQH#/GPIO5 |
| B2 | AD18 |
| B3 | PIRQA# |

### Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| B4 | GNTB#/ GNT5#GPIO17 |
| B5 | VCC3_3 |
| B6 | REQ3# |
| B7 | GNT2# |
| B8 | AC_SYNC |
| B9 | EE_DOUT |
| B10 | EE_CS |
| B11 | EE_DIN |
| B12 | LAN_TXD2 |
| B13 | VSS |
| B14 | OC4# / GPIO9 |
| B15 | VCCSUS3_3 |
| B16 | USBP7N |
| B17 | VSS |
| B18 | USBP5N |
| B19 | VSS |
| B20 | USBP3N |
| B21 | VSS |
| B22 | USBP1N |
| B23 | VSS |
| B24 | USBRBIAS# |
| C1 | REQ1# |
| C2 | PIRQD# |
| C3 | VSS |
| C4 | AD22 |
| C5 | REQ2# |
| C6 | REQ4#/GPIO40 |
| C7 | GNT3# |
| C8 | VSS |
| C9 | LAN_RXD1 |
| C10 | LAN_RXD0 |
| C11 | LAN_RXD2 |
| C12 | AC_RST# |
| C13 | OC7# / GPIO15 |
| C14 | OC3# |
| C15 | OC0# |
| C16 | VSS |
| C17 | USBP6P |
| C18 | VSS |
| C19 | USBP4P |

### Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| C20 | VSS |
| C21 | USBP2P |
| C22 | VSS |
| C23 | USBP0P |
| C24 | VCCUSBPLL |
| D1 | VSS |
| D2 | FRAME# |
| D3 | AD26 |
| D4 | GNT0# |
| D5 | REQ0# |
| D6 | VSS |
| D7 | PIRQE#/GPIO2 |
| D8 | AC_BIT_CLK |
| D9 | LAN_TXD0 |
| D10 | LAN_RSTSYNC |
| D11 | VSS |
| D12 | AC_SDIN1 |
| D13 | OC6# / GPIO14 |
| D14 | OC2# |
| D15 | OC1# |
| D16 | VSS |
| D17 | USBP6N |
| D18 | VSS |
| D19 | USBP4N |
| D20 | VSS |
| D21 | USBP2N |
| D22 | VSS |
| D23 | USBP0N |
| D24 | VSS |
| E1 | PIRQB# |
| E2 | PIRQG#/GPIO4 |
| E3 | C/BE0# |
| E4 | TRDY# |
| E5 | STOP# |
| E6 | AD24 |
| E7 | REQB#/REQ5#/ GPIO1 |
| E8 | GNTA#/GPIO16 |
| E9 | LAN_TXD1 |
| E10 | LAN_CLK |
| E11 | VCCSUS3_3 |

### Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| E12 | AC_SDIN0 |
| E13 | VCCSUS3_3 |
| E14 | VCCSUS3_3 |
| E15 | VCC1_5 |
| E16 | V5REF_SUS |
| E17 | VSS |
| E18 | VCCSUS3_3 |
| E19 | VSS |
| E20 | VSS |
| E21 | VSS |
| E22 | VCC1_5 |
| E23 | VSS |
| E24 | SPKR |
| F1 | PAR |
| F2 | AD9 |
| F3 | VSS |
| F4 | AD30 |
| F5 | AD28 |
| F6 | VCC3_3 |
| F7 | VCCSUS1_5_C |
| F8 | VCCSUS1_5_C |
| F9 | VSS |
| F10 | VCCSUS3_3 |
| F11 | VCCSUS3_3 |
| F14 | VCC1_5 |
| F15 | VCC1_5 |
| F16 | VCCSUS3_3 |
| F17 | VCCSUS3_3 |
| F18 | VCCSUS3_3 |
| F19 | VCCSUS1_5_A |
| F20 | CLK14 |
| F21 | GPIO34 |
| F22 | GPIO23 |
| F23 | SERIRQ |
| F24 | CLK48 |
| G1 | VCC3_3 |
| G2 | AD13 |
| G3 | AD2 |
| G4 | AD16 |
| G5 | AD15 |
| G6 | VSS |

### Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| G19 | VCC3_3 |
| G20 | VSS |
| G21 | VCC3_3 |
| G22 | HI11 |
| G23 | SATALED# |
| G24 | VSS |
| H1 | VSS |
| H2 | AD5 |
| H3 | AD20 |
| H4 | AD11 |
| H5 | AD4 |
| H6 | VCC3_3 |
| H19 | VSS |
| H20 | HI0 |
| H21 | HI1 |
| H22 | VSS |
| H23 | HI3 |
| H24 | VCC1_5 |
| J1 | C/BE1# |
| J2 | AD7 |
| J3 | AD6 |
| J4 | AD0 |
| J5 | AD1 |
| J6 | VSS |
| J19 | VCC1_5 |
| J20 | HI2 |
| J21 | VSS |
| J22 | HI9 |
| J23 | VSS |
| J24 | HI_STBS |
| K1 | AD14 |
| K2 | PERR# |
| K3 | VSS |
| K4 | AD3 |
| K5 | AD8 |
| K6 | VCC3_3 |
| K10 | VCC1_5 |
| K11 | VSS |
| K12 | VCC1_5 |
| K13 | VCC1_5 |
| K14 | VSS |

### Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| K15 | VCCSUS3_3 |
| K19 | VCC1_5 |
| K20 | VSS |
| K21 | HI10 |
| K22 | VSS |
| K23 | HI_STBF |
| K24 | VSS |
| L1 | AD17 |
| L2 | PLOCK# |
| L3 | DEVSEL# |
| L4 | SERR# |
| L5 | AD12 |
| L6 | VCC3_3 |
| L10 | VSS |
| L11 | VSS |
| L12 | VSS |
| L13 | VSS |
| L14 | VSS |
| L15 | VSS |
| L19 | VCC1_5 |
| L20 | HI_VSWING |
| L21 | VSS |
| L22 | HI8 |
| L23 | VSS |
| L24 | HIREF |
| M1 | VSS |
| M2 | C/BE3# |
| M3 | IRDY# |
| M4 | AD10 |
| M5 | VSS |
| M10 | VCC3_3 |
| M11 | VSS |
| M12 | VSS |
| M13 | VSS |
| M14 | VSS |
| M15 | VCC1_5 |
| M20 | HI7 |
| M21 | HI5 |
| M22 | VSS |
| M23 | HI4 |
| M24 | VSS |

### Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| N1 | PCICLK |
| N2 | AD27 |
| N3 | C/BE2# |
| N4 | AD23 |
| N5 | AD21 |
| N10 | VCC3_3 |
| N11 | VSS |
| N12 | VSS |
| N13 | VSS |
| N14 | VSS |
| N15 | VCC1_5 |
| N20 | VSS |
| N21 | HI6 |
| N22 | CLK66 |
| N23 | VCC1_5 |
| N24 | HIRCOMP |
| P1 | VSS |
| P2 | AD31 |
| P3 | AD25 |
| P4 | AD29 |
| P5 | AD19 |
| P6 | VCC3_3 |
| P10 | VSS |
| P11 | VSS |
| P12 | VSS |
| P13 | VSS |
| P14 | VSS |
| P15 | VSS |
| P19 | VCC1_5 |
| P20 | TP1 |
| P21 | VSS |
| P22 | CPUSLP# |
| P23 | RCIN# |
| P24 | CPUPWRGD/GPIO49 |
| R1 | GPIO21 |
| R2 | LDRQ1# / GPIO41 |
| R3 | LAD2 |
| R4 | LAD1 |
| R5 | GPIO6 |
| R6 | VCC1_5 |
| R10 | VCC1_5 |

### Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| R11 | VSS |
| R12 | VCC1_5 |
| R13 | VCC3_3 |
| R14 | VSS |
| R15 | V_CPU_IO |
| R19 | V_CPU_IO |
| R20 | VRMPWRGD |
| R21 | IGNNE# |
| R22 | NMI |
| R23 | INIT# |
| R24 | TP2 |
| T1 | GPIO32 |
| T2 | THRM# |
| T3 | VSS |
| T4 | LFRAME# |
| T5 | LAD0 |
| T6 | VSS |
| T19 | V_CPU_IO |
| T20 | GPIO19 |
| T21 | THRMTRIP# |
| T22 | A20GATE |
| T23 | VSS |
| T24 | STPCLK# |
| U1 | SYS_RESET# |
| U2 | SLP_S4# |
| U3 | GPIO7 |
| U4 | LAD3 |
| U5 | LDRQ0# |
| U6 | VCCSUS3_3 |
| U19 | VSS |
| U20 | GPIO22 |
| U21 | GPIO18 |
| U22 | GPIO20 |
| U23 | INTR |
| U24 | FERR# |
| V1 | VSS |
| V2 | PME# |
| V3 | GPIO27 |
| V4 | PCIRST# |
| V5 | LINKALERT# |
| V6 | VCCSUS3_3 |

### Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| V19 | VCC3_3 |
| V20 | SDCS3# |
| V21 | VSS |
| V22 | SDCS1# |
| V23 | A20M# |
| V24 | SMI# |
| W1 | SLP_S3# |
| W2 | GPIO28 |
| W3 | GPIO25 |
| W4 | GPIO12 |
| W5 | GPIO13 |
| W6 | VCC1_5 |
| W7 | VCC1_5 |
| W8 | VCC1_5 |
| W9 | VCC1_5 |
| W10 | VCC1_5 |
| W11 | VCC1_5 |
| W14 | V5REF |
| W15 | VCC3_3 |
| W16 | VSS |
| W17 | VCC3_3 |
| W18 | VSS |
| W19 | VCC1_5 |
| W20 | SDDACK# |
| W21 | SDA2 |
| W22 | SDA0 |
| W23 | SDA1 |
| W24 | VCC3_3 |
| Y1 | SUSCLK |
| Y2 | GPIO8 |
| Y3 | VSS |
| Y4 | PWRBTN# |
| Y5 | VCCSUS1_5_B |
| Y6 | VSS |
| Y7 | VSS |
| Y8 | VSS |
| Y9 | SATARBIAS# |
| Y10 | VSS |
| Y11 | SATARBIAS |
| Y12 | INTRUDER# |
| Y13 | PDD1 |

### Table 176. Intel® ICH5 Ballout by Ball Number

| Ball # | Signal Name |
|---|---|
| Y14 | PDD2 |
| Y15 | PDD9 |
| Y16 | PDD13 |
| Y17 | IRQ14 |
| Y18 | PDCS3# |
| Y19 | SDD8 |
| Y20 | SDDREQ |
| Y21 | SIORDY |
| Y22 | SDIOW# |
| Y23 | SDIOR# |
| Y24 | IRQ15 |

This page is intentionally left blank.

# Electrical Characteristics 19

This chapter contains thermal, DC, and AC characteristics for the ICH5. AC timing diagrams are included.

## 19.1 Thermal Specifications

Refer to the *Intel® 82801EB I/O Controller Hub (ICH5) / Intel® 82801ER I/O Controller Hub 5 R (ICH5R) Thermal Design Guide* for ICH5 thermal information.

## 19.2 DC Characteristics

**Table 177. DC Current Characteristics**

| Power Plane | Maximum Power Consumption | | | | |
|---|---|---|---|---|---|
| | SO | S1 | S3 | S4/S5 | G3 |
| Vcc1_5 Core | 770 mA | 201 mA | N/A | N/A | N/A |
| Vcc3_3 I/O | 480 mA | 1 mA | N/A | N/A | N/A |
| VccSus3_3 (3) | 360 mA | 73 mA | 73 mA | 73 mA | N/A |
| VccRTC | N/A | N/A | N/A | N/A | 6 µA (1,2) |
| V_CPU_IO | 2.5 mA | 2.5 mA | N/A | N/A | N/A |
| V5REF | 250 µa | 250 µa | N/A | N/A | N/A |
| V5REF_Sus | 200 µa | 200 µa | 200 µa | 200 µa | N/A |

1. Only the G3 state for this power well is shown to provide an estimate of battery life.
2. Icc(RTC) data is taken with VccRTC at 3.0 V while the system is in a mechanical off (G3) state at room temperature.
3. Due to the integrated voltage regulator, VccSus1_5 is part of the VccSus3_3 power rail.

### Table 178. DC Characteristic Input Signal Association

| Symbol | Associated Signals |
|---|---|
| $V_{IH1}/V_{IL1}$ (5V Tolerant) | **PC/PCI Signals:** AD[31:0], C/BE[3:0]#, DEVSEL#, FRAME#, IRDY#, TRDY#, STOP#, PAR, PERR#, PLOCK#, SERR#, REQ[4:0]#, REQA#, REQB#/REQ5#<br>**GPIO Signals:** GPIO[40, 1:0] |
| $V_{IH2}/V_{IL2}$ (5V Tolerant) | **Interrupt Signals:** IRQ[15:14], PIRQ[D:A]#, PIRQ[H:E]#/GPIO[5:2]<br>**Legacy Signals:** RCIN#, A20GATE<br>**GPIO Signals:** GPIO[7:6] |
| $V_{IH3}/V_{IL3}$ | **Clock Signals**: CLK66, CLK48, CLK14<br>**Interrupt Signals:** SERIRQ<br>**Power Management Signals:** PME#, PWRBTN#, RI#, LAN_RST#, RTCRST#, SYS_RESET#, THRM#, VRMPWRGD<br>**EEPROM Signals:** EE_DIN<br>**SATA Signals:** SATA_LED#<br>**GPIO Signals:** GPIO[34, 32, 28:27, 25:24, 13:12, 8] |
| $V_{IH4}/V_{IL4}$ | **Clock Signals:** PCICLK<br>**LPC/flash BIOS Signals:** LDRQ[1:0]#, LAD[3:0]/FB[3:0]<br>**GPIO Signals:** GPIO41 |
| $V_{IH5}/V_{IL5}$ | **SMBus Signals:** SMBCLK, SMBDATA<br>**System Management Signals:** INTRUDER#, SMLINK[1:0], SMBALERT#/GPIO11, LINKALERT#<br>**Power Management Signals:** RSMRST#, PWROK |
| $V_{IL6}/V_{IH6}$ | **LAN Signals:** LAN_RXD[2:0], LAN_CLK |
| $V_{IL7}/V_{IH7}$ | **Processor Signals:** FERR#, THRMTRIP# |
| $V_{IL8}/V_{IH8}$ | **Hub Interface Signals**: HI[11:0], HI_STBS, HI_STBF |
| $V_{IL9}/V_{IH9}$ | **Real Time Clock Signals:** RTCX1 |
| $V_{IL10}/V_{IH10}$ | **SATA Signals:** SATA[1:0]RX[P,N] |
| $V_{IL11}/V_{IH11}$ (5V Tolerant) | **USB Signals:** OC[7:0]#<br>**GPIO Signals:** GPIO[15:14, 10:9] |
| $V_{IL12}/V_{IH12}$ | **AC'97 Signals:** AC_BITCLK, AC_SDIN[2:0] |
| $V_{IL13}/V_{IH13}$/ Vcross(abs) | **Clock Signals:** CLK100P, CLK100N |
| V+/V-/VHYS/ VTHRAVG/VRING (5V Tolerant) | **IDE Signals:** PDD[15:0], SDD[15:0], PDDREQ, PIORDY, SDDREQ, SIORDY<br>For Ultra DMA Mode 4 and lower these signals, follow the DC characteristics for $V_{IH2}/V_{IL2}$. |
| $V_{DI}$ / $V_{CM}$ / $V_{SE}$ (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 179. DC Input Characteristics (Sheet 1 of 2)

| Symbol | Parameter | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| $V_{IL1}$ | Input Low Voltage | –0.5 | 0.3Vcc3_3 | V | |
| $V_{IH1}$ | Input High Voltage | 0.5Vcc3_3 | V5REF + 0.5 | V | |
| $V_{IL2}$ | Input Low Voltage | –0.5 | 0.8 | V | |
| $V_{IH2}$ | Input High Voltage | 2.0 | V5REF + 0.5 | V | |
| $V_{IL3}$ | Input Low Voltage | –0.5 | 0.8 | V | |
| $V_{IH3}$ | Input High Voltage | 2.0 | Vcc3_3 + 0.5 | V | |
| $V_{IL4}$ | Input Low Voltage | –0.5 | 0.3Vcc3_3 | V | |
| $V_{IH4}$ | Input High Voltage | 0.5Vcc3_3 | Vcc3_3 + 0.5 | V | |
| $V_{IL5}$ | Input Low Voltage | –0.5 | 0.8 | V | |
| $V_{IH5}$ | Input High Voltage | 2.1 | VccSus3_3 + 0.5 | V | |
| $V_{IL6}$ | Input Low Voltage | –0.5 | 0.3Vcc3_3 | V | |
| $V_{IH6}$ | Input High Voltage | 0.6Vcc3_3 | Vcc3_3 + 0.5 | V | |
| $V_{IL7}$ | Input Low Voltage | –0.15 | 0.58(V_CPU_IO) | V | |
| $V_{IH7}$ | Input High Voltage | 0.73(V_CPU_IO) | V_CPU_IO | V | |
| $V_{IL8}$ | Input Low Voltage | –0.3 | HIREF – 0.10 | V | |
| $V_{IH8}$ | Input High Voltage | HIREF + 0.10 | 1.2 | V | |
| $V_{IL9}$ | Input Low Voltage | –0.5 | 0.10 | V | |
| $V_{IH9}$ | Input High Voltage | 0.40 | 2.0 | V | |
| $V_{IL10}$ | Input Low Voltage | 325 | | mVp-p | 6 |
| $V_{IH10}$ | Input High Voltage | | 600 | mVp-p | 6 |
| $V_{IL11}$ | Input Low Voltage | –0.5 | 0.8 | V | |
| $V_{IH11}$ | Input High Voltage | 2.0 | V5REF_SUS + 0.5 | V | |
| $V_{IL12}$ | Input Low Voltage | –0.5 | 0.35Vcc3_3 | V | |
| $V_{IH12}$ | Input High Voltage | 0.65Vcc3_3 | Vcc3_3 + 0.3 | V | |
| $V_{IL13}$ | Input Low Voltage | –0.150 | 0.150 | V | |
| $V_{IH13}$ | Input High Voltage | 0.660 | 0.850 | V | |
| Vcross(abs) | Absolute Crossing Point | 0.250 | 0.550 | | 7,8 |
| V+ | Low to high input threshold | 1.5 | 2.0 | V | 1 |
| V– | High to low input threshold | 1.0 | 1.5 | V | 1 |
| VHYS | Difference between input thresholds:<br>(V+current value) – (V– current value) | 320 | | mV | 1 |
| VTHRAVG | Average of thresholds:<br>((V+current value) + (V– current value))/2 | 1.3 | 1.7 | V | 1 |
| VRING | AC Voltage at recipient connector | –1 | 6 | V | 1,2 |
| $V_{DI}$ | Differential Input Sensitivity | 0.2 | | V | 3,5 |

### Table 179. DC Input Characteristics (Sheet 2 of 2)

| Symbol | Parameter | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| $V_{CM}$ | Differential Common Mode Range | 0.8 | 2.5 | V | 4,5 |
| $V_{SE}$ | Single-Ended Receiver Threshold | 0.8 | 2.0 | V | 5 |
| $V_{HSSQ}$ | HS Squelch Detection Threshold | 100 | 150 | mV | 5 |
| $V_{HSDSC}$ | HS Disconnect Detection Threshold | 525 | 625 | mV | 5 |
| $V_{HSCM}$ | HS Data Signaling Common Mode Voltage Range | –50 | 500 | mV | 5 |
| $V_{HSSQ}$ | HS Squelch detection threshold | 100 | 150 | mV | 5 |
| $V_{HSDSC}$ | HS disconnect detection threshold | 525 | 625 | mV | 5 |
| $V_{HSCM}$ | HS data signaling common mode voltage range | –50 | 500 | mV | 5 |

**NOTES:**
1. Applies to Ultra DMA Modes greater than Ultra DMA Mode 4.
2. This is an AC Characteristic that represents transient values for these signals.
3. $V_{DI}$ = | USBPx[P] – USBPx[N]
4. Includes $V_{DI}$ range.
5. Applies to High-speed USB 2.0.
6. SATA Vdiff,rx is measured at the SATA connector on the receive side.
7. Crossing voltage is defined as the instantaneous voltage value when the rising edge of CLK100P equals the falling edge of CLK100N.
8. Vhavg is the statistical average of the Vh measured by the oscilloscope

## Table 180. DC Characteristic Output Signal Association

| Symbol | Associated Signals |
|---|---|
| $V_{OH1}/V_{OL1}$ | **IDE Signals:** PDD[15:0], SDD[15:0], PDIOW#/PDSTOP, SDIOW#/SDSTOP, PDIOR#/ PDWSTB/PRDMARDY, SDIOR#/STWSTB/SRDMARDY, PDDACK#, SDDACK#, PDA[2:0], SDA[2:0], PDCS[3,1]#, SDCS[3,1]# |
| $V_{OH2}/V_{OL2}$ | **Processor Signals:** A20M#, CPUPWRGD(1), CPUSLP#, IGNNE#, INIT#, INTR, NMI, SMI#, STPCLK# |
| $V_{OH3}/V_{OL3}$ | **EEPROM Signals:** EE_CS, EE_DOUT, EE_SHCLK |
| $V_{OH4}/V_{OL4}$ | **PCI Signals:** GNT5#/GNTB#/GPIO17, GNTA#/GPIO16, AD[31:0], C/BE[3:0]#, PCIRST#, GNT[4:0]#, PAR, DEVSEL#, PERR#, PLOCK#, STOP#, TRDY#, IRDY#, FRAME#, SERR# (1)<br>**LPC/flash BIOS Signals:** LAD[3:0]/FB[3:0], LFRAME#/FB4<br>**AC'97 Signals:** AC_RST#, AC_SDOUT, AC_SYNC<br>**LAN Signals:** LAN_RSTSYNC, LAN_TXD[2:0] |
| $V_{OL5}/V_{OH5}$ | **SMBus Signals:** SMBCLK (1), SMBDATA (1)<br>**System Management Signals:** LINKALERT#, SMLINK[1:0](1) |
| $V_{OL6}/V_{OH6}$ | **Power Management Signals:** PME#(1), SLP_S3#, SLP_S4#, SLP_S5#, SUS_STAT#, SUSCLK<br>**GPIO Signals:** GPIO[34, 32, 28:27, 25:22, 21:18]<br>**Interrupt Signals:** SERIRQ, PIRQ[D:A]# (1), PIRQ[H:E]#/GPIO[5:2](1)<br>**Other Signals:** SPKR, SATALED# |
| $V_{OL7}/V_{OH7}$ | **USB Signals**: USBP[7:0][P,N] in Low and Full Speed Modes |
| $V_{OL8}/V_{OH8}$ Zpd/Zpu | **Hub Interface Signals**: HI[11:0], HI_STBS, HI_STBF |
| $V_{OL9}/V_{OH9}$ | **SATA Signals:** SATA[1:0]TX[P,N] |
| $V_{HSOI}$<br>$V_{HSOH}$<br>$V_{HSOL}$<br>$V_{CHIRPJ}$<br>$V_{CHIRPK}$ | **USB Signals:** USBP[7:0][P:N] in High Speed Modes |

**NOTE:**
1. These signals are open drain.

## Table 181. DC Output Characteristics

| Symbol | Parameter | Min | Max | Unit | $I_{OL}$ / $I_{OH}$ | Notes |
|---|---|---|---|---|---|---|
| $V_{OL1}$ | Output Low Voltage | | 0.51 | V | 6 mA | |
| $V_{OH1}$ | Output High Voltage | Vcc3_3 – 0.51 | | V | –6 mA | |
| $V_{OL2}$ | Output Low Voltage | –0.15 | .25(V_CPU_IO) | V | 1.5 mA | |
| $V_{OH2}$ | Output High Voltage | 0.9(V_CPU_IO) | | V | Note 3 | 1 |
| $V_{OL3}$ | Output Low Voltage | | 0.4 | V | 6 mA | |
| $V_{OH3}$ | Output High Voltage | 2.4 | | V | –1 mA | 1 |
| $V_{OL4}$ | Output Low Voltage | | 0.1Vcc3_3 | V | 6 mA | |
| $V_{OH4}$ | Output High Voltage | 0.9Vcc3_3 | | V | –0.5 mA | 1 |
| $V_{OL5}$ | Output Low Voltage | | 0.4 | V | 4 mA | |
| $V_{OH5}$ | Output High Voltage | N/A | | V | N/A | 1 |
| $V_{OL6}$ | Output Low Voltage | | 0.4 | V | 4 mA | |
| $V_{OH6}$ | Output High Voltage | Vcc3_3 – 0.5 | | V | –2 mA | 1 |
| $V_{OL7}$ | Output Low Voltage | | 0.4 | V | 5 mA | |
| $V_{OH7}$ | Output High Voltage | Vcc3_3 – 0.5 | | V | –2 mA | |
| $V_{OL8}$ | Output Low Voltage | | 0.05 | V | 0.5 mA | |
| $V_{OH8}$ | Output High Voltage | 0.750 | .850 | V | –12 mA | |
| $V_{OL9}$ | Output Low Voltage | 400 | | mVp-p | | 2 |
| $V_{OH9}$ | Output High Voltage | | 600 | mVp-p | | 2 |
| Zpd | Pull Down Impedance | 48 | | Ohm | | |
| Zpu | Pull Up Impedance | 46 | | Ohm | | |
| $V_{HSOI}$ | 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 | | |

**NOTES:**
1. The CPUPWRGD, SERR#, PIRQ[A:H], GPIO22, SMBDATA, SMBCLK, LINKALERT#, and SMLINK[1:0] signal has an open drain driver, and the $V_{OH}$ spec does not apply. This signal must have external pull up resistor.
2. SATA Vdiff,tx is measured at the SATA connector on the transmit side
3. $I_{OH2}$ = - (V_CPU_IO - 0.9) * e-3

## Table 182. Other DC Characteristics

| Symbol | Parameter | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| V5REF | Intel® ICH5 Core Well Reference Voltage | 4.75 | 5.25 | V | |
| Vcc3_3 | I/O Buffer Voltage | 3.135 | 3.465 | V | |
| Vcc1_5, VccPLL | Internal Logic Voltage | 1.425 | 1.575 | V | |
| HIREF | Hub Interface Reference Voltage | 0.343 | 0.357 | V | Note 1, 3 |
| HIVSWING | Hub Interface Voltage Swing (Input to HI_VSWING pin) | 0.784 | 0.816 | V | Note 1,4 |
| V5REF_Sus | Suspend Well Reference Voltage | 4.75 | 5.25 | V | |
| VccSus3_3 | Suspend Well I/O Buffer Voltage | 3.135 | 3.465 | V | |
| VccSus1_5 | Suspend Well Logic Voltage | 1.425 | 1.575 | V | |
| VccRTC | Battery Voltage | 1.0 | 3.6 | V | |
| $V_{IT+}$ | Hysteresis Input Rising Threshold | 1.9 | | V | Applied to USBP[7:0][P,N] |
| $V_{IT-}$ | Hysteresis Input Falling Threshold | | 1.3 | V | Applied to USBP[7:0]P,N] |
| $V_{DI}$ | 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 < $V_{IN}$ < 5V) |
| $I_{LI2}$ | PCI_3V Hi-Z State Data Line Leakage | –10 | 10 | µA | (0 V < $V_{IN}$ < 3.3V) |
| $I_{LI3}$ | 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 | Note 2 |
| $C_{IN}$ | Input Capacitance – Hub interface<br>Input Capacitance – All Other | | 8<br>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 |
| | | Typical Value | | | |
| $C_L$ | XTAL1 | 6 | | pF | |
| $C_L$ | XTAL2 | 6 | | pF | |

**NOTES:**
1. HIREF and HI_VSWING are derived from 1.5 V which is the nominal core voltage for the ICH5. Voltage supply tolerance for a particular interface driver voltage must be within a 5% range of nominal.
2. Includes CLK14, CLK48, CLK66, LAN_CLK and PCICLK
3. Nominal value of HIREF is 0.350 V. The spec is at nominal Vcc1_5. Note that HIREF will vary linearly with Vcc1_5, and so Vcc1_5 variation (± 5%) must be accounted for in the HIREF spec in addition to the 2% variation of HIREF in the table.
4. Nominal value of HIVSWING is 0.800 V. The spec is at nominal Vcc1_5. Note that HIVSWING will vary linearly with Vcc1_5, and so Vcc1_5 variation (± 5%) must be accounted for in the HIVSWING spec in addition to the 2% variation of HIVSWING in the table.

# 19.3 AC Characteristics

### Table 183. Clock Timings (Sheet 1 of 2)

| Sym | Parameter | Min | Max | Unit | Notes | Figure |
|---|---|---|---|---|---|---|
| **PCI Clock (PCICLK)** | | | | | | |
| t1 | Period | 30 | 33.3 | ns | | 3 |
| t2 | High Time | 12 | | ns | | 3 |
| t3 | Low Time | 12 | | ns | | 3 |
| t4 | Rise Time | | 3 | ns | | 3 |
| t5 | Fall Time | | 3 | ns | | 3 |
| **Oscillator Clock (OSC)** | | | | | | |
| t6 | Period | 67 | 70 | ns | | 3 |
| t7 | High Time | 20 | | | | 3 |
| t8 | Low Time | 20 | | ns | | 3 |
| **USB Clock (USBCLK)** | | | | | | |
| $f_{clk48}$ | Operating Frequency | 48 | | MHz | 1 | |
| t9 | Frequency Tolerance | | 500 | ppm | 2 | |
| t10 | High Time | 7 | | ns | | 3 |
| t11 | Low Time | 7 | | ns | | 3 |
| t12 | Rise Time | | 1.2 | ns | | 3 |
| t13 | Fall Time | | 1.2 | ns | | 3 |
| **SMBus Clock (SMBCLK)** | | | | | | |
| $f_{smb}$ | Operating Frequency | 10 | 16 | KHz | | |
| t18 | High time | 4.0 | 50 | us | 3 | 18 |
| t19 | Low time | 4.7 | | us | | 18 |
| t20 | Rise time | | 1000 | ns | | 18 |
| t21 | Fall time | | 300 | ns | | 18 |

## Table 183. Clock Timings (Sheet 2 of 2)

| Sym | Parameter | Min | Max | Unit | Notes | Figure |
|---|---|---|---|---|---|---|
| **AC'97 Clock (BITCLK)** | | | | | | |
| $f_{ac97}$ | Operating Frequency | 12.288 | | MHz | | |
| t26 | Output Jitter | | 750 | ps | | |
| t27 | High time | 32.56 | 48.84 | ns | | 3 |
| t28 | Low time | 32.56 | 48.84 | ns | | 3 |
| t29 | Rise time | 2.0 | 6.0 | ns | 4 | 3 |
| t30 | Fall time | 2.0 | 6.0 | ns | 4 | 3 |
| **Hub Interface Clock** | | | | | | |
| $f_{hi}$ | Operating Frequency | 66 | | MHz | | |
| t31 | High time | 6.0 | | ns | | 3 |
| t32 | Low time | 6.0 | | ns | | 3 |
| t33 | Rise time | 0.25 | 1.2 | ns | | 3 |
| t34 | Fall time | 0.25 | 1.2 | ns | | 3 |
| t35 | CLK66 leads PCICLK | 1.0 | 4.5 | ns | 5 | |
| **SATA Clock (CLk100P, CLK100N)** | | | | | | |
| t36 | Period | 9.997 | 10.003 | ns | | |
| t37 | Rise time | 175 | 700 | ps | | |
| t38 | Fall time | 175 | 700 | ps | | |
| **Suspend Clock (SUSCLK)** | | | | | | |
| $f_{susclk}$ | Operating Frequency | 32 | | kHz | 6 | |
| t39 | High Time | 10 | | us | 6 | |
| t39b | Low Time | 10 | | us | 6 | |

**NOTES:**
1. The USBCLK is a 48 MHz that expects a 40/60% duty cycle.
2. USBCLK is a pass-thru clock that is not altered by the ICH5. This frequency tolerance specification is required for USB 2.0 compliance and is affected by external elements such as the clock generator and the system board.
3. The maximum high time (t18 Max) provide a simple guaranteed method for devices to detect bus idle conditions.
4. BITCLK Rise and Fall times are measured from 10%VDD and 90%VDD.
5. This specification includes pin-to-pin skew from the clock generator as well as board skew.
6. SUSCLK duty cycle can range from 30% minimum to 70% maximum.

### Table 184. PCI Interface Timing

| Sym | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| t40 | AD[31:0] Valid Delay | 2 | 11 | ns | Min: 0 pF<br>Max: 50 pF<br>Note 1 | 4 |
| t41 | AD[31:0] Setup Time to PCICLK Rising | 7 | | ns | | 5 |
| t42 | AD[31:0] Hold Time from PCICLK Rising | 0 | | ns | | 5 |
| t43 | C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PAR, PERR#, PLOCK#, DEVSEL# Valid Delay from PCICLK Rising | 2 | 11 | ns | Min: 0 pF<br>Max: 50 pF | 4 |
| t44 | C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PAR, PERR#, PLOCK#, IDSEL, DEVSEL# Output Enable Delay from PCICLK Rising | 2 | | ns | | 8 |
| t45 | C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PERR#, PLOCK#, DEVSEL#, GNT[A:B]# Float Delay from PCICLK Rising | 2 | 28 | ns | | 6 |
| t46 | C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, SERR#, PERR#, DEVSEL#, Setup Time to PCICLK Rising | 7 | | ns | | 5 |
| t47 | C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, SERR#, PERR#, DEVSEL#, REQ[A:B]# Hold Time from PCLKIN Rising | 0 | | ns | | 5 |
| t48 | PCIRST# Low Pulse Width | 1 | | ms | | 7 |
| t49 | GNT[A:B}#, GNT[5:0]# Valid Delay from PCICLK Rising | 2 | 12 | ns | | |
| t50 | REQ[A:B]#, REQ[5:0]# Setup Timer to PCICLK Rising | 12 | | ns | | |

**NOTES:**
1. Refer to *PCI Local Bus Specification, Revision 2.3.*

### Table 185. IDE PIO and Multiword DMA ModeTiming

| Sym | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| t60 | PDIOR#/PDIOW#/SDIOR#/SDIOW# Active From CLK66 Rising | 2 | 20 | ns | | 9<br>10 |
| t61 | PDIOR#/PDIOW#/SDIOR#/SDIOW# Inactive From CLK66 Rising | 2 | 20 | ns | | 9<br>10 |
| t62 | PDA[2:0]/SDA[2:0] Valid Delay From CLK66 Rising | 2 | 30 | ns | | 9 |
| t63 | PDCS1#/SDCS1#, PDCS3#/SDCS3# Active From CLK66 Rising | 2 | 30 | ns | | 9 |
| t64 | PDCS1#/SDCS1#, PDCS3#/SDCS3# Inactive From CLK66 Rising | 2 | 30 | ns | | 9 |
| t65 | PDDACK#/SDDACK# Active From CLK66 Rising | 2 | 20 | ns | | 10 |
| t66 | PDDACK#/SDDACK# Inactive From CLK66 Rising | 2 | 20 | ns | | |
| t67 | PDDREQ/SDDREQ Setup Time to CLK66 Rising | 7 | | ns | | 10 |
| t68 | PDDREQ/SDDREQ Hold From CLK66 Rising | 7 | | ns | | 10 |
| t69 | PDD[15:0]/SDD[15:0] Valid Delay From CLK66 Rising | 2 | 30 | ns | | 9<br>10 |
| t70 | PDD[15:0]/SDD[15:0] Setup Time to CLK66 Rising | 10 | | ns | | 9<br>10 |
| t71 | PDD[15:0]/SDD[15:0] Hold From CLK66 Rising | 7 | | ns | | 9<br>10 |
| t72 | PIORDY/SIORDY Setup Time to CLK66 Rising | 7 | | ns | 1 | 9 |
| t73 | PIORDY/SIORDY Hold From CLK66 Rising | 7 | | ns | 1 | 9 |
| t74 | PIORDY/SIORDY Inactive Pulse Width | 48 | | ns | | 9 |
| t75 | PDIOR#/PDIOW#/SDIOR#/SDIOW# Pulse Width Low | | | | 2,3 | 9<br>10 |
| t76 | PDIOR#/PDIOW#/SDIOR#/SDIOW# Pulse Width High | | | | 3,4 | 9<br>10 |

**NOTES:**
1. IORDY is internally synchronized. This timing is to guarantee recognition on the next clock.
2. PIORDY sample point from DIOx# assertion and PDIOx# active pulse width is programmable from 2–5 PCI clocks when the drive mode is Mode 2 or greater. Refer to the ISP field in the IDE Timing Register.
3. PIORDY sample point from DIOx# assertion, PDIOx# active pulse width and PDIOx# inactive pulse width cycle time is the compatible timing when the drive mode is Mode 0/1. Refer to the TIM0/1 field in the IDE timing register.
4. PDIOx# inactive pulse width is programmable from 1–4 PCI clocks when the drive mode is Mode 2 or greater. Refer to the RCT field in the IDE Timing Register.

### Table 186. Ultra ATA Timing (Mode 0, Mode 1, Mode 2) (Sheet 1 of 2)

| Sym | Parameter (1) | Mode 0 (ns) | | Mode 1 (ns) | | Mode 2 (ns) | | Measuring Location | Figure |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| t80 | Sustained Cycle Time (T2cyctyp) | 240 | | 160 | | 120 | | Sender Connector | |
| t81 | Cycle Time (Tcyc) | 112 | | 73 | | 54 | | End Recipient Connector | 12 |
| t82 | Two Cycle Time (T2cyc) | 230 | | 153 | | 115 | | Sender Connector | 12 |
| t83a | Data Setup Time (Tds) | 15 | | 10 | | 7 | | Recipient Connector | 12 |
| t83b | Recipient IC data setup time (from data valid until STROBE edge) (see Note 2) (Tdsic) | 14.7 | | 9.7 | | 6.8 | | Intel® ICH5 ball | |
| t84a | Data Hold Time (Tdh) | 5 | | 5 | | 5 | | Recipient Connector | 12 |
| t84b | Recipient IC data hold time (from STROBE edge until data may become invalid) (see Note 2) (Tdhic) | 4.8 | | 4.8 | | 4.8 | | ICH5 ball | |
| t85a | Data Valid Setup Time (Tdvs) | 70 | | 48 | | 31 | | Sender Connector | 12 |
| t85b | Sender IC data valid setup time (from data valid until STROBE edge) (see Note 2) (Tdvsic) | 72.9 | | 50.9 | | 33.9 | | ICH5 ball | |
| t86a | Data Valid Hold Time (Tdvh) | 6.2 | | 6.2 | | 6.2 | | Sender Connector | 12 |
| t86b | Sender IC data valid hold time (from STROBE edge until data may become invalid) (see Note 2) (Tdvhic) | 9 | | 9 | | 9 | | ICH5 ball | |
| t87 | Limited Interlock Time (Tli) | 0 | 150 | 0 | 150 | 0 | 150 | See Note 2 | 14 |
| t88 | Interlock Time w/ Minimum (Tmli) | 20 | | 20 | | 20 | | Host Connector | 14 |
| t89 | Envelope Time (Tenv) | 20 | 70 | 20 | 70 | 20 | 70 | Host Connector | 11 |
| t90 | Ready to Pause Time (Trp) | 160 | | 125 | | 100 | | Recipient Connector | 13 |
| t91 | DMACK setup/hold Time (Tack) | 20 | | 20 | | 20 | | Host Connector | 11, 14 |
| t92a | CRC Word Setup Time at Host (Tcvs) | 70 | | 48 | | 31 | | Host Connector | |
| t92b | CRC word valid hold time at sender (from DMACK# negation until CRC may become invalid) (see Note 2) (Tcvh) | 6.2 | | 6.2 | | 6.2 | | Host Connector | |

## Table 186. Ultra ATA Timing (Mode 0, Mode 1, Mode 2) (Sheet 2 of 2)

| Sym | Parameter (1) | Mode 0 (ns) | | Mode 1 (ns) | | Mode 2 (ns) | | Measuring Location | Figure |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| t93 | STROBE output released-to-driving to the first transition of critical timing (Tzfs) | 0 | | 0 | | 0 | | Device Connector | 14 |
| t94 | Data Output Released-to-Driving Until the First Tunisian of Critical Timing (Tdzfs) | 70 | | 48 | | 31 | | Sender Connector | 11 |
| t95 | Unlimited Interlock Time (Tui) | 0 | | 0 | | 0 | | Host Connector | 11 |
| t96a | Maximum time allowed for output drivers to release (from asserted or negated) (Taz) | | 10 | | 10 | | 10 | See Note 2 | |
| t96b | Minimum time for drivers to assert or negate (from released) (Tzad) | 0 | | 0 | | 0 | | Device Connector | |
| t97 | Ready-to-final-STROBE time (no STROBE edges shall be sent this long after negation of DMARDY#) (Trfs) | | 75 | | 70 | | 60 | Sender Connector | 11 |
| t98a | Maximum time before releasing IORDY (Tiordyz) | | 20 | | 20 | | 20 | Device Connector | |
| t98b | Minimum time before driving IORDY (see Note 2) (Tziordy) | 0 | | 0 | | 0 | | Device Connector | |
| t99 | Time from STROBE edge to negation of DMARQ or assertion of STOP (when sender terminates a burst) (Tss) | 50 | | 50 | | 50 | | Sender Connector | 13 |

**NOTES:**
1. The specification symbols in parentheses correspond to the AT Attachment – 6 with Packet Interface (ATA/ATAPI – 6) specification name.
2. See the AT Attachment – 6 with Packet Interface (ATA/ATAPI – 6) specification for further details on measuring these timing parameters.

## Table 187. Ultra ATA Timing (Mode 3, Mode 4, Mode 5) (Sheet 1 of 2)

| Sym | Parameter (1) | Mode 3 (ns) | | Mode 4 (ns) | | Mode 5 (ns) | | Measuring Location | Figure |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| t80 | Sustained Cycle Time (T2cyctyp) | 90 | | 60 | | 40 | | Sender Connector | |
| t81 | Cycle Time (Tcyc) | 39 | | 25 | | 16.8 | | End Recipient Connector | 12 |
| t82 | Two Cycle Time (T2cyc) | 86 | | 57 | | 38 | | Sender Connector | 12 |
| t83 | Data Setup Time (Tds) | 7 | | 5 | | 4.0 | | Recipient Connector | 12 |
| t83b | Recipient IC data setup time (from data valid until STROBE edge) (see Note 2) (Tdsic) | 6.8 | | 4.8 | | 2.3 | | Intel® ICH5 Balls | |
| t84 | Data Hold Time (Tdh) | 5 | | 5 | | 4.6 | | Recipient Connector | 12 |
| t84b | Recipient IC data hold time (from STROBE edge until data may become invalid) (see Note 2) (Tdhic) | 4.8 | | 4.8 | | 2.8 | | ICH5 Balls | |
| t85 | Data Valid Setup Time (Tdvs) | 20 | | 6.7 | | 4.8 | | Sender Connector | 11<br>12 |
| t85b | Sender IC data valid setup time (from data valid until STROBE edge) (see Note 2) (Tdvsic) | 22.6 | | 9.5 | | 6.0 | | ICH5 Balls | |
| t86 | Data Valid Hold Time (Tdvh) | 6.2 | | 6.2 | | 4.8 | | Sender Connector | 11<br>12 |
| t86b | Sender IC data valid hold time (from STROBE edge until data may become invalid) (see Note 2) (Tdvhic) | 9.0 | | 9.0 | | 6.0 | | ICH5 Balls | |
| t87 | Limited Interlock Time (Tli) | 0 | 100 | 0 | 100 | 0 | 75 | See Note 2 | 14 |
| t88 | Interlock Time w/ Minimum (Tmli) | 20 | | 20 | | 20 | | Host Connector | 14 |
| t89 | Envelope Time (Tenv) | 20 | 55 | 20 | 55 | 20 | 50 | Host Connector | 12 |
| t90 | Ready to Pause Time (Trp) | 100 | | 100 | | 85 | | Recipient Connector | 13 |
| t91 | DMACK setup/hold Time (Tack) | 20 | | 20 | | 20 | | Host Connector | 14 |
| t92a | CRC Word Setup Time at Host (Tcvs) | 20 | | 6.7 | | 10 | | Host Connector | |
| t92b | CRC Word Hold Time at Sender<br>CRC word valid hold time at sender (from DMACK# negation until CRC may become invalid) (see Note 2) (Tcvh) | 6.2 | | 6.2 | | 10.0 | | Host Connector | |

## Table 187. Ultra ATA Timing (Mode 3, Mode 4, Mode 5) (Sheet 2 of 2)

| Sym | Parameter (1) | Mode 3 (ns) | | Mode 4 (ns) | | Mode 5 (ns) | | Measuring Location | Figure |
|---|---|---|---|---|---|---|---|---|---|
| | | Min | Max | Min | Max | Min | Max | | |
| t93 | STROBE output released-to-driving to the first transition of critical timing (Tzfs) | 0 | | 0 | | 35 | | Device Connector | 14 |
| t94 | Data Output Released-to-Driving Until the First Transition of Critical Timing (Tdzfs) | 20.0 | | 6.7 | | 25 | | Sender Connector | |
| t95 | Unlimited Interlock Time (Tui) | 0 | | 0 | | 0 | | Host Connector | |
| t96a | Maximum time allowed for output drivers to release (from asserted or negated) (Taz) | | 10 | | 10 | | 10 | See Note 2 | |
| t96b | Drivers to assert or negate (from released) (Tzad) | 0 | | 0 | | 0 | | Device Connector | |
| t97 | Ready-to-final-STROBE time (no STROBE edges shall be sent this long after negation of DMARDY#) (Trfs) | | 60 | | 60 | | 50 | Sender Connector | |
| t98a | Maximum time before releasing IORDY (Tiordyz) | | 20 | | 20 | | 20 | Device Connector | |
| t98b | Minimum time before driving IORDY (see Note 2) (Tziordy) | 0 | | 0 | | 0 | | Device Connector | |
| t99 | Time from STROBE edge to negation of DMARQ or assertion of STOP (when sender terminates a burst) (Tss) | 50 | | 50 | | 50 | | Sender Connector | 13 |

**NOTES:**
1. The specification symbols in parentheses correspond to the AT Attachment – 6 with Packet Interface (ATA/ATAPI – 6) specification name.
2. See the AT Attachment – 6 with Packet Interface (ATA/ATAPI – 6) specification for further details on measuring these timing parameters.

## Table 188. Universal Serial Bus Timing

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| **Full Speed Source (Note 7)** | | | | | | |
| t100 | USBPx+, USBPx- Driver Rise Time | 4 | 20 | ns | 1, $C_L$ = 50 pF | 15 |
| t101 | USBPx+, USBPx- Driver Fall Time | 4 | 20 | ns | 1, $C_L$ = 50 pF | 15 |
| t102 | Source Differential Driver Jitter<br>To Next Transition<br>For Paired Transitions | <br>–3.5<br>–4 | <br>3.5<br>4 | <br>ns<br>ns | 2, 3 | 16 |
| t103 | Source SE0 interval of EOP | 160 | 175 | ns | 4 | 17 |
| t104 | Source Jitter for Differential Transition to SE0 Transition | –2 | 5 | ns | 5 | |
| t105 | Receiver Data Jitter Tolerance<br>To Next Transition<br>For Paired Transitions | <br>–18.5<br>–9 | <br>18.5<br>9 | <br>ns<br>ns | 3 | 16 |
| t106 | EOP Width: Must accept as EOP | 82 | | ns | 4 | 17 |
| t107 | Width of SE0 interval during differential transition | | 14 | ns | | |
| **Low Speed Source (Note 8)** | | | | | | |
| t108 | USBPx+, USBPx – Driver Rise Time | 75 | 300 | ns | 1, 6<br>$C_L$ = 50 pF<br>$C_L$ = 350 pF | 15 |
| t109 | USBPx+, USBPx – Driver Fall Time | 75 | 300 | ns | 1,6<br>$C_L$ = 50 pF<br>$C_L$ = 350 pF | 15 |
| t110 | Source Differential Driver Jitter<br>To Next Transition<br>For Paired Transitions | <br>–25<br>–14 | <br>25<br>14 | <br>ns<br>ns | 2, 3 | 16 |
| t111 | Source SE0 interval of EOP | 1.25 | 1.50 | µs | 4 | 17 |
| t112 | Source Jitter for Differential Transition to SE0 Transition | –40 | 100 | ns | 5 | |
| t113 | Receiver Data Jitter Tolerance<br>To Next Transition<br>For Paired Transitions | <br>–152<br>–200 | <br>152<br>200 | <br>ns<br>ns | 3 | 16 |
| t114 | EOP Width: Must accept as EOP | 670 | | ns | 4 | 17 |
| t115 | Width of SE0 interval during differential transition | | 210 | ns | | |

**NOTES:**
1. Driver output resistance under steady state drive is spec'd at 28 ohms at minimum and 43 ohms at maximum.
2. Timing difference between the differential data signals.
3. Measured at crossover point of differential data signals.
4. Measured at 50% swing point of data signals.
5. Measured from last crossover point to 50% swing point of data line at leading edge of EOP.
6. Measured from 10% to 90% of the data signal.
7. Full Speed Data Rate has minimum of 11.97 Mbps and maximum of 12.03 Mbps.
8. Low Speed Data Rate has a minimum of 1.48 Mbps and a maximum of 1.52 Mbps.

### Table 189. SATA Interface Timings

| Sym | Parameter | Min | Max | Units | Notes | Figure |
|---|---|---|---|---|---|---|
| UI | Operating Data Period | 666.43 | 670.12 | ps | | |
| | Rise Time | 0.2 | 0.41 | UI | 1 | |
| | Fall Time | 0.2 | 0.41 | UI | 2 | |
| | TX differential skew | | 20 | ps | | |
| | COMRESET | 310.4 | 329.6 | ns | 3 | |
| | COMWAKE transmit spacing | 103.5 | 109.9 | ns | 3 | |
| | OOB Operating Data period | 646.67 | 686.67 | ns | 4 | |

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

### Table 190. SMBus Timing

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t130 | Bus Tree Time Between Stop and Start Condition | 4.7 | | µs | | 18 |
| t131 | Hold Time after (repeated) Start Condition. After this period, the first clock is generated. | 4.0 | | µs | | 18 |
| t132 | Repeated Start Condition Setup Time | 4.7 | | µs | | 18 |
| t133 | Stop Condition Setup Time | 4.0 | | µs | | 18 |
| t134 | Data Hold Time | 0 | | ns | 4 | 18 |
| t135 | Data Setup Time | 250 | | ns | | 18 |
| t136 | Device Time Out | 25 | 35 | ms | 1 | |
| t137 | Cumulative Clock Low Extend Time (slave device) | | 25 | ms | 2 | 19 |
| t138 | Cumulative Clock Low Extend Time (master device) | | 10 | ms | 3 | 19 |

NOTES:
1. A device will timeout when any clock low exceeds this value.
2. t137 is the cumulative time a slave device is allowed to extend the clock cycles in one message from the initial start to stop. If a slave device exceeds this time, it is expected to release both its clock and data lines and reset itself.
3. t138 is the cumulative time a master device is allowed to extend its clock cycles within each byte of a message as defined from start-to-ack, ack-to-ack or ack-to-stop.
4. t134 has a minimum timing for I2C of 0 ns, while the minimum timing for SMBus is 300 ns.

### Table 191. AC'97 Timing

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t140 | ACSDIN[2:0] Setup to Falling Edge of BITCLK | 10 | | ns | | 24 |
| t141 | ACSDIN[2:0] Hold from Falling Edge of BITCLK | 10 | | ns | | 24 |
| t142 | ACSYNC, ACSDOUTvalid delay from rising edge of BITCLK | | 15 | ns | | 24 |

### Table 192. LPC Timing

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t150 | LAD[3:0] Valid Delay from PCICLK Rising | 2 | 11 | ns | | 4 |
| t151 | LAD[3:0] Output Enable Delay from PCICLK Rising | 2 | | ns | | 8 |
| t152 | LAD[3:0] Float Delay from PCICLK Rising | | 28 | ns | | 6 |
| t153 | LAD[3:0] Setup Time to PCICLK Rising | 7 | | ns | | 5 |
| t154 | LAD[3:0] Hold Time from PCICLK Rising | 0 | | ns | | 5 |
| t155 | LDRQ[1:0]# Setup Time to PCICLK Rising | 12 | | ns | | 5 |
| t156 | LDRQ[1:0]# Hold Time from PCICLK Rising | 0 | | ns | | 5 |
| t157 | LFRAME# Valid Delay from PCICLK Rising | 2 | 12 | ns | | 4 |

### Table 193. Miscellaneous Timings

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t160 | SERIRQ Setup Time to PCICLK Rising | 7 | | ns | | 5 |
| t161 | SERIRQ Hold Time from PCICLK Rising | 0 | | ns | | 5 |
| t162 | RI#, EXTSMI#, GPI, USB Resume Pulse Width | 2 | | RTCCLK | | 7 |
| t163 | SPKR Valid Delay from OSC Rising | | 200 | ns | | 4 |
| t164 | SERR# Active to NMI Active | | 200 | ns | | |
| t165 | IGNNE# Inactive from FERR# Inactive | | 230 | ns | | |

### Table 194. Power Sequencing and Reset Signal Timings

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t170 | VccRTC active to RTCRST# inactive | 5 | – | ms | | 20 |
| t171 | V5RefSus active to VccSus3_3, VccSus1_5_x active | 0 | – | ms | 1, 2 | 20 |
| t172 | VccRTC supply active to VccSus supplies active | 0 | – | ms | 3 | 20 |
| t173 | VccSus supplies active to LAN_RST# active, RSMRST# inactive | 10 | – | ms | | 20<br>21 |
| t174 | V5Ref active to Vcc3_3, Vcc1_5 active | 0 | – | ms | 1, 2 | 20 |
| t175 | VccSus supplies active to Vcc supplies active | 0 | – | ms | 3 | 20 |
| t176 | Vcc supplies active to PWROK, VRMPWRGD active | 99 | – | ms | | 20<br>21<br>23 |
| t177 | PWROK and VRMPWRGD active to SUS_STAT# inactive and Frequency straps at appropriate value | 32 | 38 | RTCCLK | 4 | 21<br>23 |
| t178 | SUS_STAT# inactive to PCIRST# inactive | 2 | 3 | RTCCLK | | 21<br>23 |
| t179 | AC_RST# active low pulse width | 1 | | us | | |
| t180 | AC_RST# inactive to AC_BIT_CLK startup delay | 162.8 | | ns | | |

**NOTES:**
1. The V5Ref supply must power up before or simultaneous with its associated 3.3 V supply, and must power down simultaneous with or after the 3.3 V supply. See Section 2.21.3.1 for details.
2. The associated 3.3 V and 1.5 V supplies are assumed to power up or down ‘together’. VccSus3_3 must ramp up with or before VccSus1_5_x and VccSus3_3 must power down after VccSus1_5.
3. The VccSus supplies must **never** be active while the VccRTC supply is inactive. Likewise, the Vcc supplies must **never** be active while the VccSus supplies are inactive.
4. SYSRESET# is not checked for triggering t177.

## Table 195. Power Management Timings

| Sym | Parameter | Min | Max | Units | Notes | Fig |
|---|---|---|---|---|---|---|
| t181 | VccSus active to SLP_S5#, SUS_STAT# and PCIRST# active | | 50 | ns | | 21 |
| t182<br>t183 | RSMRST# inactive to SUSCLK running, SLP_S5# inactive | | 110 | ms | 7 | 21 |
| t183a | SLPS5# inactive to SLP_S4# inactive | 1 | 2 | RTCCLK | | 21 |
| t183b | SLPS4# inactive to SLP_S3# inactive | 1 | 2 | RTCCLK | | 21 |
| t184 | Vcc active to STPCLK# and CPUSLP# inactive, and Processor Frequency Strap signals high | | 50 | ns | | 21<br>23 |
| t185 | PWROK and VRMPWRGD active and SYS_RESET# inactive to SUS_STAT# inactive and Processor Frequency Straps latched to Strap Values | 32 | 38 | RTCCLK | 1 | 21 |
| t186 | Processor Reset Complete to Frequency Strap signals unlatched from Strap Values | 7 | 9 | CLK66 | 2 | 21 |
| t187 | STPCLK# active to Stop Grant cycle | N/A | N/A | | 3 | 22 |
| t188 | Stop Grant cycle to CPUSLP# active | 60 | 63 | PCICLK | 4 | 22<br>23 |
| t189 | S1 Wake Event to CPUSLP# inactive | 1 | 25 | PCICLK | 4 | 22 |
| t190 | CPUSLP# inactive to STPCLK# inactive | 3.87 | 245 | µs | | 22 |
| t192 | CPUSLP# active to SUS_STAT# active | 2 | 4 | RTCCLK | 1 | 23 |
| t193 | SUS_STAT# active to PCIRST# active | 9 | 21 | RTCCLK | 1 | |
| t194 | PCIRST# active to SLP_S3# active | 1 | 2 | RTCCLK | 1 | 23 |
| t194a | SLP_S3# active to SLP_S4# active | 1 | 2 | RTCCLK | 1 | 23 |
| t195 | SLP_S4# active to SLP_S5# active | 1 | 2 | RTCCLK | 1, 6 | 23 |
| t196 | SLP_S3# active to PWROK, VRMPWRGD inactive | 0 | | ms | 5 | 23 |
| t197 | PWROK, VRMPWRGD inactive to Vcc supplies inactive | 20 | | ns | | 23 |
| t198 | Wake Event to SLP_S5# inactive | 1 | 10 | RTCCLK | 1 | |
| t198a | Wake Event to SLP_S4# inactive(S4 Wake) | 1 | 10 | RTCCLK | 1 | |
| t198b | S3 Wake Event to SLP_S3# inactive(S3 Wake) | 0 | 2 | RTCCLK | 1 | |
| t198d | SLP_S5# inactive or S4 Wake Event to SLP_S4# inactive | See Note Below | | | 9 | 23 |
| t198e | SLP_S4# inactive to SLP_S3# inactive | 1 | 2 | RTCCLK | 1 | 23 |
| t220 | THRMTRIP# active to SLP_S3#, SLP_S4#, SLP_S5# active | | 3 | PCI CLK | | |

**NOTES:**
1. These transitions are clocked off the internal RTC. 1 RTC clock is approximately 32 µs.
2. This transition is clocked off the 66 MHz CLK66. 1 CLK66 is approximately 15 ns.
3. The ICH5 STPCLK# assertion will trigger the processor to send a stop grant acknowledge cycle. The timing for this cycle getting to the ICH5 is dependant on the processor and the memory controller.
4. These transitions are clocked off the 33 MHz PCICLK. 1 PCICLK is approximately 30ns.
5. The ICH5 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.
6. If the transition to S5 is due to Power Button Override, SLP_S3#, SLP_S4# and SLP_S5# are asserted together similar to timing t194 (PCIRST# active to SLP_S3# active).
7. 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.
8. This value is programmable in multiples of 1024 PCI CLKs. Maximum is 8192 PCI CLKs (245.6 µs).
9. For timing t198d, 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).

## 19.4 Timing Diagrams

**Figure 3. Clock Timing**

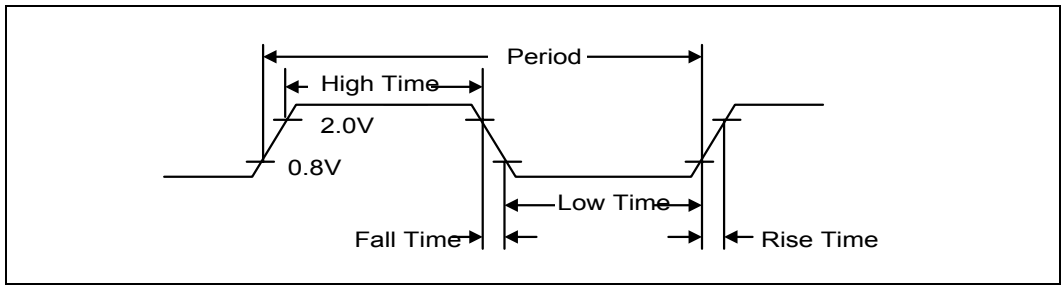



**Figure 4. Valid Delay from Rising Clock Edge**

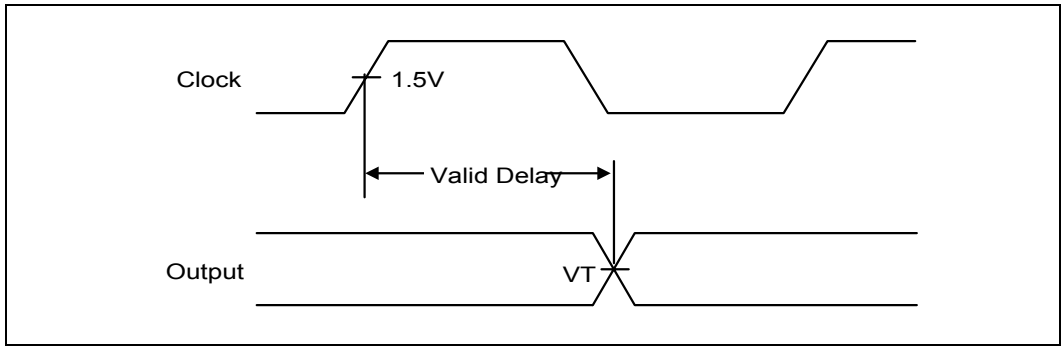



**Figure 5. Setup and Hold Times**

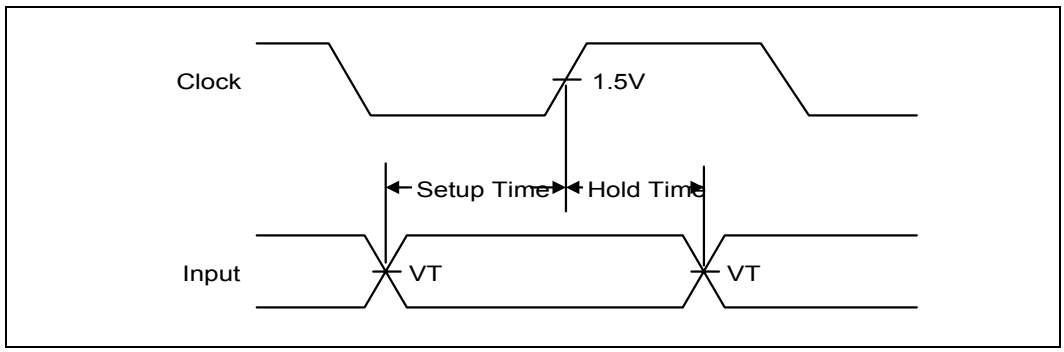


**Figure 6. Float Delay**

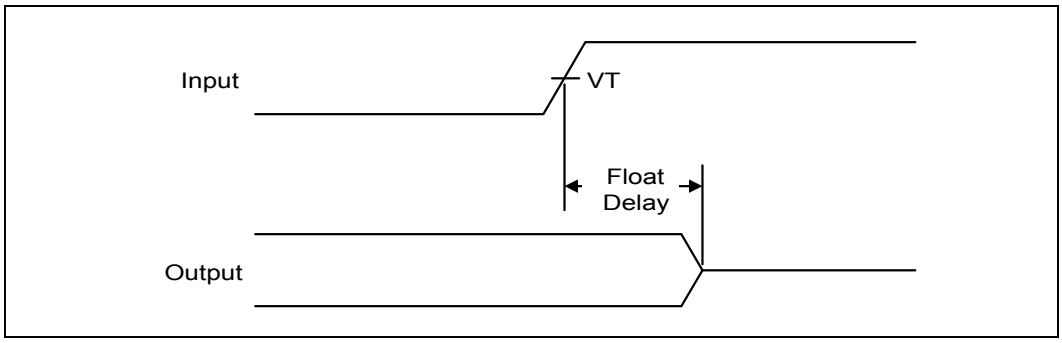



**Figure 7. Pulse Width**

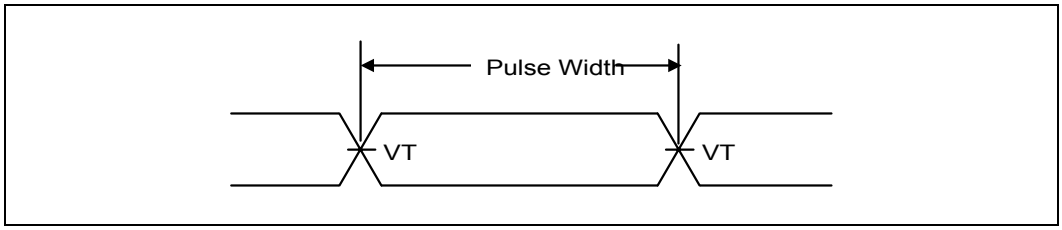



**Figure 8. Output Enable Delay**

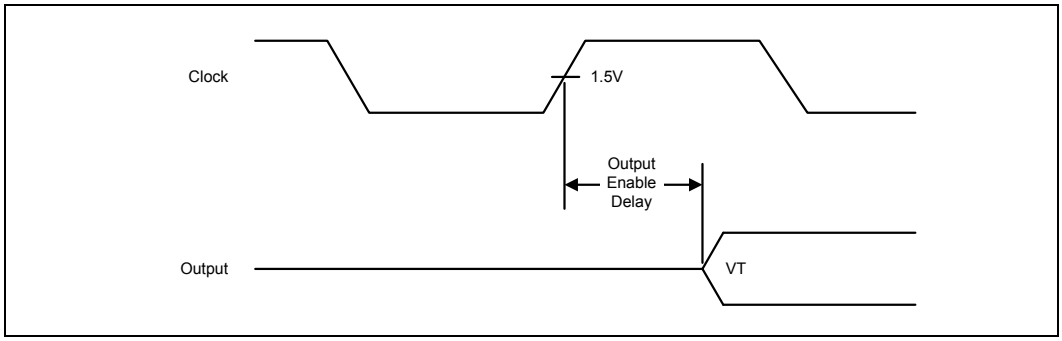


**Figure 9. IDE PIO Mode**

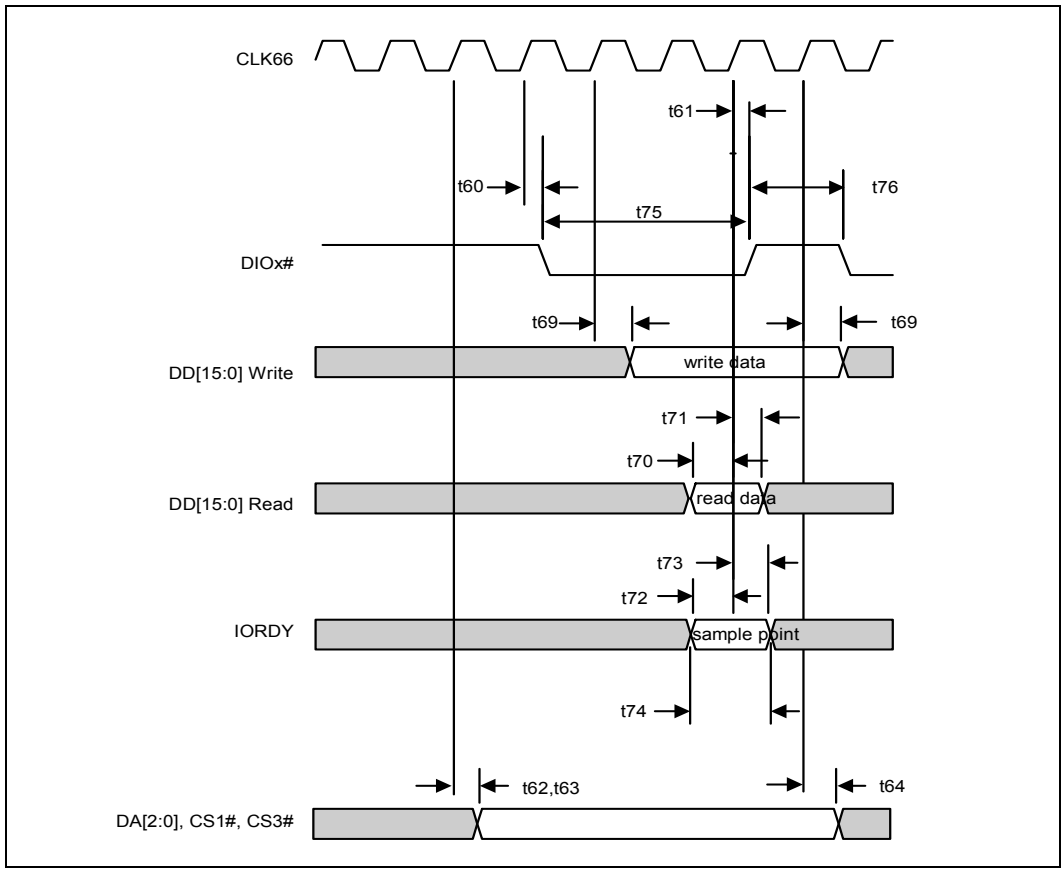



**Figure 10. IDE Multiword DMA**

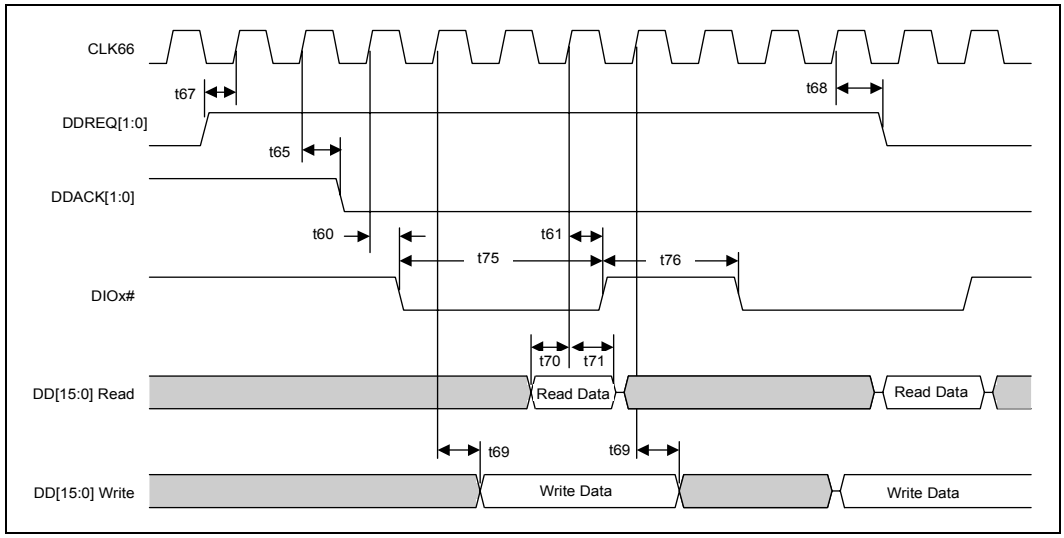


**Figure 11. Ultra ATA Mode (Drive Initiating a Burst Read)**

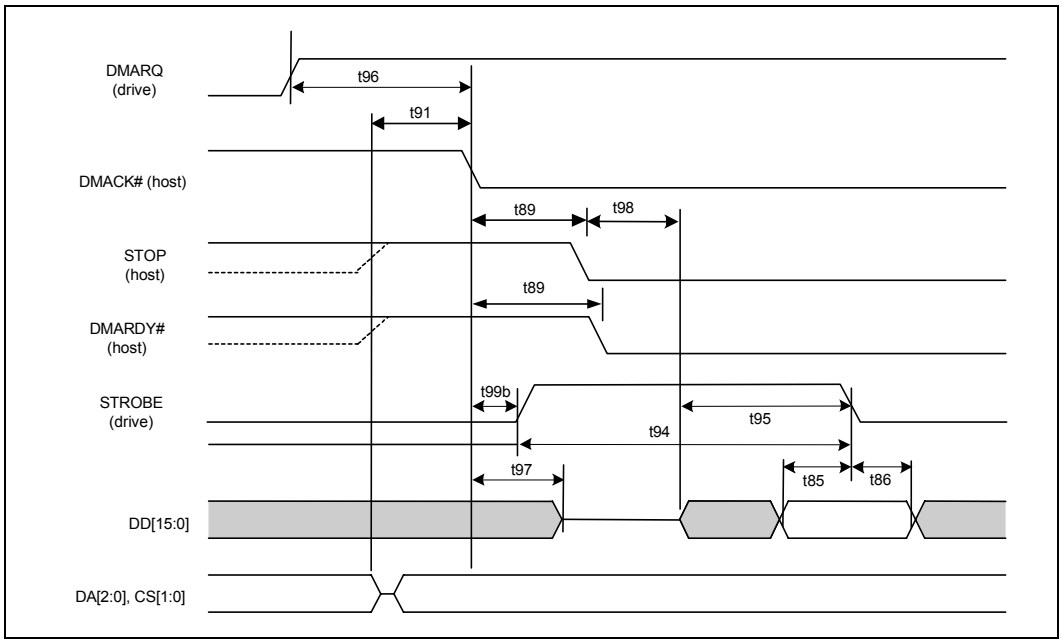



**Figure 12. Ultra ATA Mode (Sustained Burst)**

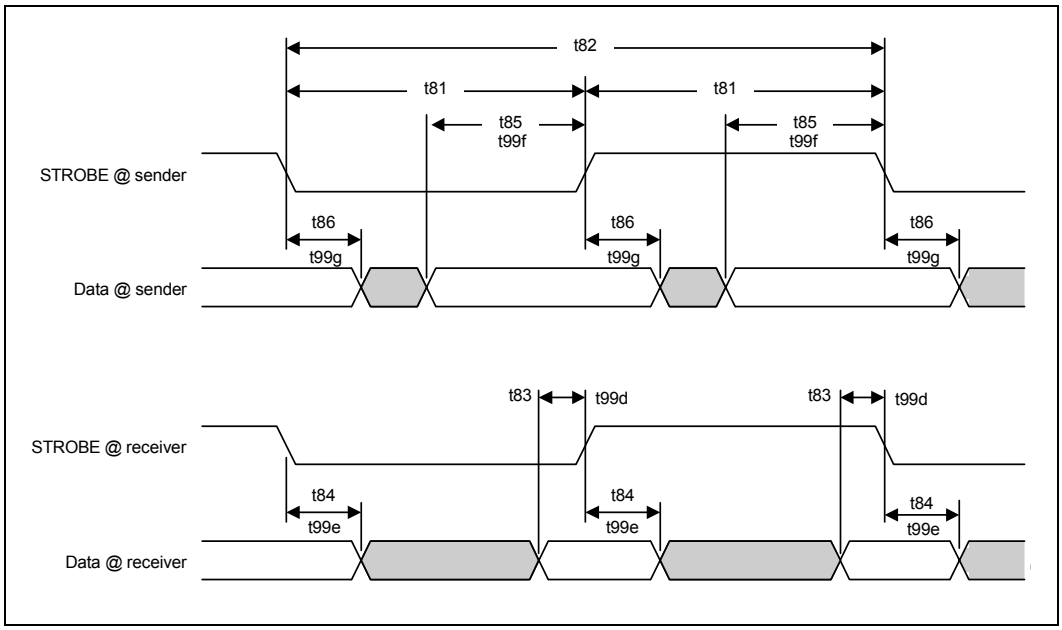


**Figure 13. Ultra ATA Mode (Pausing a DMA Burst)**

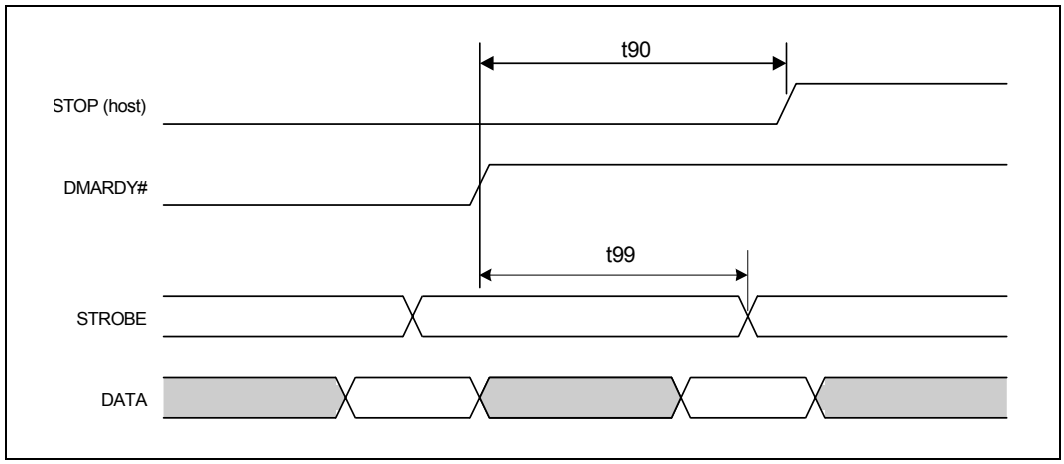



**Figure 14. Ultra ATA Mode (Terminating a DMA Burst)**

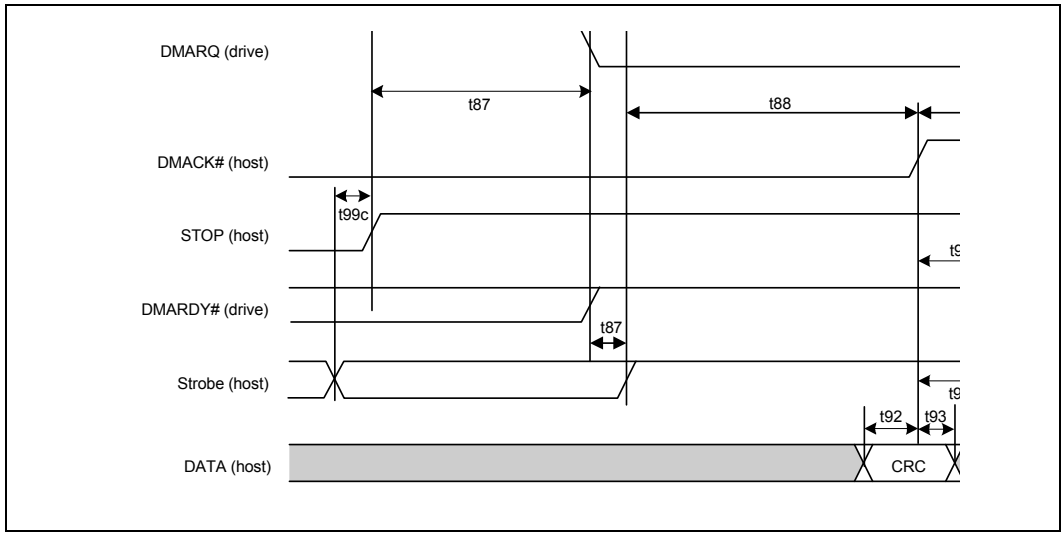



**Figure 15. USB Rise and Fall Times**

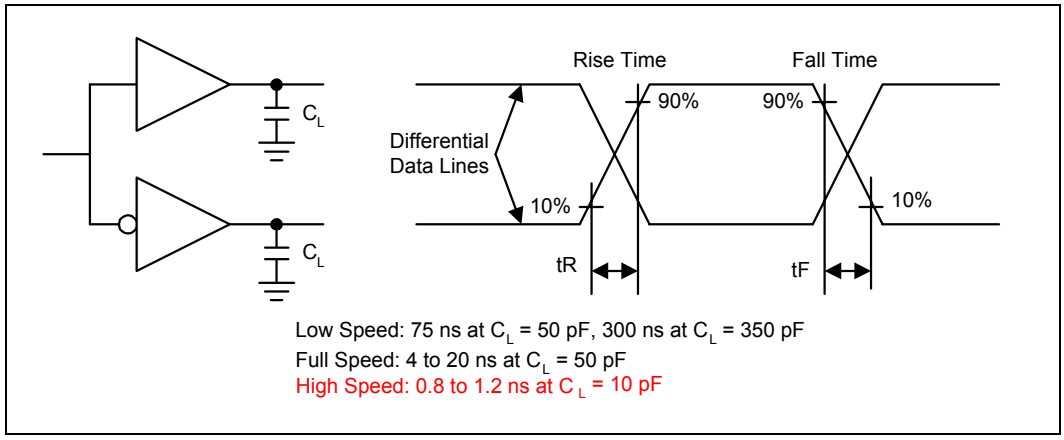


**Figure 16. USB Jitter**

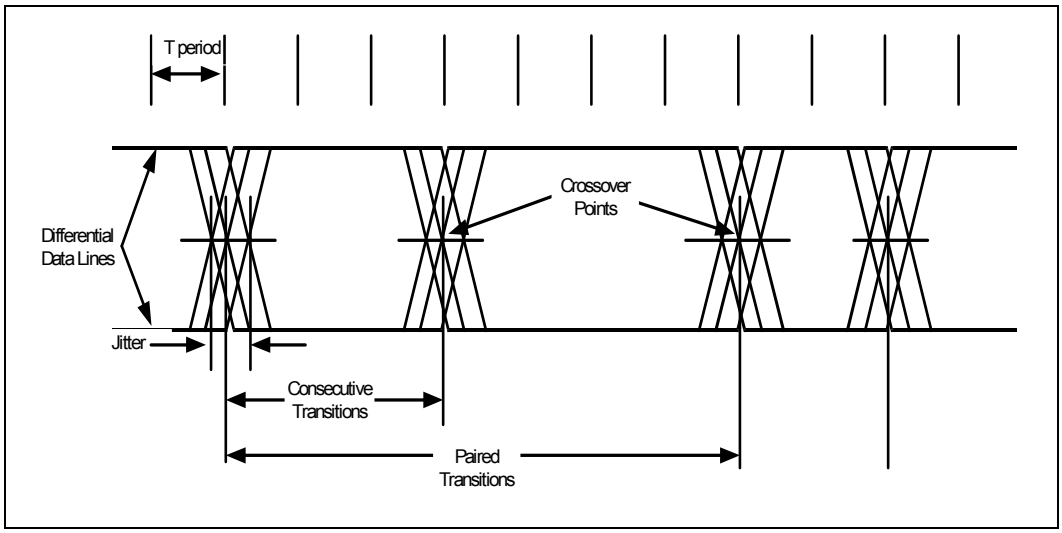



**Figure 17. USB EOP Width**

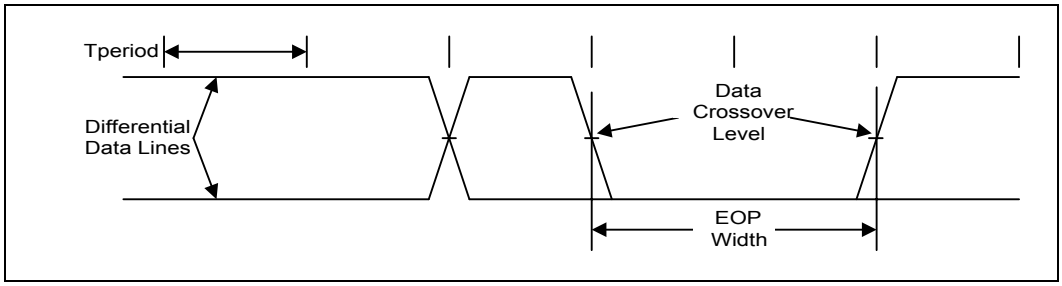



**Figure 18. SMBus Transaction**

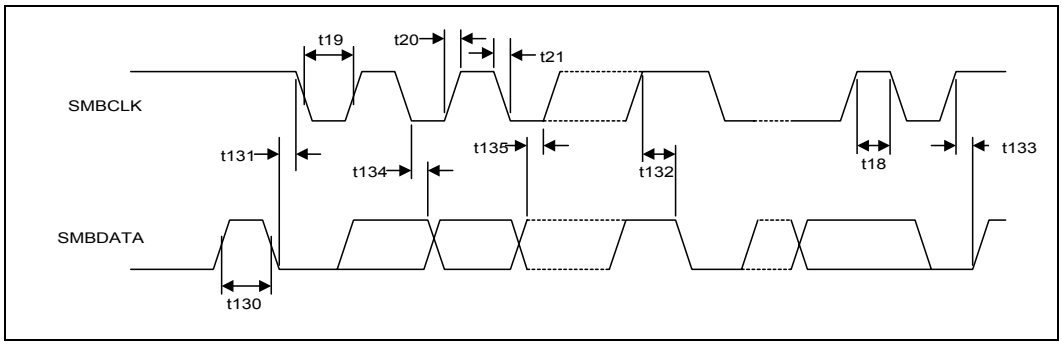


**Figure 19. SMBus Timeout**

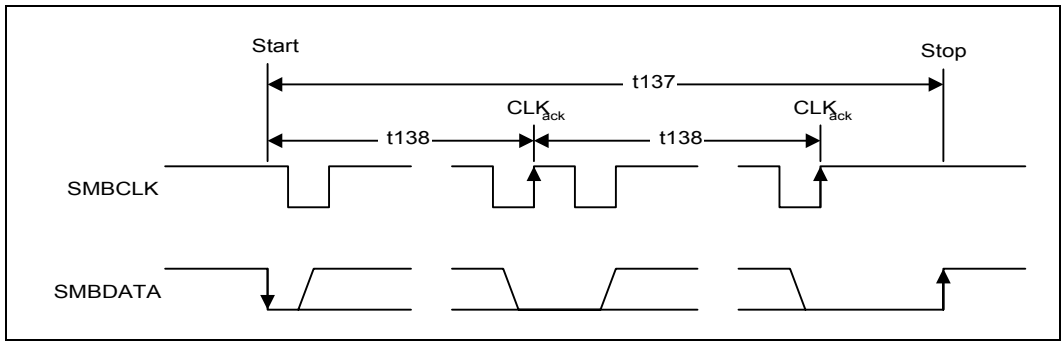



**Figure 20. Power Sequencing and Reset Signal Timings**

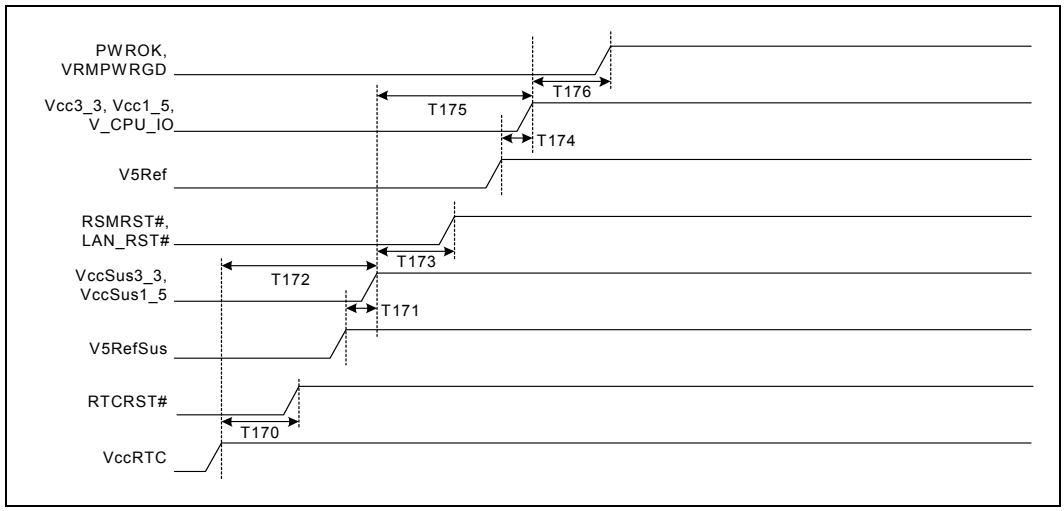


**Figure 21. G3 (Mechanical Off) to S0 Timings**

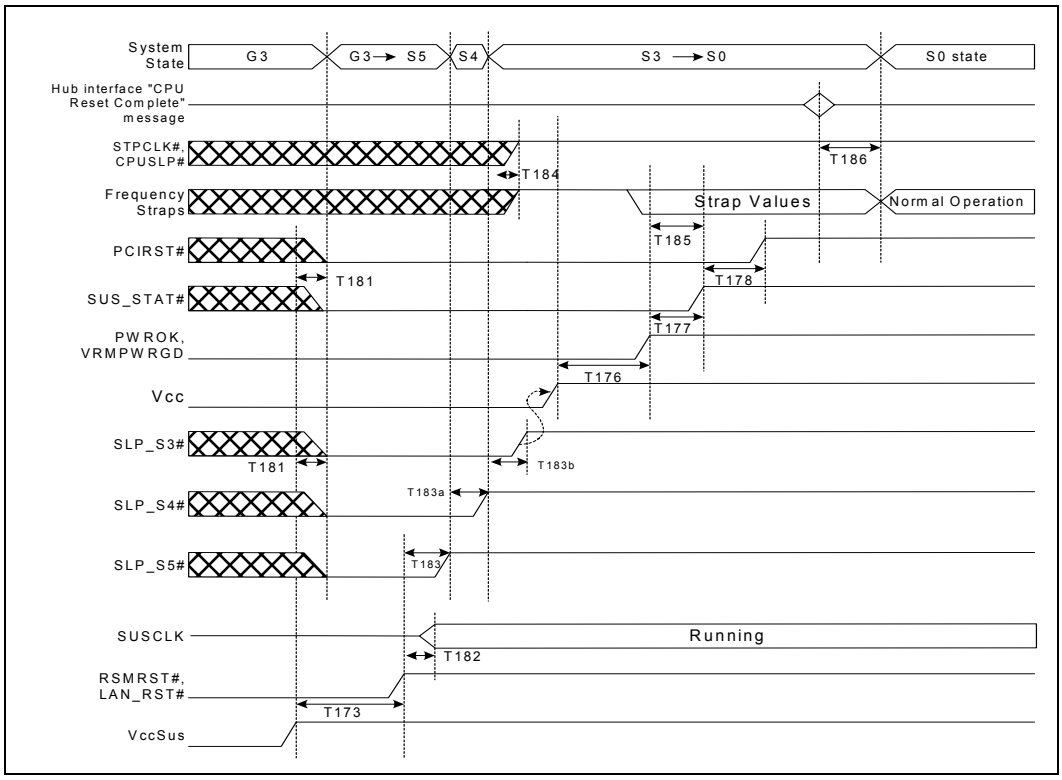



**Figure 22. S0 to S1 to S0 Timing**

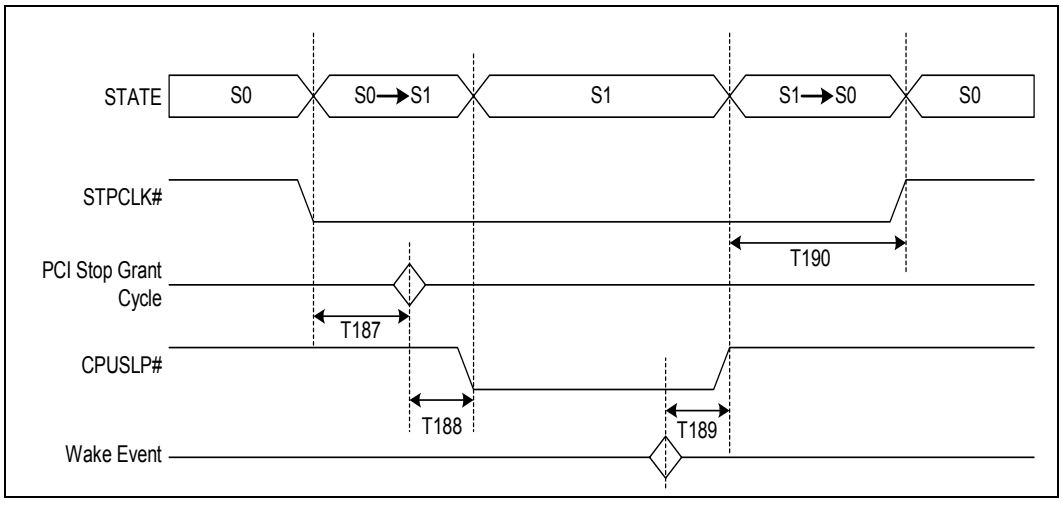
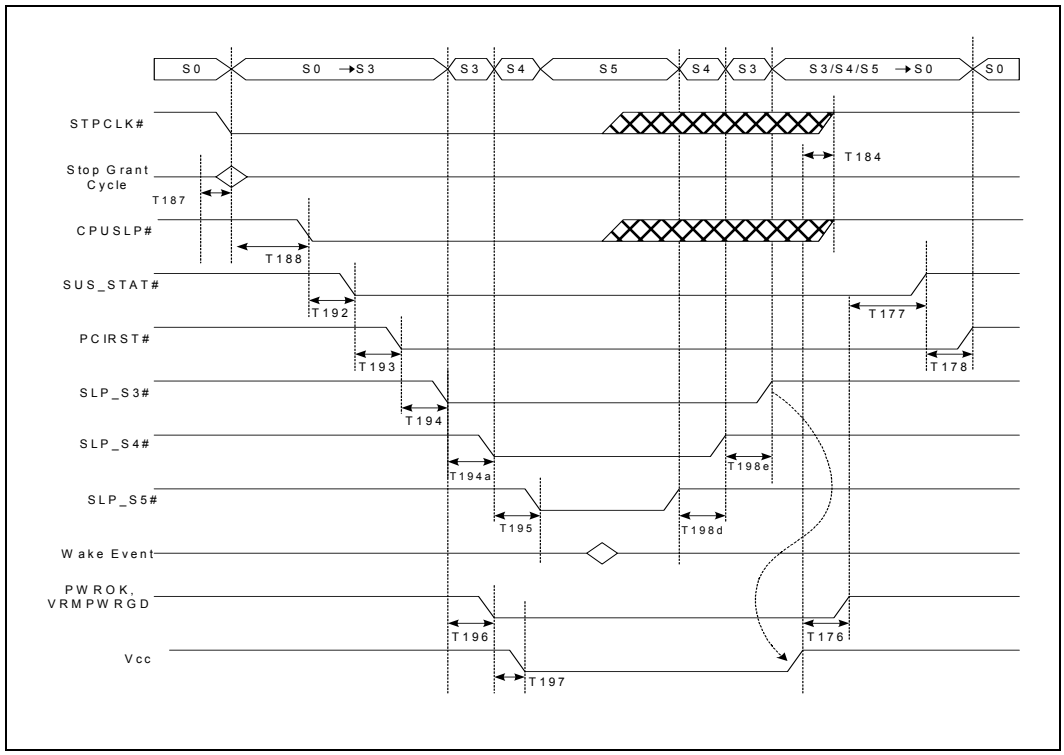
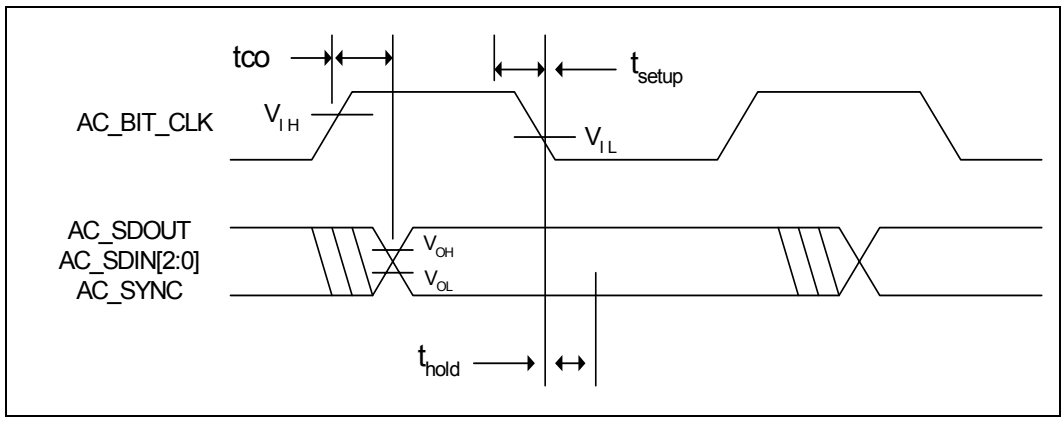


**Figure 23. S0 to S5 to S0 Timings**




***Note:*** T198d - Refer to Table 195 note #9 for SLP_S4# assertion width timing details.

**Figure 24. AC'97 Data Input and Output Timings**

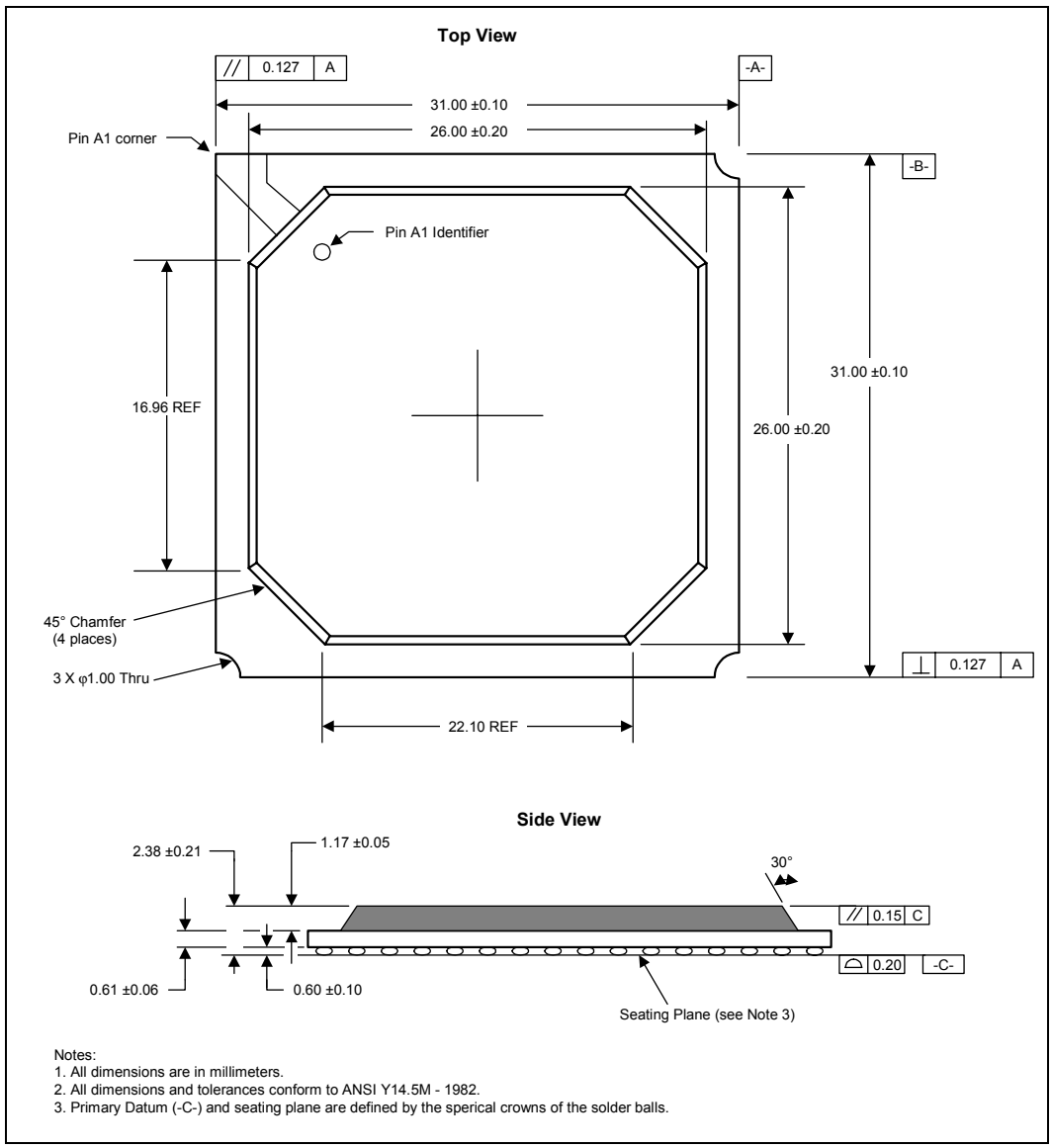


This page is intentionally left blank.

# Package Information 20

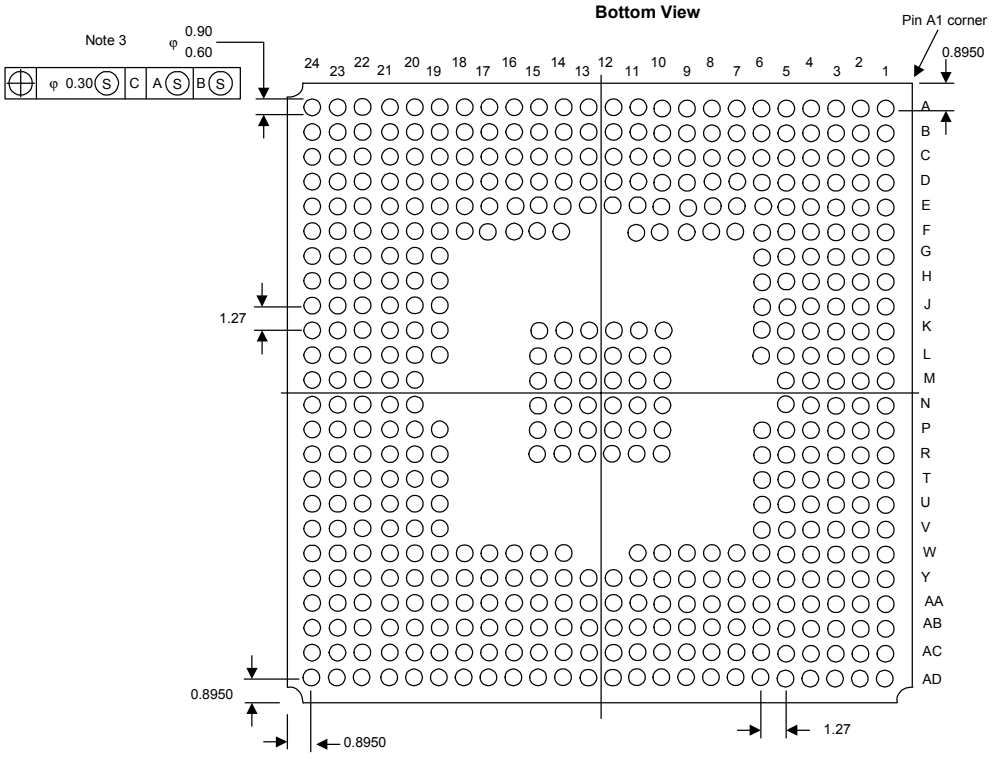
The ICH5 package information is shown in Figure 25 and Figure 26.

**Figure 25. Intel® ICH5 Package (Top and Side Views)**




Notes:
1. All dimensions are in millimeters.
2. All dimensions and tolerances conform to ANSI Y14.5M - 1982.
3. Primary Datum (-C-) and seating plane are defined by the spherical crowns of the solder balls.

## Figure 26. Intel® ICH5 Package (Bottom View)




Notes:
1. All dimensions are in millimeters.
2. All dimensions and tolerances conform to ANSI Y14.5M - 1982.
3. Dimension is measured at the maximum solder ball diameter. Parallel to Datum (-C-) on Side View illustration.

# Testability 21

## 21.1 Test Mode Description

The ICH5 supports two types of test modes, a tri-state test mode and a XOR Chain test mode. Driving RTCRST# low for a specific number of PCI clocks while PWROK is high will activate a particular test mode as described in Table 196.

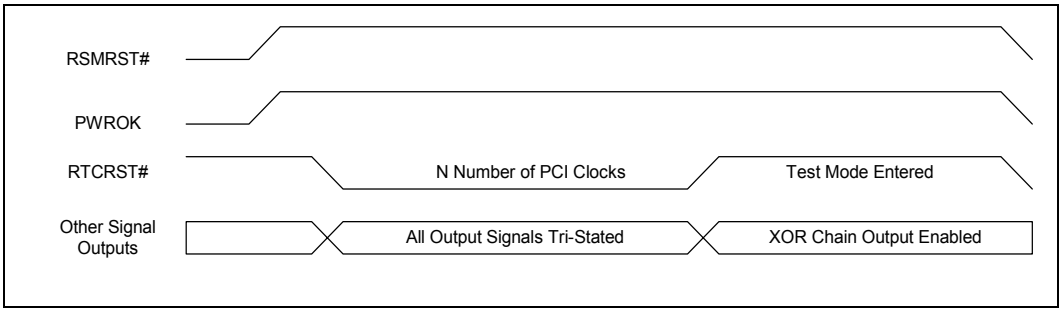
***Note:*** RTCRST# can be driven low any time after PCIRST# is inactive.

**Table 196. Test Mode Selection**

| Number of PCI Clocks RTCRST# driven low after PWROK active | Test Mode |
|---|---|
| <4 | No Test Mode Selected |
| 4 | XOR Chain 1 |
| 5 | XOR Chain 2 |
| 6 | XOR Chain 3 |
| 7 | XOR Chain 4 |
| 8 | All “Z” |
| 9–42 | Reserved. DO NOT ATTEMPT |
| 43–51 | No Test Mode Selected |
| 52 | XOR Chain 6 |
| 53 | XOR Chain 4 Bandgap |
| >53 | No Test Mode Selected |

Figure 27 illustrates the entry into a test mode. A particular test mode is entered upon the rising edge of the RTCRST# after being asserted for a specific number of PCI clocks while PWROK is active. To change test modes, the same sequence should be followed again. To restore the ICH5 to normal operation, execute the sequence with RTCRST# being asserted so that no test mode is selected as specified in Table 196.

**Figure 27. Test Mode Entry (XOR Chain Example)**



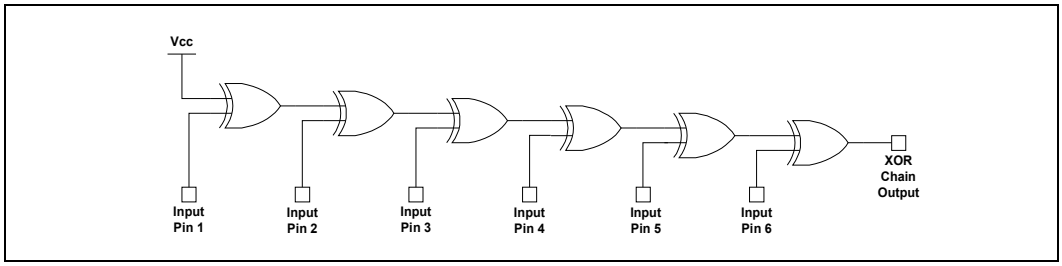
## 21.2 Tri-State Mode

When in the tri-state mode, all outputs and bi-directional pin are tri-stated, including the XOR Chain outputs.

## 21.3 XOR Chain Mode

In the ICH5, provisions for Automated Test Equipment (ATE) board level testing are implemented with XOR Chains. The ICH5 signals are grouped into four independent XOR chains which are enabled individually. When an XOR chain is enabled, all output and bi-directional buffers within that chain are tri-stated, except for the XOR chain output. Every signal in the enabled XOR chain (except for the XOR chain's output) functions as an input. All output and bi-directional buffers for pins not in the selected XOR chain are tri-stated. Figure 28 is a schematic example of XOR chain circuitry.

**Figure 28. Example XOR Chain Circuitry**




### 21.3.1 XOR Chain Testability Algorithm Example

XOR chain testing allows motherboard manufacturers to check component connectivity (e.g., opens and shorts to VCC or GND). An example algorithm to do this is shown in Table 197.

**Table 197. XOR Test Pattern Example**

| Vector | Input Pin 1 | Input Pin 2 | Input Pin 3 | Input Pin 4 | Input Pin 5 | Input Pin 6 | XOR Output |
|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | 0 | 0 | 0 | **1** |
| 2 | 1 | 0 | 0 | 0 | 0 | 0 | **0** |
| 3 | 1 | 1 | 0 | 0 | 0 | 0 | **1** |
| 4 | 1 | 1 | 1 | 0 | 0 | 0 | **0** |
| 5 | 1 | 1 | 1 | 1 | 0 | 0 | **1** |
| 6 | 1 | 1 | 1 | 1 | 1 | 0 | **0** |
| 7 | 1 | 1 | 1 | 1 | 1 | 1 | **1** |

In this example, Vector 1 applies all 0s to the chain inputs. The outputs being non-inverting will consistently produce a 1 at the XOR output on a good board. One short to VCC (or open floating to VCC) will result in a 0 at the chain output, signaling a defect.

Likewise, applying Vector 7 (all 1s) to the chain inputs (given that there are an even number of input signals in the chain), will consistently produce a 1 at the XOR chain output on a good board. One short to VSS (or open floating to VSS) will result in a 0 at the chain output, signaling a defect.

It is important to note that the number of inputs pulled to 1 will affect the expected chain output value. If the number of chain inputs pulled to 1 is even, then expect 1 at the output. If the number of chain inputs pulled to 1 is odd, expect 0 at the output.

Continuing with the example in Table 197, as the input pins are driven to 1 across the chain in sequence, the XOR Output will toggle between 0 and 1. Any break in the toggling sequence (e.g., "1011") will identify the location of the short or open.

**Table 198. XOR Chain #1 (RTCRST# Asserted for 4 PCI Clocks While PWROK Active)**

| Pin Name | Ball # | Notes |
|---|---|---|
| AC_BIT_CLK | D8 | Top of XOR Chain |
| AC_SYNC | B8 | 2nd signal in XOR |
| AC_SDOUT | A9 | |
| PIRQE#/GPIO2 | D7 | |
| GNT2# | B7 | |
| REQ3 | B6 | |
| GNT3# | C7 | |
| GNTA#/GPIO16 | E8 | |
| REQB#/REQ5#/ GPIO1 | E7 | |
| REQ4# | C6 | |
| REQA#/GPIO0 | A5 | |
| PIRQF#/GPIO3 | A6 | |
| GNT4# | A4 | |
| GNTB#/ GNT5#GPIO17 | B4 | |
| GNT1# | A3 | |
| PIRQC# | A2 | |
| PIRQA# | B3 | |
| PIRQH#/GPIO5 | B1 | |
| PIRQD# | C2 | |
| REQ1# | C1 | |
| REQ2# | C5 | |
| AD18 | B2 | |
| REQ0# | D5 | |
| PIRQG#/GPIO4 | E2 | |
| AD28 | F5 | |
| PIRQB# | E1 | |
| | | |
| | | |

| Pin Name | Ball # | Notes |
|---|---|---|
| AD15 | G5 | |
| GNT0# | D4 | |
| AD22 | C4 | |
| AD30 | F4 | |
| AD20 | H3 | |
| AD16 | G4 | |
| AD4 | H5 | |
| AD24 | E6 | |
| AD0 | J4 | |
| STOP# | E5 | |
| AD11 | H4 | |
| AD26 | D3 | |
| AD6 | J3 | |
| TRDY# | E4 | |
| FRAME# | D2 | |
| AD9 | F2 | |
| AD2 | G3 | |
| PAR | F1 | |
| AD5 | H2 | |
| AD13 | G2 | |
| AD1 | J5 | |
| SERR# | L4 | |
| C/BE0# | E3 | |
| C/BE1# | J1 | |
| AD3 | K4 | |
| AD10 | M4 | |
| | | **XOR Chain #1** |
| **TP0** | AB2 | **OUTPUT** |

### Table 199. XOR Chain #2 (RTCRST# Asserted for 5 PCI Clocks While PWROK Active)

| Pin Name | Ball # | Notes |
|---|---|---|
| AD7 | J2 | Top of XOR Chain |
| AD8 | K5 | 2nd signal in XOR |
| IRDY# | M3 | |
| PERR# | K2 | |
| AD14 | K1 | |
| AD12 | L5 | |
| AD23 | N4 | |
| C/BE2# | N3 | |
| DEVSEL# | L3 | |
| PLOCK# | L2 | |
| AD17 | L1 | |
| AD19 | P5 | |
| AD21 | N5 | |
| C/BE3# | M2 | |
| AD25 | P3 | |
| AD27 | N2 | |
| AD29 | P4 | |
| AD31 | P2 | |
| GPIO6 | R5 | |
| PCICLK | N1 | |
| GPIO7 | U3 | |
| LAD0/FWH0 | T5 | |
| LAD1/FWH1 | R4 | |
| GPIO21 | R1 | |
| LDRQ0# | U5 | |
| LAD2/FWH2 | R3 | |
| | | |
| | | |

| Pin Name | Ball # | Notes |
|---|---|---|
| LAD3/FWH3 | U4 | |
| LDRQ1# | R2 | |
| LFRAME# / FWH4 | T4 | |
| GPIO32 | T1 | |
| THRM# | T2 | |
| CLK100N | AD5 | |
| CLK100P | AC5 | |
| SATA0RXN | AD7 | |
| SATA0RXP | AC7 | |
| SATA0TXN | AB8 | |
| SATA0TXP | AA8 | |
| SATA1RXN | AD9 | |
| SATA1RXP | AC9 | |
| SATA1TXN | AB10 | |
| SATA1TXP | AA10 | |
| SATARBIASN | Y9 | |
| SATARBIASP | Y11 | |
| GPIO8 | Y2 | |
| RI# | AB3 | |
| PWRBTN# | Y4 | |
| SLP_S5# | AA3 | |
| TP0 | AB2 | |
| AC_SDIN0 | E12 | |
| AC_SDIN2 | A13 | |
| AC_SDIN1 | D12 | |
| | | |
| | | **XOR Chain #2** |
| **GPIO0 / REQA#** | A5 | **OUTPUT** |

### Table 200. XOR Chain #3 (RTCRST# Asserted for 6 PCI Clocks While PWROK Active)

| Pin Name | Ball # | Notes |
|---|---|---|
| PDD6 | AD14 | Top of XOR Chain |
| PDD4 | AA14 | 2nd signal in XOR |
| PDD7 | AB14 | |
| PDD11 | AA15 | |
| PDD5 | AC15 | |
| PDD8 | AD15 | |
| PDD9 | Y15 | |
| PDD13 | Y16 | |
| PDD2 | Y14 | |
| PDD3 | AC14 | |
| PDD10 | AD16 | |
| PDDREQ | AC17 | |
| PDD12 | AC16 | |
| PDD14 | AA16 | |
| PDD1 | Y13 | |
| PDD15 | AB17 | |
| PDD0 | AB16 | |
| PDIOR# | AD18 | |
| PIORDY | AA18 | |
| PDDACK# | AC18 | |
| PDIOW# | AA17 | |
| IRQ14 | Y17 | |
| PDA1 | AD19 | |
| PDA0 | AA19 | |
| PDCS1# | AB19 | |
| PDA2 | AC19 | |
| | | |
| | | |
| | | |

| Pin Name | Ball # | Notes |
|---|---|---|
| PDCS3# | Y18 | |
| IRQ15 | Y24 | |
| GPIO19 | T20 | |
| GPIO18 | U21 | |
| GPIO22 | U20 | |
| GPIO20 | U22 | |
| VRMPWRGD | R20 | |
| A20GATE | T22 | |
| RCIN# | P23 | |
| TP1 | P20 | |
| THRMTRIP# | T21 | |
| A20M# | V23 | |
| CPUPWRGD | P24 | |
| INTR | U23 | |
| NMI | R22 | |
| INIT# | R23 | |
| STPCLK# | T24 | |
| CPUSLP# | P22 | |
| TP2 | R24 | |
| HI_STB/ HI_STBS | J24 | |
| GPIO23 | F22 | |
| SATALED# | G23 | |
| GPIO34 | F21 | |
| INTVRMEN | AD10 | |
| INTRUDER | Y12 | |
| RTC_RST# | AA12 | |
| | | |
| | | **XOR Chain #3** |
| **RI#** | AB3 | **OUTPUT** |

## Table 201. XOR Chain #4-1 (RTCRST# Asserted for 7 PCI Clocks While PWROK Active)

| Pin Name | Ball # | Notes |
|---|---|---|
| SDD7 | AC20 | Top of XOR Chain |
| SDD9 | AD22 | 2nd signal in XOR |
| SDD5 | AC21 | |
| SDD8 | Y19 | |
| SDD3 | AD24 | |
| SDD6 | AB20 | |
| SDD10 | AC22 | |
| SDD12 | AB22 | |
| SDD11 | AA20 | |
| SDD1 | AB23 | |
| SDD4 | AB21 | |
| SDD13 | AC24 | |
| SDDREQ | Y20 | |
| SDD14 | AB24 | |
| SDDACK# | W20 | |
| SIORDY | Y21 | |
| SDA1 | W23 | |
| SDD15 | AA23 | |
| SDD2 | AD23 | |
| SDA2 | W21 | |
| SDIOW# | Y22 | |
| SDD0 | AA22 | |
| SDCS1# | V22 | |
| SDIOR# | Y23 | |
| | | |
| | | |
| | | |

| Pin Name | Ball # | Notes |
|---|---|---|
| SDCS3# | V20 | |
| SDA0 | W22 | |
| FERR# | U24 | |
| IGNNE# | R21 | |
| SMI# | V24 | |
| CLK66 | N22 | |
| HI6 | N21 | |
| HI5 | M21 | |
| HI7 | M20 | |
| HI4 | M23 | |
| HICOMP | N24 | |
| HI_STB#/ HI_STBF | K23 | |
| HI3 | H23 | |
| HI2 | J20 | |
| HI1 | H21 | |
| HI0 | H20 | |
| HI10 | K21 | |
| HI8 | L22 | |
| HI9 | J22 | |
| HI11 | G22 | |
| SERIRQ | F23 | |
| CLK14 | F20 | |
| SPKR | E24 | |
| CLK48 | F24 | |
| | | |
| | | **XOR Chain #4-1** |
| **GPIO8** | Y2 | **OUTPUT** |

### Table 202. XOR Chain #4-2 (RTCRST# Asserted for 7 PCI Clocks While PWROK Active)

| Pin Name | Ball # | Notes |
|---|---|---|
| PME# | V2 | Top of XOR Chain |
| GPIO25 | W3 | 2nd signal in XOR |
| GPIO12 | W4 | |
| GPIO27 | V3 | |
| PCIRST# | V4 | |
| GPIO13 | W5 | |
| GPIO28 | W2 | |
| SLP_S4# | U2 | |
| SYS_RESET# | U1 | |
| SMLINK1 | AA2 | |
| GPIO24 | AC1 | |
| SUSCLK | Y1 | |
| SUS_STAT#/ LPCPD# | AB1 | |
| SMLINK0 | AD3 | |
| SLP_S3# | W1 | |
| SMBDATA | AD1 | |
| SMBCLK | AD2 | |
| SMBALERT#/ GPIO11 | AC3 | |
| LINKALERT# | V5 | |
| OC0# | C15 | |
| OC2# | D14 | |
| OC5# | A14 | |
| OC1# | D15 | |
| OC3# | C14 | |
| OC4# | B14 | |
| OC6# | B13 | |
| OC7# | C13 | |
| | | |
| | | |
| | | |
| | | |
| | | |

| Pin Name | Ball # | Notes |
|---|---|---|
| USBP0P | C23 | |
| USBP0N | D23 | |
| USBP1P | A22 | |
| USBP1N | B22 | |
| USBP2P | C21 | |
| USBP2N | D21 | |
| USBP3P | A20 | |
| USBP3N | B20 | |
| USBP4P | C19 | |
| USBP4N | D19 | |
| USBP5P | A18 | |
| USBP5N | B18 | |
| USBP6P | C17 | |
| USBP6N | D17 | |
| USBP7P | A16 | |
| USBP7N | B16 | |
| EE_SHCLK | A12 | |
| LAN_TXD2 | B12 | |
| LAN_CLK | E10 | |
| EE_DIN | B11 | |
| LAN_RXD2 | C11 | |
| LAN_RSTSYNC | D10 | |
| EE_CS | B10 | |
| LAN_TXD0 | D9 | |
| LAN_RXD0 | C10 | |
| No Connect | A11 | |
| LAN_TXD1 | E9 | |
| EE_DOUT | B9 | |
| LAN_RXD1 | C9 | |
| | | |
| | | **XOR Chain #4-2** |
| **AC_SDIN1** | D12 | **OUTPUT** |

### Table 203. XOR Chain #6 (RTCRST# Asserted for 52 PCI Clocks While PWROK Active)

| Pin Name | Ball # | Notes |
|---|---|---|
| RTCX1 | AC11 | Top of XOR Chain |
| | | |
| | | |

| Pin Name | Ball # | Notes |
|---|---|---|
| | | |
| | | **XOR Chain #6** |
| **TP0** | AB2 | **OUTPUT** |

This page is intentionally left blank.

# Register Index A

### Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 1 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| **LAN Controller (B1:D8:F0)** | | |
| Vendor Identification | 00–01h | Section 7.1.1, “VID—Vendor Identification Register (LAN Controller—B1:D8:F0)” on page 276 |
| Device Identification | 02–03h | Section 7.1.2, “DID—Device Identification Register (LAN Controller—B1:D8:F0)” on page 276 |
| PCI Command | 04–05h | Section 7.1.3, “PCICMD—PCI Command Register (LAN Controller—B1:D8:F0)” on page 277 |
| PCI Device Status | 06–07h | Section 7.1.4, “PCISTS—PCI Status Register (LAN Controller—B1:D8:F0)” on page 278 |
| Revision Identification | 08h | Section 7.1.5, “RID—Revision Identification Register (LAN Controller—B1:D8:F0)” on page 279 |
| Programming Interface | 09h | |
| Sub Class Code | 0Ah | Section 7.1.6, “SCC—Sub-Class Code Register (LAN Controller—B1:D8:F0)” on page 279 |
| Base Class Code | 0Bh | Section 7.1.7, “BCC—Base-Class Code Register (LAN Controller—B1:D8:F0)” on page 279 |
| Cache Line Size | 0Ch | Section 7.1.8, “CLS—Cache Line Size Register (LAN Controller—B1:D8:F0)” on page 280 |
| Master Latency Timer | 0Dh | Section 7.1.9, “PMLT—Primary Master Latency Timer Register (LAN Controller—B1:D8:F0)” on page 280 |
| Header Type | 0Eh | Section 7.1.10, “HEADTYP—Header Type Register (LAN Controller—B1:D8:F0)” on page 280 |
| CSR Memory-Mapped Base Address | 10–13h | Section 7.1.11, “CSR_MEM_BASE — CSR Memory-Mapped Base Address Register (LAN Controller—B1:D8:F0)” on page 281 |
| CSR I/O-Mapped Base Address | 14–17h | Section 7.1.12, “CSR_IO_BASE — CSR I/O-Mapped Base Address Register (LAN Controller—B1:D8:F0)” on page 281 |
| Subsystem Vendor ID | 2C–2Dh | Section 7.1.13, “SVID — Subsystem Vendor Identification Register (LAN Controller—B1:D8:F0)” on page 281 |
| Subsystem ID | 2E–2Fh | Section 7.1.14, “SID — Subsystem Identification Register (LAN Controller—B1:D8:F0)” on page 282 |
| Capabilities Pointer | 34h | Section 7.1.15, “CAP_PTR — Capabilities Pointer Register (LAN Controller—B1:D8:F0)” on page 282 |
| Interrupt Line Register | 3Ch | Section 7.1.16, “INT_LN — Interrupt Line Register (LAN Controller—B1:D8:F0)” on page 282 |
| Interrupt Pin Register | 3Dh | Section 7.1.17, “INT_PN — Interrupt Pin Register (LAN Controller—B1:D8:F0)” on page 283 |
| Minimum Grant Register | 3Eh | Section 7.1.18, “MIN_GNT — Minimum Grant Register (LAN Controller—B1:D8:F0)” on page 283 |
| Maximum Latency Register | 3Fh | Section 7.1.19, “MAX_LAT — Maximum Latency Register (LAN Controller—B1:D8:F0)” on page 283 |

## Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 2 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Capability ID Register | DCh | Section 7.1.20, “CAP_ID — Capability Identification Register (LAN Controller—B1:D8:F0)” on page 283 |
| Next Item Pointer | DDh | Section 7.1.21, “NXT_PTR — Next Item Pointer Register (LAN Controller—B1:D8:F0)” on page 284 |
| Power Management Capabilities | DE–DFh | Section 7.1.22, “PM_CAP — Power Management Capabilities Register (LAN Controller—B1:D8:F0)” on page 284 |
| Power Management Control/Status Register | E0–E1h | Section 7.1.23, “PMCSR — Power Management Control/Status Register (LAN Controller—B1:D8:F0)” on page 285 |
| Data Register | E3h | Section 7.1.24, “PCIDATA — PCI Power Management Data Register (LAN Controller—B1:D8:F0)” on page 286 |
| **Hub Interface to PCI Bridge D30:F0** | | |
| Vendor ID | 00–01h | Section 8.1.1, “VID—Vendor Identification Register (HUB-PCI—D30:F0)” on page 302 |
| Device ID | 02–03h | Section 8.1.2, “DID—Device Identification Register (HUB-PCI—D30:F0)” on page 302 |
| PCI Device Command Register | 04–05h | Section 8.1.3, “PCICMD—PCI Command Register (HUB-PCI—D30:F0)” on page 303 |
| PCI Device Status Register | 06–07h | Section 8.1.4, “PCISTS—PCI Status Register (HUB-PCI—D30:F0)” on page 304 |
| Revision ID | 08h | Section 8.1.6, “SCC—Sub-Class Code Register (HUB-PCI—D30:F0)” on page 305 |
| Sub Class Code | 0Ah | Section 8.1.6, “SCC—Sub-Class Code Register (HUB-PCI—D30:F0)” on page 305 |
| Base Class Code | 0Bh | Section 8.1.7, “BCC—Base-Class Code Register (HUB-PCI—D30:F0)” on page 305 |
| Primary Master Latency Timer | 0Dh | Section 8.1.8, “PMLT—Primary Master Latency Timer Register (HUB-PCI—D30:F0)” on page 305 |
| Header Type | 0Eh | Section 8.1.9, “HEADTYP—Header Type Register (HUB-PCI—D30:F0)” on page 306 |
| Primary Bus Number | 18h | Section 8.1.10, “PBUS_NUM—Primary Bus Number Register (HUB-PCI—D30:F0)” on page 306 |
| Secondary Bus Number | 19h | Section 8.1.11, “SBUS_NUM—Secondary Bus Number Register (HUB-PCI—D30:F0)” on page 306 |
| Subordinate Bus Number | 1Ah | Section 8.1.12, “SUB_BUS_NUM—Subordinate Bus Number Register (HUB-PCI—D30:F0)” on page 306 |
| Secondary Master Latency Timer | 1Bh | Section 8.1.13, “SMLT—Secondary Master Latency Timer Register (HUB-PCI—D30:F0)” on page 307 |
| IO Base Register | 1Ch | Section 8.1.14, “IOBASE—I/O Base Register (HUB-PCI—D30:F0)” on page 307 |
| IO Limit Register | 1Dh | Section 8.1.15, “IOLIM—I/O Limit Register (HUB-PCI—D30:F0)” on page 307 |
| Secondary Status Register | 1E–1Fh | Section 8.1.16, “SECSTS—Secondary Status Register (HUB-PCI—D30:F0)” on page 308 |
| Memory Base | 20–21h | Section 8.1.17, “MEMBASE—Memory Base Register (HUB-PCI—D30:F0)” on page 309 |
| Memory Limit | 22–23h | Section 8.1.18, “MEMLIM—Memory Limit Register (HUB-PCI—D30:F0)” on page 309 |
| Prefetchable Memory Base | 24–25h | Section 8.1.19, “PREF_MEM_BASE—Prefetchable Memory Base Register (HUB-PCI—D30:F0)” on page 309 |

## Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 3 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Prefetchable Memory Limit | 26–27h | Section 8.1.20, “PREF_MEM_MLT—Prefetchable Memory Limit Register (HUB-PCI—D30:F0)” on page 310 |
| I/O Base Upper 16 Bits | 30–31h | Section 8.1.21, “IOBASE_HI—I/O Base Upper 16 Bits Register (HUB-PCI—D30:F0)” on page 310 |
| I/O Limit Upper 16 Bits | 32–33h | Section 8.1.22, “IOLIM_HI—I/O Limit Upper 16 Bits Register (HUB-PCI—D30:F0)” on page 310 |
| Interrupt Line | 3Ch | Section 8.1.23, “INT_LN—Interrupt Line Register (HUB-PCI—D30:F0)” on page 310 |
| Bridge Control | 3E–3Fh | Section 8.1.24, “BRIDGE_CNT—Bridge Control Register (HUB-PCI—D30:F0)” on page 311 |
| Hub Interface 1 Command Control | 40–43h | Section 8.1.25, “HI1_CMD—Hub Interface 1 Command Control Register (HUB-PCI—D30:F0)” on page 312 |
| Secondary PCI Device Hiding | 44–45h | Section 8.1.26, “DEVICE_HIDE—Secondary PCI Device Hiding Register (HUB-PCI—D30:F0)” on page 313 |
| Policy Configuration Register | 50–53h | Section 8.1.27, “CNF—Policy Configuration Register (HUB-PCI—D30:F0)” on page 314 |
| Multi-Transaction Timer | 70h | Section 8.1.28, “MTT—Multi-Transaction Timer Register (HUB-PCI—D30:F0)” on page 315 |
| PCI Master Status | 82h | Section 8.1.29, “PCI_MAST_STS—PCI Master Status Register (HUB-PCI—D30:F0)” on page 315 |
| Error Command Register | 90h | Section 8.1.30, “ERR_CMD—Error Command Register (HUB-PCI—D30:F0)” on page 316 |
| Error Status Register | 92h | Section 8.1.31, “ERR_STS—Error Status Register (HUB-PCI—D30:F0)” on page 316 |
| **LPC Bridge D31:F0** | | |
| Vendor ID | 00–01h | Section 9.1.1, “VID—Vendor Identification Register (LPC I/F—D31:F0)” on page 318 |
| Device ID | 02–03h | Section 9.1.2, “DID—Device Identification Register (LPC I/F—D31:F0)” on page 318 |
| PCI Command Register | 04–05h | Section 9.1.3, “PCICMD—PCI COMMAND Register (LPC I/F—D31:F0)” on page 319 |
| PCI Device Status Register | 06–07h | Section 9.1.4, “PCISTS—PCI Status Register (LPC I/F—D31:F0)” on page 320 |
| Revision ID | 08h | Section 9.1.6, “PI—Programming Interface Register (LPC I/F—D31:F0)” on page 321 |
| Programming Interface | 09h | Section 9.1.6, “PI—Programming Interface Register (LPC I/F—D31:F0)” on page 321 |
| Sub Class Code | 0Ah | Section 9.1.7, “SCC—Sub Class Code Register (LPC I/F—D31:F0)” on page 321 |
| Base Class Code | 0Bh | Section 9.1.8, “BCC—Base Class Code Register (LPC I/F—D31:F0)” on page 321 |
| Header Type | 0Eh | Section 9.1.9, “HEADTYP—Header Type Register (LPC I/F—D31:F0)” on page 322 |
| ACPI Base Address Register | 40–43h | Section 9.1.10, “PMBASE—ACPI Base Address Register (LPC I/F—D31:F0)” on page 322 |
| ACPI Control | 44h | Section 9.1.11, “ACPI_CNTL—ACPI Control Register (LPC I/F — D31:F0)” on page 323 |
| BIOS Control Register | 4E–4Fh | Section 9.1.13, “TCO_CNTL — TCO Control Register (LPC I/F — D31:F0)” on page 324 |

## Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 4 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| TCO Control | 54h | Section 9.1.13, “TCO_CNTL — TCO Control Register (LPC I/F — D31:F0)” on page 324 |
| GPIO Base Address Register | 58–5Bh | Section 9.1.14, “GPIO_BASE—GPIO Base Address Register (LPC I/F—D31:F0)” on page 325 |
| GPIO Control Register | 5Ch | Section 9.1.15, “GPIO_CNTL—GPIO Control Register (LPC I/F—D31:F0)” on page 325 |
| PIRQ[A:D] Routing Control | 60–63h | Section 9.1.16, “PIRQ[n]_ROUT—PIRQ[A,B,C,D] Routing Control Register (LPC I/F—D31:F0)” on page 326 |
| Serial IRQ Control Register | 64h | Section 9.1.17, “SIRQ_CNTL—Serial IRQ Control Register (LPC I/F—D31:F0)” on page 327 |
| PIRQ[E:H] Routing Control | 68–6Bh | Section 9.1.8, “BCC—Base Class Code Register (LPC I/F—D31:F0)” on page 321 |
| Device 31 Error Config Register | 88h | Section 9.1.19, “D31_ERR_CFG—Device 31 Error Configuration Register (LPC I/F—D31:F0)” on page 328 |
| Device 31 Error Status Register | 8Ah | Section 9.1.20, “D31_ERR_STS—Device 31 Error Status Register (LPC I/F—D31:F0)” on page 329 |
| PCI DMA Configuration Registers | 90–91h | Section 9.1.21, “PCI_DMA_CFG—PCI DMA Configuration Register (LPC I/F—D31:F0)” on page 329 |
| General Power Management Configuration 1 | A0h | Section 9.8.1, “GEN_PMCON_1—General PM Configuration 1 Register (PM—D31:F0)” on page 377 |
| General Power Management Configuration 2 | A2h | Section 9.8.2, “GEN_PMCON_2—General PM Configuration 2 Register (PM—D31:F0)” on page 378 |
| General Power Management Configuration 3 | A4h | Section 9.8.3, “GEN_PMCON_3—General PM Configuration 3 Register (PM—D31:F0)” on page 379 |
| Stop Clock Delay Register | A8h | Section 9.8.4, “STPCLK_DEL—Stop Clock Delay Register (PM—D31:F0)” on page 380 |
| GPI_ROUT | B8–BBh | Section 9.8.7, “GPI_ROUT—GPI Routing Control Register (PM—D31:F0)” on page 381 |
| I/O Monitor Trap Forwarding Enable Register | C0h | Section 9.8.8, “TRP_FWD_EN—IO Monitor Trap Forwarding Enable Register (PM—D31:F0)” on page 382 |
| I/O Monitor [4:7] Trap Range Registers | C4h, C6h, C8h, CAh | Section 9.8.9, “MON[n]_TRP_RNG—I/O Monitor [4:7] Trap Range Register for Devices 4–7 (PM—D31:F0)” on page 383 |
| I/O Monitor [4:7] Trap Mask Register | CCh | Section 9.8.10, “MON_TRP_MSK—I/O Monitor Trap Range Mask Register for Devices 4–7 (PM—D31:F0)” on page 383 |
| General Control | D0h–D3h | Section 9.1.22, “GEN_CNTL — General Control Register (LPC I/F — D31:F0)” on page 330 |
| General Status | D4h | Section 9.1.23, “GEN_STA—General Status Register (LPC I/F—D31:F0)” on page 332 |
| Backed Up Control | D5h | Section 9.1.24, “BACK_CNTL—Backed Up Control Register (LPC I/F—D31:F0)” on page 333 |
| Real Time Clock Configuration | D8h | Section 9.1.25, “RTC_CONF—Real Time Clock Configuration Register (LPC I/F—D31:F0)” on page 334 |
| LPC COM Port Decode Ranges | E0h | Section 9.1.26, “COM_DEC—LPC I/F Communication Port Decode Ranges Register (LPC I/F—D31:F0)” on page 335 |
| LPC FDD & LPT Decode Ranges | E1h | Section 9.1.27, “LPCFDD_DEC—LPC I/F FDD and LPT Decode Ranges Register (LPC I/F—D31:F0)” on page 335 |

## Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 5 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Flash BIOS Decode Enable 1 Register | E3h | Section 9.1.28, “FB_DEC_EN1—Flash BIOS Decode Enable 1 Register (LPC I/F—D31:F0)” on page 336 |
| LPC Generic Decode Range 1 | E4h–E5h | Section 9.1.29, “GEN1_DEC—LPC I/F Generic Decode Range 1 Register (LPC I/F—D31:F0)” on page 337 |
| LPC Enables | E6h–E7h | Section 9.1.30, “LPC_EN—LPC I/F Enables Register (LPC I/F—D31:F0)” on page 337 |
| Flash BIOS Select 1 Register | E8h–EBh | Section 9.1.31, “FB_SEL1—Flash BIOS Select 1 Register (LPC I/F—D31:F0)” on page 339 |
| LPC Generic Decode Range 2 | ECh–EDh | Section 9.1.32, “GEN2_DEC—LPC I/F Generic Decode Range 2 Register (LPC I/F—D31:F0)” on page 340 |
| Flash BIOS Select 2 Register | EE–EFh | Section 9.1.33, “FB_SEL2—Flash BIOS Select 2 Register (LPC I/F—D31:F0)” on page 340 |
| Flash BIOS Decode Enable 2 Register | F0h | Section 9.1.34, “FB_DEC_EN2—Flash BIOS Decode Enable 2 Register (LPC I/F—D31:F0)” on page 341 |
| Function Disable Register | F2h | Section 9.1.35, “FUNC_DIS—Function Disable Register (LPC I/F—D31:F0)” on page 342 |
| **IDE Controller (D31:F1)** | | |
| Vendor ID | 00h–01h | Section 10.1.1, “VID—Vendor Identification Register (IDE—D31:F0)” on page 416 |
| Device ID | 02h–03h | Section 10.1.2, “DID—Device Identification Register (IDE—D31:F0)” on page 416 |
| Command Register | 04h–05h | Section 10.1.3, “PCICMD—PCI Command Register (IDE—D31:F1)” on page 417 |
| Device Status | 06h–07h | Section 10.1.4, “PCISTS — PCI Status Register (IDE—D31:F1)” on page 418 |
| Revision ID | 08h | |
| Programming Interface | 09h | Section 10.1.5, “RID—Revision Identification Register (IDE—D31:F1)” on page 419 |
| Sub Class Code | 0Ah | Section 10.1.7, “SCC—Sub Class Code Register (IDE—D31:F1)” on page 419 |
| Base Class Code | 0Bh | Section 10.1.8, “BCC—Base Class Code Register (IDE—D31:F1)” on page 420 |
| Master Latency Timer | 0Dh | Section 10.1.9, “PMLT—Primary Master Latency Timer Register (IDE—D31:F1)” on page 420 |
| Header Type | 0Eh | |
| Primary Command Block Base Address | 10–13h | Section 10.1.10, “PCMD_BAR—Primary Command Block Base Address Register (IDE—D31:F1)” on page 420 |
| Primary Control Block Base Address | 14–17h | Section 10.1.11, “PCNL_BAR—Primary Control Block Base Address Register (IDE—D31:F1)” on page 421 |
| Secondary Command Block Base Address | 18–1Bh | Section 10.1.12, “SCMD_BAR—Secondary Command Block Base Address Register (IDE D31:F1)” on page 421 |
| Secondary Control Block Base Address | 1C–1Fh | Section 10.1.13, “SCNL_BAR—Secondary Control Block Base Address Register (IDE D31:F1)” on page 421 |
| Bus Master Base Address Register | 20h–23h | Section 10.1.14, “BM_BASE — Bus Master Base Address Register (IDE—D31:F1)” on page 422 |
| Subsystem Vendor ID | 2C–2Dh | Section 10.1.15, “IDE_SVID — Subsystem Vendor Identification (IDE—D31:F1)” on page 422 |

## Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 6 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Subsystem ID | 2E–2Fh | Section 10.1.16, “IDE_SID — Subsystem Identification Register (IDE—D31:F1)” on page 422 |
| Interrupt Line | 3Ch | Section 10.1.17, “INTR_LN—Interrupt Line Register (IDE—D31:F1)” on page 423 |
| Interrupt Pin | 3Dh | Section 10.1.18, “INTR_PN—Interrupt Pin Register (IDE—D31:F1)” on page 423 |
| Primary/Secondary IDE Timing | 40h–43h | Section 10.1.19, “IDE_TIM — IDE Timing Register (IDE—D31:F1)” on page 424 |
| Slave IDE Timing | 44h | Section 10.1.20, “SLV_IDETIM—Slave (Drive 1) IDE Timing Register (IDE—D31:F1)” on page 425 |
| Synchronous DMA Control Register | 48h | Section 10.1.21, “SDMA_CNT—Synchronous DMA Control Register (IDE—D31:F1)” on page 427 |
| Synchronous DMA Timing Register | 4Ah–4Bh | Section 10.1.22, “SDMA_TIM—Synchronous DMA Timing Register (IDE—D31:F1)” on page 428 |
| IDE I/O Configuration Register | 54h | Section 10.1.23, “IDE_CONFIG—IDE I/O Configuration Register (IDE—D31:F1)” on page 429 |
| **SATA Controller (D31:F2)** | | |
| Vendor ID | 00h–01h | Section 11.1.1, “VID—Vendor Identification Register (SATA—D31:F2)” on page 434 |
| Device ID | 02h–03h | Section 11.1.2, “DID—Device Identification Register (SATA—D31:F2)” on page 434 |
| Command Register | 04h–05h | Section 11.1.3, “PCICMD—PCI Command Register (SATA–D31:F2)” on page 435 |
| Device Status | 06h–07h | Section 11.1.4, “PCISTS — PCI Status Register (SATA–D31:F2)” on page 436 |
| Revision ID | 08h | |
| Programming Interface | 09h | Section 11.1.6, “PI—Programming Interface Register (SATA–D31:F2)” on page 437 |
| Sub Class Code | 0Ah | Section 11.1.7, “SCC—Sub Class Code Register (SATA–D31:F2)” on page 437 |
| Base Class Code | 0Bh | Section 11.1.8, “BCC—Base Class Code Register (SATA–D31:F2)” on page 438 |
| Master Latency Timer | 0Dh | Section 11.1.9, “PMLT—Primary Master Latency Timer Register (SATA–D31:F2)” on page 438 |
| Header Type | 0Eh | |
| Primary Command Block Base Address | 10–13h | Section 11.1.10, “PCMD_BAR—Primary Command Block Base Address Register (SATA–D31:F2)” on page 438 |
| Primary Control Block Base Address | 14–17h | Section 11.1.11, “PCNL_BAR—Primary Control Block Base Address Register (SATA–D31:F2)” on page 439 |
| Secondary Command Block Base Address | 18–1Bh | Section 11.1.12, “SCMD_BAR—Secondary Command Block Base Address Register (IDE D31:F1)” on page 439 |
| Secondary Control Block Base Address | 1C–1Fh | Section 11.1.13, “SCNL_BAR—Secondary Control Block Base Address Register (IDE D31:F1)” on page 439 |
| Bus Master Base Address Register | 20h–23h | Section 11.1.14, “BAR — Legacy Bus Master Base Address Register (SATA–D31:F2)” on page 440 |
| Subsystem Vendor ID | 2C–2Dh | Section 11.1.15, “SVID—Subsystem Vendor Identification Register (SATA–D31:F2)” on page 440 |

## Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 7 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Subsystem ID | 2E–2Fh | Section 11.1.16, “SID—Subsystem Identification Register (SATA–D31:F2)” on page 440 |
| Capabilities Pointer | 34h | Section 11.1.17, “CAP—Capabilities Pointer Register (SATA–D31:F2)” on page 441 |
| Interrupt Line | 3Ch | Section 11.1.18, “INT_LN—Interrupt Line Register (SATA–D31:F2)” on page 441 |
| Interrupt Pin | 3Dh | Section 11.1.19, “INT_PN—Interrupt Pin Register (SATA–D31:F2)” on page 441 |
| Primary/Secondary IDE Timing | 40h–43h | Section 11.1.20, “IDE_TIM — IDE Timing Register (SATA–D31:F2)” on page 442 |
| Slave IDE Timing | 44h | Section 11.1.21, “SIDETIM—Slave IDE Timing Register (SATA–D31:F2)” on page 444 |
| Synchronous DMA Control Register | 48h | Section 11.1.22, “SDMA_CNT—Synchronous DMA Control Register (SATA–D31:F2)” on page 445 |
| Synchronous DMA Timing Register | 4Ah–4Bh | Section 11.1.23, “SDMA_TIM—Synchronous DMA Timing Register (SATA–D31:F2)” on page 446 |
| IDE I/O Configuration Register | 54h | Section 11.1.24, “IDE_CONFIG—IDE I/O Configuration Register (SATA–D31:F2)” on page 447 |
| PCI Power Management Capability ID | 70–71h | Section 11.1.25, “PID—PCI Power Management Capability Identification Register (SATA–D31:F2)” on page 448 |
| PCI Power Management Capabilities | 72–73h | Section 11.1.26, “PC—PCI Power Management Capabilities Register (SATA–D31:F2)” on page 448 |
| PCI Power Management Control and Status | 74–77h | Section 11.1.27, “PMCS—PCI Power Management Control and Status Register (SATA–D31:F2)” on page 449 |
| Message Signaled Interrupt Capability ID | 80–81h | Section 11.1.28, “MID—Message Signaled Interrupt Identifiers Register (SATA–D31:F2)” on page 449 |
| Message Signaled Interrupt Message Control | 82–83h | Section 11.1.29, “MC—Message Signaled Interrupt Message Control Register (SATA–D31:F2)” on page 450 |
| Message Signaled Interrupt Message Address | 84–87h | Section 11.1.30, “MA—Message Signaled Interrupt Message Address Register (SATA–D31:F2)” on page 450 |
| Message Signaled Interrupt Message Data | 88–89h | Section 11.1.31, “MD—Message Signaled Interrupt Message Data Register (SATA–D31:F2)” on page 450 |
| Address Map | 90h | Section 11.1.32, “MAP—Address Map Register (SATA–D31:F2)” on page 451 |
| Port Status and Control | 92h | Section 11.1.33, “PCS—Port Control and Status Register (SATA–D31:F2)” on page 451 |
| **UHCI Controller (D29:F0/F1/F2/F3)** | | |
| Vendor ID | 00–01h | Section 12.1.1, “VID—Vendor Identification Register (USB—D29:F0/F1/F2/F3)” on page 462 |
| Device ID | 02–03h | Section 12.1.2, “DID—Device Identification Register (USB—D29:F0/F1/F2/F3)” on page 462 |
| Command Register | 04–05h | Section 12.1.3, “PCICMD—PCI Command Register (USB—D29:F0/F1/F2/F3)” on page 462 |
| Device Status | 06–07h | Section 12.1.4, “PCISTS—PCI Status Register (USB—D29:F0/F1/F2/F3)” on page 463 |
| Revision ID | 08h | Section 12.1.5, “RID—Revision Identification Register (USB—D29:F0/F1/F2/F3)” on page 463 |
| Programming Interface | 09h | Section 12.1.6, “PI—Programming Interface Register (USB—D29:F0/F1/F2/F3)” on page 464 |

### Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 8 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Sub Class Code | 0Ah | Section 12.1.7, “SCC—Sub Class Code Register (USB—D29:F0/F1/F2/F3)” on page 464 |
| Base Class Code | 0Bh | Section 12.1.8, “BCC—Base Class Code Register (USB—D29:F0/F1/F2/F3)” on page 464 |
| Header Type | 0Eh | Section 12.1.9, “HEADTYP—Header Type Register (USB—D29:F0/F1/F2/F3)” on page 465 |
| Base Address Register | 20–23h | Section 12.1.10, “BASE—Base Address Register (USB—D29:F0/F1/F2/F3)” on page 465 |
| Subsystem Vendor ID | 2C–2Dh | Section 12.1.11, “SVID — Subsystem Vendor Identification Register (USB—D29:F0/F1/F2/F3)” on page 466 |
| Subsystem ID | 2E–2Fh | Section 12.1.12, “SID — Subsystem Identification Register (USB—D29:F0/F1/F2/F3)” on page 466 |
| Interrupt Line | 3Ch | Section 12.1.13, “INT_LN—Interrupt Line Register (USB—D29:F0/F1/F2/F3)” on page 466 |
| Interrupt Pin | 3Dh | Section 12.1.14, “INT_PN—Interrupt Pin Register (USB—D29:F0/F1/F2/F3)” on page 467 |
| Serial Bus Release Number | 60h | Section 12.1.15, “USB_RELNUM—Serial Bus Release Number Register (USB—D29:F0/F1/F2/F3)” on page 467 |
| USB Legacy Keyboard/ Mouse Control | C0–C1h | Section 12.1.16, “USB_LEGKEY—USB Legacy Keyboard/ Mouse Control Register (USB—D29:F0/F1/F2/F3)” on page 468 |
| USB Resume Enable | C4h | Section 12.1.17, “USB_RES—USB Resume Enable Register (USB—D29:F0/F1/F2/F3)” on page 470 |
| **EHCI Controller (D29:F7)** | | |
| Vendor ID | 00–01h | |
| Device ID | 02–03h | |
| Command Register | 04–05h | Section 13.1.3, “PCICMD—PCI Command Register (USB EHCI—D29:F7)” on page 481 |
| Device Status | 06–07h | Section 13.1.4, “PCISTS—PCI Status Register (USB EHCI—D29:F7)” on page 482 |
| Revision ID | 08h | |
| Programming Interface | 09h | Section 13.1.6, “PI—Programming Interface Register (USB EHCI—D29:F7)” on page 483 |
| Sub Class Code | 0Ah | Section 13.1.7, “SCC—Sub Class Code Register (USB EHCI—D29:F7)” on page 483 |
| Base Class Code | 0Bh | Section 13.1.8, “BCC—Base Class Code Register (USB EHCI—D29:F7)” on page 483 |
| Master Latency Timer | 0Dh | Section 13.1.9, “PMLT—Primary Master Latency Timer Register (USB EHCI—D29:F7)” on page 484 |
| Header Type | 0Eh | |
| Base Address Register | 20–23h | Section 13.1.10, “MEM_BASE—Memory Base Address Register (USB EHCI—D29:F7)” on page 484 |
| Subsystem Vendor ID | 2C–2Dh | Section 13.1.11, “SVID—USB EHCI Subsystem Vendor ID Register (USB EHCI—D29:F7)” on page 484 |
| Subsystem ID | 2E–2Fh | Section 13.1.12, “SID—USB EHCI Subsystem ID Register (USB EHCI—D29:F7)” on page 485 |
| Capabilities Pointer | 34h | Section 13.1.13, “CAP_PTR—Capabilities Pointer Register (USB EHCI—D29:F7)” on page 485 |

## Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 9 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Interrupt Line | 3Ch | Section 13.1.14, “INT_LN—Interrupt Line Register (USB EHCI—D29:F7)” on page 485 |
| Interrupt Pin | 3Dh | Section 13.1.15, “INT_PN—Interrupt Pin Register (USB EHCI—D29:F7)” on page 485 |
| Power Management Capability ID | 50h | Section 13.1.16, “PWR_CAPID—PCI Power Management Capability ID Register (USB EHCI—D29:F7)” on page 486 |
| Next Item Pointer | 51h | Section 13.1.17, “NXT_PTR1—Next Item Pointer #1 Register (USB EHCI—D29:F7)” on page 486 |
| Power Management Capabilities | 52–53h | Section 13.1.18, “PWR_CAP—Power Management Capabilities Register (USB EHCI—D29:F7)” on page 487 |
| Power Management Control and Status | 54–55h | Section 13.1.19, “PWR_CNTL_STS—Power Management Control/Status Register (USB EHCI—D29:F7)” on page 488 |
| Debug Port Capability ID | 58h | Section 13.1.20, “DEBUG_CAPID—Debug Port Capability ID Register (USB EHCI—D29:F7)” on page 488 |
| Next Item Pointer #2 | 59h | Section 13.1.21, “NXT_PTR2—Next Item Pointer #2 Register (USB EHCI—D29:F7)” on page 489 |
| Debug Port Base Offset | 5A–5Bh | Section 13.1.22, “DEBUG_BASE—Debug Port Base Offset Register (USB EHCI—D29:F7)” on page 489 |
| USB Release Number | 60h | Section 13.1.23, “USB_RELNUM—USB Release Number Register (USB EHCI—D29:F7)” on page 489 |
| Frame Length Adjustment | 61h | Section 13.1.24, “FL_ADJ—Frame Length Adjustment Register (USB EHCI—D29:F7)” on page 490 |
| Power Wake Capabilities | 62–63h | Section 13.1.25, “PWAKE_CAP—Port Wake Capability Register (USB EHCI—D29:F7)” on page 491 |
| USB 2.0 Legacy Support Extended Capability | 68–6Bh | Section 13.1.26, “LEG_EXT_CAP—USB EHCI Legacy Support Extended Capability Register (USB EHCI—D29:F7)” on page 491 |
| USB 2.0 Legacy Support Control and Status | 6C–6Fh | Section 13.1.27, “LEG_EXT_CS—USB EHCI Legacy Support Extended Control / Status Register (USB EHCI—D29:F7)” on page 492 |
| Intel Specific USB 2.0 SMI | 70–73h | Section 13.1.28, “SPECIAL_SMI—Intel Specific USB 2.0 SMI Register (USB EHCI—D29:F7)” on page 493 |
| Access Control | 80h | Section 13.1.28, “SPECIAL_SMI—Intel Specific USB 2.0 SMI Register (USB EHCI—D29:F7)” on page 493 |
| **SMBus Controller (D31:F3)** | | |
| Vendor ID | 00–01h | Section 14.1.1, “VID—Vendor Identification Register (SMBUS—D31:F3)” on page 515 |
| Device ID | 02–03h | Section 14.1.2, “DID—Device Identification Register (SMBUS—D31:F3)” on page 516 |
| Command Register | 04–05h | Section 14.1.3, “PCICMD—PCI Command Register (SMBUS—D31:F3)” on page 516 |
| Device Status | 06–07h | Section 14.1.4, “PCISTS—PCI Status Register (SMBUS—D31:F3)” on page 517 |
| Revision ID | 08h | Section 14.1.5, “RID—Revision Identification Register (SMBUS—D31:F3)” on page 517 |
| Programming Interface | 09h | |
| Sub Class Code | 0Ah | Section 14.1.7, “BCC—Base Class Code Register (SMBUS—D31:F3)” on page 518 |

## Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 10 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Base Class Code | 0Bh | Section 14.1.8, “SMB_BASE—SMBUS Base Address Register (SMBUS—D31:F3)” on page 518 |
| SMB Base Address Register | 20–23h | Section 14.1.8, “SMB_BASE—SMBUS Base Address Register (SMBUS—D31:F3)” on page 518 |
| Interrupt Line | 3Ch | Section 14.1.11, “INT_LN—Interrupt Line Register (SMBUS—D31:F3)” on page 519 |
| Interrupt Pin | 3Dh | Section 14.1.12, “INT_PN—Interrupt Pin Register (SMBUS—D31:F3)” on page 519 |
| Host Configuration | 40h | Section 14.1.13, “HOSTC—Host Configuration Register (SMBUS—D31:F3)” on page 520 |
| **AC’97 Audio Controller (D31:F5)** | | |
| Vendor Identification | 00h–01h | Section 15.1.1, “VID—Vendor Identification Register (Audio—D31:F5)” on page 534 |
| Device Identification | 02h–03h | Section 15.1.2, “DID—Device Identification Register (Audio—D31:F5)” on page 534 |
| PCI Command | 04h–05h | Section 15.1.3, “PCICMD—PCI Command Register (Audio—D31:F5)” on page 535 |
| PCI Device Status | 06h–07h | Section 15.1.4, “PCISTS—PCI Status Register (Audio—D31:F5)” on page 536 |
| Revision Identification | 08h | Section 15.1.5, “RID—Revision Identification Register (Audio—D31:F5)” on page 537 |
| Programming Interface | 09h | Section 15.1.6, “PI—Programming Interface Register (Audio—D31:F5)” on page 537 |
| Sub Class Code | 0Ah | Section 15.1.7, “SCC—Sub Class Code Register (Audio—D31:F5)” on page 537 |
| Base Class Code | 0Bh | Section 15.1.8, “BCC—Base Class Code Register (Audio—D31:F5)” on page 537 |
| Header Type | 0Eh | Section 15.1.9, “HEADTYP—Header Type Register (Audio—D31:F5)” on page 538 |
| Native Audio Mixer Base Address | 10h–13h | Section 15.1.10, “NAMBAR—Native Audio Mixer Base Address Register (Audio—D31:F5)” on page 538 |
| Native Audio Bus Mastering Base Address | 14h–17h | Section 15.1.10, “NAMBAR—Native Audio Mixer Base Address Register (Audio—D31:F5)” on page 538 |
| Mixer Base Address(Mem) | 18–1Bh | Section 15.1.12, “MMBAR—Mixer Base Address Register (Audio—D31:F5)” on page 539 |
| Bus Master Base Address(Mem) | 1C–1F | Section 15.1.13, “MBBAR—Bus Master Base Address Register (Audio—D31:F5)” on page 540 |
| Subsystem Vendor ID | 2Ch–2Dh | Section 15.1.14, “SVID—Subsystem Vendor Identification Register (Audio—D31:F5)” on page 540 |
| Subsystem ID | 2Eh–2Fh | Section 15.1.15, “SID—Subsystem Identification Register (Audio—D31:F5)” on page 541 |
| Capabilities Pointer | 34h | Section 15.1.16, “CAP_PTR—Capabilities Pointer Register (Audio—D31:F5)” on page 541 |
| Interrupt Line | 3Ch | Section 15.1.17, “INT_LN—Interrupt Line Register (Audio—D31:F5)” on page 541 |
| Interrupt Pin | 3Dh | Section 15.1.18, “INT_PN—Interrupt Pin Register (Audio—D31:F5)” on page 542 |
| Programmable Codec ID | 40h | Section 15.1.19, “PCID—Programmable Codec Identification Register (Audio—D31:F5)” on page 542 |

## Table 204. Intel® ICH5 PCI Configuration Registers (Sheet 11 of 11)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Configuration | 41h | Section 15.1.20, “CFG—Configuration Register (Audio—D31:F5)” on page 543 |
| PCI Power Management ID | 50–51h | Section 15.1.21, “PID—PCI Power Management Capability Identification Register (Audio—D31:F5)” on page 543 |
| PC -Power Management Capabilities | 52–53h | Section 15.1.22, “PC—Power Management Capabilities Register (Audio—D31:F5)” on page 544 |
| Power Management Control and Status | 54–55h | Section 15.1.23, “PCS—Power Management Control and Status Register (Audio—D31:F5)” on page 545 |
| **AC’97 Modem Controller (D31:F6)** | | |
| Vendor Identification | 00h–01h | Section 16.1.1, “VID—Vendor Identification Register (Modem—D31:F6)” on page 560 |
| Device Identification | 02h–03h | Section 16.1.2, “DID—Device Identification Register (Modem—D31:F6)” on page 560 |
| PCI Command | 04h–05h | Section 16.1.3, “PCICMD—PCI Command Register (Modem—D31:F6)” on page 561 |
| PCI Device Status | 06h–07h | Section 15.1.4, “PCISTS—PCI Status Register (Audio—D31:F5)” on page 536 |
| Revision Identification | 08h | Section 16.1.5, “RID—Revision Identification Register (Modem—D31:F6)” on page 562 |
| Programming Interface | 09h | Section 16.1.6, “PI—Programming Interface Register (Modem—D31:F6)” on page 563 |
| Sub Class Code | 0Ah | Section 15.1.7, “SCC—Sub Class Code Register (Audio—D31:F5)” on page 537 |
| Base Class Code | 0Bh | Section 16.1.8, “BCC—Base Class Code Register (Modem—D31:F6)” on page 563 |
| Header Type | 0Eh | Section 16.1.9, “HEADTYP—Header Type Register (Modem—D31:F6)” on page 563 |
| Modem Mixer Base Address | 10h–13h | Section 16.1.10, “MMBAR—Modem Mixer Base Address Register (Modem—D31:F6)” on page 564 |
| Modem Base Address | 14h–17h | Section 16.1.11, “MBAR—Modem Base Address Register (Modem—D31:F6)” on page 564 |
| Subsystem Vendor ID | 2Ch–2Dh | Section 16.1.12, “SVID—Subsystem Vendor Identification Register (Modem—D31:F6)” on page 565 |
| Subsystem ID | 2Eh–2Fh | Section 16.1.13, “SID—Subsystem Identification Register (Modem—D31:F6)” on page 565 |
| Capabilities Pointer | 34h | Section 16.1.14, “CAP_PTR—Capabilities Pointer Register (Modem—D31:F6)” on page 565 |
| Interrupt Line | 3C | Section 16.1.15, “INT_LN—Interrupt Line Register (Modem—D31:F6)” on page 566 |
| Interrupt Pin | 3Dh | Section 16.1.16, “INT_PIN—Interrupt Pin Register (Modem—D31:F6)” on page 566 |
| PCI Power Management ID | 50–51h | Section 16.1.17, “PID—PCI Power Management Capability Identification Register (Modem—D31:F6)” on page 566 |
| PC - Power Management Capabilities | 52–53h | Section 16.1.18, “PC—Power Management Capabilities Register (Modem—D31:F6)” on page 567 |
| Power Management Control and Status | 54–55h | Section 16.1.19, “PCS—Power Management Control and Status Register (Modem—D31:F6)” on page 567 |

## Table 205. Intel® ICH5 Fixed I/O Registers (Sheet 1 of 5)

| Register Name | Port | Datasheet Section and Location |
|---|---|---|
| Channel 0 DMA Base & Current Address Register | 00h | Section 9.2.1, “DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0)” on page 345 |
| Channel 0 DMA Base & Current Count Register | 01h | Section 9.2.2, “DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)” on page 346 |
| Channel 1 DMA Base & Current Address Register | 02h | Section 9.2.1, “DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0)” on page 345 |
| Channel 1 DMA Base & Current Count Register | 03h | Section 9.2.2, “DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)” on page 346 |
| Channel 2 DMA Base & Current Address Register | 04h | Section 9.2.1, “DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0)” on page 345 |
| Channel 2 DMA Base & Current Count Register | 05h | Section 9.2.2, “DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)” on page 346 |
| Channel 3 DMA Base & Current Address Register | 06h | Section 9.2.1, “DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0)” on page 345 |
| Channel 3 DMA Base & Current Count Register | 07h | Section 9.2.2, “DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)” on page 346 |
| Channel 0–3 DMA Command Register<br>Channel 0–3 DMA Status Register | 08h | Section 9.2.4, “DMACMD—DMA Command Register (LPC I/F—D31:F0)” on page 347<br>Section 9.2.5, “DMASTA—DMA Status Register (LPC I/F—D31:F0)” on page 347 |
| Channel 0–3 DMA Write Single Mask Register | 0Ah | Section 9.2.6, “DMA_WRSMSK—DMA Write Single Mask Register (LPC I/F—D31:F0)” on page 348 |
| Channel 0–3 DMA Channel Mode Register | 0Bh | Section 9.2.7, “DMACH_MODE—DMA Channel Mode Register (LPC I/F—D31:F0)” on page 349 |
| Channel 0–3 DMA Clear Byte Pointer Register | 0Ch | Section 9.2.8, “DMA Clear Byte Pointer Register (LPC I/F—D31:F0)” on page 349 |
| Channel 0–3 DMA Master Clear Register | 0Dh | Section 9.2.9, “DMA Master Clear Register (LPC I/F—D31:F0)” on page 350 |
| Channel 0–3 DMA Clear Mask Register | 0Eh | Section 9.2.10, “DMA_CLMSK—DMA Clear Mask Register (LPC I/F—D31:F0)” on page 350 |
| Channel 0–3 DMA Write All Mask Register | 0Fh | Section 9.2.11, “DMA_WRMSK—DMA Write All Mask Register (LPC I/F—D31:F0)” on page 350 |
| **Aliased at 00h–0Fh** | 10h–1Fh | |
| Master PIC ICW1 Init. Cmd Word 1 Register<br>Master PIC OCW2 Op Ctrl Word 2 Register<br>Master PIC OCW3 Op Ctrl Word 3 Register | 20h | Section 9.4.2, “ICW1—Initialization Command Word 1 Register (LPC I/F—D31:F0)” on page 356<br>Section 9.4.8, “OCW2—Operational Control Word 2 Register (LPC I/F—D31:F0)” on page 359<br>Section 9.4.9, “OCW3—Operational Control Word 3 Register (LPC I/F—D31:F0)” on page 360 |
| Master PIC ICW2 Init. Cmd Word 2 Register<br>Master PIC ICW3 Init. Cmd Word 3 Register<br>Master PIC ICW4 Init. Cmd Word 4 Register<br>Master PIC OCW1 Op Ctrl Word 1 Register | 21h | Section 9.4.3, “ICW2—Initialization Command Word 2 Register (LPC I/F—D31:F0)” on page 357<br>Section 9.4.4, “ICW3—Master Controller Initialization Command Word 3 Register (LPC I/F—D31:F0)” on page 357<br>Section 9.4.6, “ICW4—Initialization Command Word 4 Register (LPC I/F—D31:F0)” on page 358<br>Section 9.4.7, “OCW1—Operational Control Word 1 (Interrupt Mask) Register (LPC I/F—D31:F0)” on page 359 |
| **Aliased at 20h–21h** | 24h–25h | |
| **Aliased at 20h–21h** | 28h–29h | |

## Table 205. Intel® ICH5 Fixed I/O Registers (Sheet 2 of 5)

| Register Name | Port | Datasheet Section and Location |
|---|---|---|
| **Aliased at 20h–21h** | 24h–25h | |
| **Aliased at 20h–21h** | 2Ch–2Dh | |
| **Aliased at 20h–21h** | 30h–31h | |
| **Aliased at 20h–21h** | 34h–35h | |
| **Aliased at 20h–21h** | 38h–39h | |
| **Aliased at 20h–21h** | 3Ch–3Dh | |
| Counter 0 Interval Time Status Byte Format<br>Counter 0 Counter Access Port Register | 40h | Section 9.3.2, “SBYTE_FMT—Interval Timer Status Byte Format Register (LPC I/F—D31:F0)” on page 354<br>Section 9.3.3, “Counter Access Ports Register (LPC I/F—D31:F0)” on page 355 |
| Counter 1 Interval Time Status Byte Format<br>Counter 1 Counter Access Port Register | 41h | Section 9.3.2, “SBYTE_FMT—Interval Timer Status Byte Format Register (LPC I/F—D31:F0)” on page 354<br>Section 9.3.3, “Counter Access Ports Register (LPC I/F—D31:F0)” on page 355 |
| Counter 2 Interval Time Status Byte Format<br>Counter 2 Counter Access Port Register | 42h | Section 9.3.2, “SBYTE_FMT—Interval Timer Status Byte Format Register (LPC I/F—D31:F0)” on page 354<br>Section 9.3.3, “Counter Access Ports Register (LPC I/F—D31:F0)” on page 355 |
| Timer Control Word Register<br>Timer Control Word Register Read Back<br>Counter Latch Command | 43h | Section 9.3.1, “TCW—Timer Control Word Register (LPC I/F—D31:F0)” on page 352<br>Section 9.3.1.1, “RDBK_CMD—Read Back Command (LPC I/F—D31:F0)” on page 353<br>Section 9.3.1.2, “LTCH_CMD—Counter Latch Command (LPC I/F—D31:F0)” on page 353 |
| **Aliased at 40h–43h** | 50h–53h | |
| NMI Status and Control Register | 61h | Section 9.7.1, “NMI_SC—NMI Status and Control Register (LPC I/F—D31:F0)” on page 373 |
| NMI Enable Register | 70h | Section 9.7.2, “NMI_EN—NMI Enable (and Real Time Clock Index) Register (LPC I/F—D31:F0)” on page 374 |
| Real-Time Clock (Standard RAM) Index Register | 70h | Section 147, “RTC (Standard) RAM Bank (LPC I/F—D31:F0)” on page 369<br>Section 9.7.2, “NMI_EN—NMI Enable (and Real Time Clock Index) Register (LPC I/F—D31:F0)” on page 374 |
| Real-Time Clock (Standard RAM) Target Register | 71h | Section 147, “RTC (Standard) RAM Bank (LPC I/F—D31:F0)” on page 369 |
| Extended RAM Index Register | 72h | |
| Extended RAM Target Register | 73h | |
| **Aliased at 70h–71h** | 74h–75h | Aliased if U128E bit in RTC Configuration Register is enabled<br>Section 9.1.24, “BACK_CNTL—Backed Up Control Register (LPC I/F—D31:F0)” on page 333 |
| **Aliased at 72h–73h or 70h–71h** | 76h–77h | Aliased to 70h–71h if U128E bit in RTC Configuration Register is enabled<br>Section 9.1.24, “BACK_CNTL—Backed Up Control Register (LPC I/F—D31:F0)” on page 333 |
| Channel 2 DMA Memory Low Page Register | 81h | Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0)” on page 346 |

### Table 205. Intel® ICH5 Fixed I/O Registers (Sheet 3 of 5)

| Register Name | Port | Datasheet Section and Location |
|---|---|---|
| Channel 3 DMA Memory Low Page Register | 82h | Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0)” on page 346 |
| Channel 1 DMA Memory Low Page Register | 83h | Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0)” on page 346 |
| Reserved Page Registers | 84h–86h | |
| Channel 0 DMA Memory Low Page Register | 87h | Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0)” on page 346 |
| Reserved Page Register | 88h | |
| Channel 6 DMA Memory Low Page Register | 89h | Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0)” on page 346 |
| Channel 7 DMA Memory Low Page Register | 8Ah | Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0)” on page 346 |
| Channel 5 DMA Memory Low Page Register | 8Bh | Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers (LPC I/F—D31:F0)” on page 346 |
| Reserved Page Registers | 8Ch–8Eh | |
| Refresh Low Page Register | 8Fh | |
| **Aliased at 81h–8Fh** | 91h–9Fh (except 92h) | |
| Fast A20 and INIT Register | 92h | Section 9.7.3, “PORT92—Fast A20 and Init Register (LPC I/F—D31:F0)” on page 374 |
| Slave PIC ICW1 Init. Cmd Word 1 Register<br>Slave PIC OCW2 Op Ctrl Word 2 Register<br>Slave PIC OCW3 Op Ctrl Word 3 Register | A0h | Section 9.4.2, “ICW1—Initialization Command Word 1 Register (LPC I/F—D31:F0)” on page 356<br>Section 9.4.8, “OCW2—Operational Control Word 2 Register (LPC I/F—D31:F0)” on page 359<br>Section 9.4.9, “OCW3—Operational Control Word 3 Register (LPC I/F—D31:F0)” on page 360 |
| Slave PIC ICW2 Init. Cmd Word 2 Register<br>Slave PIC ICW3 Init. Cmd Word 3 Register<br>Slave PIC ICW4 Init. Cmd Word 4 Register<br>Slave PIC OCW1 Op Ctrl Word 1 Register | A1 | Section 9.4.3, “ICW2—Initialization Command Word 2 Register (LPC I/F—D31:F0)” on page 357<br>Section 9.4.4, “ICW3—Master Controller Initialization Command Word 3 Register (LPC I/F—D31:F0)” on page 357<br>Section 9.4.6, “ICW4—Initialization Command Word 4 Register (LPC I/F—D31:F0)” on page 358<br>Section 9.4.7, “OCW1—Operational Control Word 1 (Interrupt Mask) Register (LPC I/F—D31:F0)” on page 359 |
| **Aliased at A0h–A1h** | A4h–A5h | |
| **Aliased at A0h–A1h** | A8h–A9h | |
| **Aliased at A0h–A1h** | ACh–ADh | |
| **Aliased at A0h–A1h** | B0h–B1h | |
| Advanced Power Management Control Port Register | B2h | Section 9.9.1, “APM_CNT—Advanced Power Management Control Port Register” on page 384 |
| Advanced Power Management Status Port Register | B3h | Section 9.9.2, “APM_STS—Advanced Power Management Status Port Register” on page 384 |
| **Aliased at A0h–A1h** | B4h–B5h | |
| **Aliased at A0h–A1h** | B8h–B9h | |
| **Aliased at A0h–A1h** | BCh–BDh | |

### Table 205. Intel® ICH5 Fixed I/O Registers (Sheet 4 of 5)

| Register Name | Port | Datasheet Section and Location |
|---|---|---|
| Channel 4 DMA Base & Current Address Register | C0h | Section 9.2.1, “DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0)” on page 345 |
| **Aliased at C0h** | C1h | |
| Channel 4 DMA Base & Current Count Register | C2h | Section 9.2.2, “DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)” on page 346 |
| **Aliased at C2h** | C3h | |
| Channel 5 DMA Base & Current Address Register | C4h | Section 9.2.1, “DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0)” on page 345 |
| **Aliased at C4h** | C5h | |
| Channel 5 DMA Base & Current Count Register | C6h | Section 9.2.2, “DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)” on page 346 |
| **Aliased at C6h** | C7h | |
| Channel 6 DMA Base & Current Address Register | C8h | Section 9.2.1, “DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0)” on page 345 |
| **Aliased at C8h** | C9h | |
| Channel 6 DMA Base & Current Count Register | CAh | Section 9.2.2, “DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)” on page 346 |
| **Aliased at CAh** | CBh | |
| Channel 7 DMA Base & Current Address Register | CCh | Section 9.2.1, “DMABASE_CA—DMA Base and Current Address Registers (LPC I/F—D31:F0)” on page 345 |
| **Aliased at CCh** | CDh | |
| Channel 7 DMA Base & Current Count Register | CEh | Section 9.2.2, “DMABASE_CC—DMA Base and Current Count Registers (LPC I/F—D31:F0)” on page 346 |
| **Aliased at CEh** | CFh | |
| Channel 4–7 DMA Command Register<br>Channel 4–7 DMA Status Register | D0h | Section 9.2.4, “DMACMD—DMA Command Register (LPC I/F—D31:F0)” on page 347<br>Section 9.2.5, “DMASTA—DMA Status Register (LPC I/F—D31:F0)” on page 347 |
| **Aliased at D0h** | D1h | |
| Channel 4–7 DMA Write Single Mask Register | D4h | Section 9.2.6, “DMA_WRSMSK—DMA Write Single Mask Register (LPC I/F—D31:F0)” on page 348 |
| **Aliased at D4h** | D5h | |
| Channel 4–7 DMA Channel Mode Register | D6h | Section 9.2.7, “DMACH_MODE—DMA Channel Mode Register (LPC I/F—D31:F0)” on page 349 |
| **Aliased at D6h** | D7h | |
| Channel 4–7 DMA Clear Byte Pointer Register | D8h | Section 9.2.8, “DMA Clear Byte Pointer Register (LPC I/F—D31:F0)” on page 349 |
| **Aliased at D8h** | D9h | |
| Channel 4–7 DMA Master Clear Register | DAh | Section 9.2.9, “DMA Master Clear Register (LPC I/F—D31:F0)” on page 350 |
| **Aliased at DAh** | DBh | |
| Channel 4–7 DMA Clear Mask Register | DCh | Section 9.2.10, “DMA_CLMSK—DMA Clear Mask Register (LPC I/F—D31:F0)” on page 350 |
| **Aliased at DCh** | DEh | |
| Channel 4–7 DMA Write All Mask Register | DEh | Section 9.2.11, “DMA_WRMSK—DMA Write All Mask Register (LPC I/F—D31:F0)” on page 350 |

### Table 205. Intel® ICH5 Fixed I/O Registers (Sheet 5 of 5)

| Register Name | Port | Datasheet Section and Location |
|---|---|---|
| **Aliased at DEh** | DFh | |
| Coprocessor Error Register | F0h | Section 9.7.4, “COPROC_ERR—Coprocessor Error Register (LPC I/F—D31:F0)” on page 375 |
| PIO Mode Command Block Offset for Secondary Drive | 170h–177h | See the *AT Attachment - 6 with Packet Interface (ATA/ATAPI - 6)* for detailed register description |
| PIO Mode Command Block Offset for Primary Drive | 1F0h–1F7h | See the *AT Attachment - 6 with Packet Interface (ATA/ATAPI - 6)* for detailed register description |
| PIO Mode Control Block Offset for Secondary Drive | 376h | See the *AT Attachment - 6 with Packet Interface (ATA/ATAPI - 6)* for detailed register description |
| PIO Mode Control Block Offset for Primary Drive | 3F6h | See the *AT Attachment - 6 with Packet Interface (ATA/ATAPI - 6)* for detailed register description |
| Master PIC Edge/Level Triggered Register | 4D0h | Section 9.4.10, “ELCR1—Master Controller Edge/Level Triggered Register (LPC I/F—D31:F0)” on page 361 |
| Slave PIC Edge/Level Triggered Register | 4D1h | Section 9.4.11, “ELCR2—Slave Controller Edge/Level Triggered Register (LPC I/F—D31:F0)” on page 362 |
| Reset Control Register | CF9h | Section 9.7.5, “RST_CNT—Reset Control Register (LPC I/F—D31:F0)” on page 375 |

**NOTE:** When the POS_DEC_EN bit is set, additional I/O ports get positively decoded by the ICH5.

## Table 206. Intel® ICH5 Variable I/O Registers (Sheet 1 of 6)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| **LAN Control/Status Registers (CSR) may be mapped to either I/O space or memory space. LAN CSR at CSR_IO_BASE + Offset or CSR_MEM_BASE + Offset. CSR_MEM_BASE set in Section 7.1.11, "CSR_MEM_BASE — CSR Memory-Mapped Base Address Register (LAN Controller—B1:D8:F0)" on page 281 CSR_IO_BASE set in Section 7.1.12, "CSR_IO_BASE — CSR I/O-Mapped Base Address Register (LAN Controller—B1:D8:F0)" on page 281** | | |
| SCB Status Word | 01h–00h | Section 7.2.1, "SCB_STA—System Control Block Status Word Register (LAN Controller—B1:D8:F0)" on page 288 |
| SCB Command Word | 03h–02h | Section 7.2.2, "SCB_CMD—System Control Block Command Word Register (LAN Controller—B1:D8:F0)" on page 289 |
| SCB General Pointer | 07h–04h | Section 7.2.3, "SCB_GENPNT—System Control Block General Pointer Register (LAN Controller—B1:D8:F0)" on page 291 |
| PORT | OBh–08h | Section 7.2.4, "PORT—PORT Interface Register (LAN Controller—B1:D8:F0)" on page 291 |
| EEPROM Control Register | 0Fh–0Eh | Section 7.2.5, "EEPROM_CNTL—EEPROM Control Register (LAN Controller—B1:D8:F0)" on page 292 |
| MDI Control Register | 13h–10h | Section 7.2.6, "MDI_CNTL—Management Data Interface (MDI) Control Register (LAN Controller—B1:D8:F0)" on page 293 |
| Receive DMA Byte Count | 17h–14h | Section 7.2.7, "REC_DMA_BC—Receive DMA Byte Count Register (LAN Controller—B1:D8:F0)" on page 293 |
| Early Receive Interrupt | 18h | Section 7.2.8, "EREC_INTR—Early Receive Interrupt Register (LAN Controller—B1:D8:F0)" on page 294 |
| Flow Control Register | 1Ah–19h | Section 7.2.9, "FLOW_CNTL—Flow Control Register (LAN Controller—B1:D8:F0)" on page 295 |
| PMDR | 1Bh | Section 7.2.10, "PMDR—Power Management Driver Register (LAN Controller—B1:D8:F0)" on page 296 |
| General Control | 1Ch | Section 7.2.11, "GENCNTL—General Control Register (LAN Controller—B1:D8:F0)" on page 297 |
| General Status | 1Dh | Section 7.2.12, "GENSTA—General Status Register (LAN Controller—B1:D8:F0)" on page 297 |
| **Power Management I/O Registers at PMBASE+Offset** | | |
| PM1 Status | 00–01h | Section 9.10.1, "PM1_STS—Power Management 1 Status Register" on page 386 |
| PM1 Enable | 02–03h | Section 9.10.2, "PM1_EN—Power Management 1 Enable Register" on page 388 |
| PM1 Control | 04–07h | Section 9.10.3, "PM1_CNT—Power Management 1 Control" on page 389 |
| PM1 Timer | 08–0Bh | Section 9.10.4, "PM1_TMR—Power Management 1 Timer Register" on page 390 |
| Processor Control | 10h–13h | Section 9.10.5, "PROC_CNT—Processor Control Register" on page 390 |
| General Purpose Event 0 Status | 28–2Bh | Section 9.10.6, "GPE0_STS—General Purpose Event 0 Status Register" on page 392 |
| General Purpose Event 0 Enables | 2C–2Fh | Section 9.10.7, "GPE0_EN—General Purpose Event 0 Enables Register" on page 394 |
| SMI# Control and Enable | 30–31h | Section 9.10.8, "SMI_EN—SMI Control and Enable Register" on page 396 |
| SMI Status Register | 34–35h | Section 9.10.9, "SMI_STS—SMI Status Register" on page 398 |
| Alternate GPI SMI Enable | 38–39h | Section 9.10.10, "ALT_GP_SMI_EN—Alternate GPI SMI Enable Register" on page 400 |
| Alternate GPI SMI Status | 3A–3Bh | Section 9.10.11, "ALT_GP_SMI_STS—Alternate GPI SMI Status Register" on page 400 |

### Table 206. Intel® ICH5 Variable I/O Registers (Sheet 2 of 6)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Monitor SMI Status | 40h | Section 9.10.12, “MON_SMI—Device Monitor SMI Status and Enable Register” on page 400 |
| Device Activity Status | 44h | Section 9.10.13, “DEVACT_STS — Device Activity Status Register” on page 401 |
| Device Trap Enable | 48h | Section 9.10.14, “DEVTRAP_EN—Device Trap Enable Register” on page 402 |
| **TCO I/O Registers at TCOBASE + Offset**<br>TCOBASE = PMBASE + 40h | | |
| TCO_RLD: TCO Timer Reload and Current Value | 00h | Section 9.11.1, “TCO_RLD—TCO Timer Reload and Current Value Register” on page 403 |
| TCO_TMR: TCO Timer Initial Value | 01h | Section 9.11.2, “TCO_TMR—TCO Timer Initial Value Register” on page 404 |
| TCO_DAT_IN: TCO Data In | 02h | Section 9.11.3, “TCO_DAT_IN—TCO Data In Register” on page 404 |
| TCO_DAT_OUT: TCO Data Out | 03h | Section 9.11.4, “TCO_DAT_OUT—TCO Data Out Register” on page 404 |
| TCO1_STS: TCO Status | 04h–05h | Section 9.11.5, “TCO1_STS—TCO1 Status Register” on page 405 |
| TCO2_STS: TCO Status | 06h–07h | Section 9.11.6, “TCO2_STS—TCO2 Status Register” on page 406 |
| TCO1_CNT: TCO Control | 08h–09h | Section 9.11.7, “TCO1_CNT—TCO1 Control Register” on page 407 |
| TCO2_CNT: TCO Control | 0Ah–0Bh | Section 9.11.8, “TCO2_CNT—TCO2 Control Register” on page 408 |
| **GPIO I/O Registers at GPIOBASE + Offset** | | |
| GPIO Use Select | 00–03h | Section 9.12.1, “GPIO_USE_SEL—GPIO Use Select Register” on page 410 |
| GPIO Input/Output Select | 04–07h | Section 9.12.2, “GP_IO_SEL—GPIO Input/Output Select Register” on page 410 |
| GPIO Level for Input or Output | 0C–0Fh | Section 9.12.3, “GP_LVL—GPIO Level for Input or Output Register” on page 411 |
| GPIO Blink Enable | 18–1Bh | Section 9.12.4, “GPO_BLINK—GPO Blink Enable Register” on page 411 |
| GPIO Signal Invert | 2C–2Fh | Section 9.12.5, “GPI_INV—GPIO Signal Invert Register” on page 412 |

## Table 206. Intel® ICH5 Variable I/O Registers (Sheet 3 of 6)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| **BMIDE I/O Registers at BM_BASE + Offset**<br>BM_BASE is set at Section 10.1.12, “SCMD_BAR—Secondary Command Block Base Address Register (IDE D31:F1)” on page 421 | | |
| Command Register Primary | 00 | Section 10.2.1, “BMIC[P,S]—Bus Master IDE Command Register (IDE—D31:F1)” on page 431 |
| Status Register Primary | 02 | Section 10.2.2, “BMIS[P,S]—Bus Master IDE Status Register (IDE—D31:F1)” on page 432 |
| Descriptor Table Pointer Primary | 04–07 | Section 10.2.3, “BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register (IDE—D31:F1)” on page 432 |
| Command Register Secondary | 08 | Section 10.2.1, “BMIC[P,S]—Bus Master IDE Command Register (IDE—D31:F1)” on page 431 |
| Status Register Secondary | 0A | Section 10.2.2, “BMIS[P,S]—Bus Master IDE Status Register (IDE—D31:F1)” on page 432 |
| Descriptor Table Pointer Secondary | 0C–0F | Section 10.2.3, “BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register (IDE—D31:F1)” on page 432 |
| **USB I/O Registers at Base Address + Offset**<br>**USB Base Address is set at Section 12.1.10, “BASE—Base Address Register (USB—D29:F0/F1/F2/F3)” on page 465** | | |
| USB Command Register | 00–01 | Section 12.2.1, “USBCMD—USB Command Register” on page 471 |
| USB Status Register | 02–03 | Section 12.2.2, “USBSTS—USB Status Register” on page 474 |
| USB Interrupt Enable | 04–05 | Section 12.2.3, “USBINTR—USB Interrupt Enable Register” on page 475 |
| USB Frame Number | 06–07 | Section 12.2.4, “FRNUM—Frame Number Register” on page 475 |
| USB Frame List Base Address | 08–0B | Section 12.2.5, “FRBASEADD—Frame List Base Address Register” on page 476 |
| USB Start of Frame Modify | 0C | Section 12.2.6, “SOFMOD—Start of Frame Modify Register” on page 476 |
| Port 0, 2, 4 Status/Control | 10–11 | Section 12.2.7, “PORTSC[0,1]—Port Status and Control Register” on page 477 |
| Port 1, 3, 5 Status/Control | 12–13 | Section 12.2.7, “PORTSC[0,1]—Port Status and Control Register” on page 477 |
| Loop Back Test Data | 18h | |
| **SMBus I/O Registers at SMB_BASE + Offset**<br>SMB_BASE is set at Section 14.1.8, “SMB_BASE—SMBUS Base Address Register (SMBUS—D31:F3)” on page 518 | | |
| Host Status | 00h | Section 14.2.1, “HST_STS—Host Status Register (SMBUS—D31:F3)” on page 522 |
| Host Control | 02h | Section 14.2.2, “HST_CNT—Host Control Register (SMBUS—D31:F3)” on page 523 |
| Host Command | 03h | Section 14.2.3, “HST_CMD—Host Command Register (SMBUS—D31:F3)” on page 525 |
| Transmit Slave Address | 04h | Section 14.2.4, “XMIT_SLVA—Transmit Slave Address Register (SMBUS—D31:F3)” on page 525 |
| Host Data 0 | 05h | Section 14.2.5, “HST_D0—Host Data 0 Register (SMBUS—D31:F3)” on page 525 |
| Host Data 1 | 06h | Section 14.2.6, “HST_D1—Host Data 1 Register (SMBUS—D31:F3)” on page 525 |
| Block Data Byte | 07h | Section 14.2.7, “Host_BLOCK_DB—Host Block Data Byte Register (SMBUS—D31:F3)” on page 526 |

### Table 206. Intel® ICH5 Variable I/O Registers (Sheet 4 of 6)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Packet Error Check | 08h | Section 14.2.8, “PEC—Packet Error Check (PEC) Register (SMBUS—D31:F3)” on page 526 |
| Receive Slave Address | 09h | Section 14.2.9, “RCV_SLVA—Receive Slave Address Register (SMBUS—D31:F3)” on page 527 |
| Receive Slave Data | 0Ah | Section 14.2.10, “SLV_DATA—Receive Slave Data Register (SMBUS—D31:F3)” on page 527 |
| Auxiliary Status | 0Ch | Section 14.2.11, “AUX_STS—Auxiliary Status Register (SMBUS—D31:F3)” on page 527 |
| Auxiliary Control | 0Dh | Section 14.2.12, “AUX_CTL—Auxiliary Control Register (SMBUS—D31:F3)” on page 528 |
| **AC’97 Audio I/O Registers at NAMBAR + Offset NAMBAR is set at Section 15.1.11, “NABMBAR—Native Audio Bus Mastering Base Address Register (Audio—D31:F5)” on page 539** | | |
| PCM In Buffer Descriptor list Base Address Register | 00h | Section 15.2.1, “x_BDBAR—Buffer Descriptor Base Address Register (Audio—D31:F5)” on page 549 |
| PCM In Current Index Value | 04h | Section 15.2.2, “x_CIV—Current Index Value Register (Audio—D31:F5)” on page 550 |
| PCM In Last Valid Index | 05h | Section 15.2.3, “x_LVI—Last Valid Index Register (Audio—D31:F5)” on page 550 |
| PCM In Status Register | 06h | Section 15.2.4, “x_SR—Status Register (Audio—D31:F5)” on page 551 |
| PCM In Position In Current Buffer | 08h | Section 15.2.5, “x_PICB—Position In Current Buffer Register (Audio—D31:F5)” on page 552 |
| PCM In Prefetched Index Value | 0Ah | Section 15.2.6, “x_PIV—Prefetched Index Value Register (Audio—D31:F5)” on page 552 |
| PCM In Control Register | 0Bh | Section 15.2.7, “x_CR—Control Register (Audio—D31:F5)” on page 553 |
| PCM Out Buffer Descriptor list Base Address Register | 10h | Section 15.2.1, “x_BDBAR—Buffer Descriptor Base Address Register (Audio—D31:F5)” on page 549 |
| PCM Out Current Index Value | 14h | Section 15.2.2, “x_CIV—Current Index Value Register (Audio—D31:F5)” on page 550 |
| PCM Out Last Valid Index | 15h | Section 15.2.3, “x_LVI—Last Valid Index Register (Audio—D31:F5)” on page 550 |
| PCM Out Status Register | 16h | Section 15.2.4, “x_SR—Status Register (Audio—D31:F5)” on page 551 |
| PCM Out Position In Current Buffer | 18h | Section 15.2.5, “x_PICB—Position In Current Buffer Register (Audio—D31:F5)” on page 552 |
| PCM Out Prefetched Index Value | 1Ah | Section 15.2.6, “x_PIV—Prefetched Index Value Register (Audio—D31:F5)” on page 552 |
| PCM Out Control Register | 1Bh | Section 15.2.7, “x_CR—Control Register (Audio—D31:F5)” on page 553 |
| Mic. In Buffer Descriptor list Base Address Register | 20h | Section 15.2.1, “x_BDBAR—Buffer Descriptor Base Address Register (Audio—D31:F5)” on page 549 |
| Mic. In Current Index Value | 24h | Section 15.2.2, “x_CIV—Current Index Value Register (Audio—D31:F5)” on page 550 |
| Mic. In Last Valid Index | 25h | Section 15.2.3, “x_LVI—Last Valid Index Register (Audio—D31:F5)” on page 550 |
| Mic. In Status Register | 26h | Section 15.2.4, “x_SR—Status Register (Audio—D31:F5)” on page 551 |
| Mic In Position In Current Buffer | 28h | Section 15.2.5, “x_PICB—Position In Current Buffer Register (Audio—D31:F5)” on page 552 |

## Table 206. Intel® ICH5 Variable I/O Registers (Sheet 5 of 6)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| Mic. In Prefetched Index Value | 2Ah | Section 15.2.6, “x_PIV—Prefetched Index Value Register (Audio—D31:F5)” on page 552 |
| Mic. In Control Register | 2Bh | Section 15.2.7, “x_CR—Control Register (Audio—D31:F5)” on page 553 |
| Global Control | 2Ch | Section 15.2.8, “GLOB_CNT—Global Control Register (Audio—D31:F5)” on page 554 |
| Global Status | 30h | Section 15.2.9, “GLOB_STA—Global Status Register (Audio—D31:F5)” on page 556 |
| Codec Access Semaphore Register | 34h | Section 15.2.10, “CAS—Codec Access Semaphore Register (Audio—D31:F5)” on page 558 |
| **AC’97 Audio I/O Registers at MBBAR + Offset MBBAR is set at Section 15.1.13, “MBBAR—Bus Master Base Address Register (Audio—D31:F5)” on page 540** | | |
| Mic. 2 Buffer Descriptor list Base Address Register | 40–43h | Section 15.2.1, “x_BDBAR—Buffer Descriptor Base Address Register (Audio—D31:F5)” on page 549 |
| Mic. 2 Current Index Value | 44h | Section 15.2.2, “x_CIV—Current Index Value Register (Audio—D31:F5)” on page 550 |
| Mic. 2 Last Valid Index | 45h | Section 15.2.3, “x_LVI—Last Valid Index Register (Audio—D31:F5)” on page 550 |
| Mic. 2 Status Register | 46–47h | Section 15.2.4, “x_SR—Status Register (Audio—D31:F5)” on page 551 |
| Mic 2 Position In Current Buffer | 48–49h | Section 15.2.5, “x_PICB—Position In Current Buffer Register (Audio—D31:F5)” on page 552 |
| Mic. 2 Prefetched Index Value | 4Ah | Section 15.2.6, “x_PIV—Prefetched Index Value Register (Audio—D31:F5)” on page 552 |
| Mic. 2 Control Register | 4Bh | Section 15.2.7, “x_CR—Control Register (Audio—D31:F5)” on page 553 |
| PCM In 2 Buffer Descriptor list Base Address Register | 50–53h | Section 15.2.1, “x_BDBAR—Buffer Descriptor Base Address Register (Audio—D31:F5)” on page 549 |
| PCM In 2 Current Index Value | 54h | Section 15.2.2, “x_CIV—Current Index Value Register (Audio—D31:F5)” on page 550 |
| PCM In 2 Last Valid Index | 55h | Section 15.2.3, “x_LVI—Last Valid Index Register (Audio—D31:F5)” on page 550 |
| PCM In 2 Status Register | 56–57h | Section 15.2.4, “x_SR—Status Register (Audio—D31:F5)” on page 551 |
| PCM In 2 Position In Current Buffer | 58–59h | Section 15.2.5, “x_PICB—Position In Current Buffer Register (Audio—D31:F5)” on page 552 |
| PCM In 2 Prefetched Index Value | 5Ah | Section 15.2.6, “x_PIV—Prefetched Index Value Register (Audio—D31:F5)” on page 552 |
| PCM In 2 Control Register | 5Bh | Section 15.2.7, “x_CR—Control Register (Audio—D31:F5)” on page 553 |
| S/PDIF Buffer Descriptor list Base Address Register | 60–63 | Section 15.2.1, “x_BDBAR—Buffer Descriptor Base Address Register (Audio—D31:F5)” on page 549 |
| S/PDIF Current Index Value | 64h | Section 15.2.2, “x_CIV—Current Index Value Register (Audio—D31:F5)” on page 550 |
| S/PDIF Last Valid Index | 65h | Section 15.2.3, “x_LVI—Last Valid Index Register (Audio—D31:F5)” on page 550 |
| S/PDIF Status Register | 66–67h | Section 15.2.4, “x_SR—Status Register (Audio—D31:F5)” on page 551 |
| S/PDIF Position In Current Buffer | 68–69h | Section 15.2.5, “x_PICB—Position In Current Buffer Register (Audio—D31:F5)” on page 552 |

### Table 206. Intel® ICH5 Variable I/O Registers (Sheet 6 of 6)

| Register Name | Offset | Datasheet Section and Location |
|---|---|---|
| S/PDIF Prefetched Index Value | 6Ah | Section 15.2.6, “x_PIV—Prefetched Index Value Register (Audio—D31:F5)” on page 552 |
| S/PDIF Control Register | 6Bh | Section 15.2.7, “x_CR—Control Register (Audio—D31:F5)” on page 553 |
| SDATA_IN Map Register | 80 | Section 15.2.11, “SDM—SDATA_IN Map Register (Audio—D31:F5)” on page 558 |
| **AC’97 Modem I/O Registers at MBAR + Offset MBAR is set in Section 16.1.11, “MBAR—Modem Base Address Register (Modem—D31:F6)” on page 564** | | |
| Modem In Buffer Descriptor List Base Address Register | 00h | Section 16.2.1, “x_BDBAR—Buffer Descriptor List Base Address Register (Modem—D31:F6)” on page 570 |
| Modem In Current Index Value Register | 04h | Section 16.2.2, “x_CIV—Current Index Value Register (Modem—D31:F6)” on page 570 |
| Modem In Last Valid Index Register | 05h | Section 16.2.3, “x_LVI—Last Valid Index Register (Modem—D31:F6)” on page 571 |
| Modem In Status Register | 06h | Section 16.2.4, “x_SR—Status Register (Modem—D31:F6)” on page 572 |
| Modem In Position In Current Buffer Register | 08h | Section 16.2.5, “x_PICB—Position in Current Buffer Register (Modem—D31:F6)” on page 573 |
| Modem In Prefetch Index Value Register | 0Ah | Section 16.2.6, “x_PIV—Prefetch Index Value Register (Modem—D31:F6)” on page 573 |
| Modem In Control Register | 0Bh | Section 16.2.7, “x_CR—Control Register (Modem—D31:F6)” on page 574 |
| Modem Out Buffer Descriptor List Base Address Register | 10h | Section 16.2.1, “x_BDBAR—Buffer Descriptor List Base Address Register (Modem—D31:F6)” on page 570 |
| Modem Out Current Index Value Register | 14h | Section 16.2.2, “x_CIV—Current Index Value Register (Modem—D31:F6)” on page 570 |
| Modem Out Last Valid Register | 15h | Section 16.2.3, “x_LVI—Last Valid Index Register (Modem—D31:F6)” on page 571 |
| Modem Out Status Register | 16h | Section 16.2.4, “x_SR—Status Register (Modem—D31:F6)” on page 572 |
| Modem In Position In Current Buffer Register | 18h | Section 16.2.5, “x_PICB—Position in Current Buffer Register (Modem—D31:F6)” on page 573 |
| Modem Out Prefetched Index Register | 1Ah | Section 16.2.6, “x_PIV—Prefetch Index Value Register (Modem—D31:F6)” on page 573 |
| Modem Out Control Register | 1Bh | Section 16.2.7, “x_CR—Control Register (Modem—D31:F6)” on page 574 |
| Global Control | 3Ch | Section 16.2.8, “GLOB_CNT—Global Control Register (Modem—D31:F6)” on page 575 |
| Global Status | 40h | Section 16.2.9, “GLOB_STA—Global Status Register (Modem—D31:F6)” on page 576 |
| Codec Access Semaphore Register | 44h | Section 16.2.10, “CAS—Codec Access Semaphore Register (Modem—D31:F6)” on page 578 |

# Register Bit Index B


**Numerics**

12-Clock Retry Enable 314
Position In Current Buffer 573
64 Bit Address Capable 450
64-bit Addressing Capability 498
66 MHz Capable 278, 304, 308, 320, 463, 482, 517, 536, 562
66MHz Capable 418, 436

**A**

A20Gate Pass-Through Enable 469
AC '97 Interrupt Routing 542
AC ’97 Cold Reset# 555, 575
AC ’97 Warm Reset 555, 575
AC_SDIN0 Codec Ready 557, 577
AC_SDIN0 Interrupt Enable 554
AC_SDIN0 Resume Interrupt 557, 577
AC_SDIN0 Resume Interrupt Enable 575
AC_SDIN1 Codec Ready 557, 577
AC_SDIN1 Interrupt Enable 554
AC_SDIN1 Resume Interrupt 557, 577
AC_SDIN1 Resume Interrupt Enable 575
AC_SDIN2 Codec Ready 556, 576
AC_SDIN2 Interrupt Enable 554, 575
AC_SDIN2 Resume Interrupt 556, 576
AC97_EN 395
AC97_STS 393
ACLINK Shut Off 554, 575
ACPI Enable 323
AD3 556, 576
Address 450, 525
Address Increment/Decrement Select 349
Address of Descriptor Table 432, 459
ADI 356
AFTERG3_EN 379
Alarm Flag 372
Alarm Interrupt Enable 371
Alternate A20 Gate 374
Alternate Access Mode Enable 331
Alternate GPI SMI Enable 400
Alternate GPI SMI Status 400
APIC Data 364
APIC Enable 331
APIC ID 365
APIC Index 363
APM_STS 399
APMC_EN 396
ASF Enabled 296
Asynchronous Schedule Enable 500
Asynchronous Schedule Park Capability 498
Asynchronous Schedule Status 502
Autoinitialize Enable 349
Automatic End of Interrupt 358
Automatically Append CRC 528
Auxiliary Current 284, 448, 487, 544, 567

**B**

BAR Number 489
Base Address 281, 322, 325, 420, 421, 422, 438, 439, 440, 465, 476, 484, 518, 538, 539, 540, 564
Base Address (Low) 506
Base and Current Address 345
Base and Current Count 346
Base Class Code 279, 305, 321, 420, 438, 464, 483, 518, 537, 563
Binary/BCD Countdown Select 352
BIOS Lock Enable 324
BIOS Release 396
BIOS Write Enable 324
BIOS_EN 397
BIOS_STS 399
BIOSWR_STS 405
BIST FIS Failed 454
BIST FIS Parameters 454
BIST FIS Successful 454
BIST FIS Transmit Data 1 455
BIST FIS Transmit Data 2 455
Bit 1 of Slot 12 577
Bit 1 of slot 12 557
Bit 2 of Slot 12 557, 577
Bit 3 of Slot 12 557, 577
Bit Clock Stopped 556, 576
Block Data 526
BOOT_STS 406
Buffer Completion Interrupt Status 551, 572
Buffer Descriptor Base Address(31-3) 549
Buffer Descriptor List Base Address (31-3) 570
Buffered Mode 358
Bus Master Enable 277, 303, 319, 417, 435, 462,

481, 516, 535, 561
Bus Master IDE Active 432, 458
BUS_ERR 522
Byte Done Status 522

**C**

Cache Line Size 280
Capabilities List 278, 436, 463, 482, 562
Capabilities List Exists 536
Capabilities List Indicator 517
Capabilities Pointer 282, 441, 485, 541, 565
Capability ID 283, 448, 449, 491, 543, 566
Capability Register Length Value 496
Cascaded Interrupt Controller IRQ Connection 357
Channel 0 Select 329
Channel 1 Select 329
Channel 2 Select 329
Channel 3 Select 329
Channel 5 Select 329
Channel 6 Select 329
Channel 7 Select 329
Channel Mask Bits 350
Channel Mask Select 348
Channel Request Status 347
Channel Terminal Count Status 347
Clear Byte Pointer 349
Clear Mask Register 350
CNA Mask 289
CNF1_LPC_EN 337
CNF2_LPC_EN 337
Codec Access Semaphore 558, 578
COMA Decode Range 335
COMA_LPC_EN 338
COMB Decode Range 335
COMB_LPC_EN 338
Command Unit 288
Command Unit Command 290
Command Unit Status 288
Configuration Layout 465
Configure Flag 471, 507
Connect Status Change 478, 511
Coprocessor Error 375
Coprocessor Error Enable 330
Count Register Status 354
COUNT_SIZE_CAP 581
Countdown Type Status 354
Counter 0 Select 353
Counter 1 Select 353
Counter 2 Select 353
Counter Latch Command 353
Counter Mode Selection 352
Counter OUT Pin State 354
Counter Port 355
Counter Select 352
Counter Selection 353
Counter Value 582
CPU Configuration 330
CPU Frequency Strap 333
CPU Power Failure 378
CPU SLP# Enable 377
CPU Thermal Trip Status 378
CRC Error 527
CU Not Active 288
Current Connect Status 478, 511
Current Equals Last Valid 551, 572
Current Index Value (4-0) 550, 570
CX Mask 289

**D**

D1 Support 284, 448, 487
D2 Support 284, 448, 487
D29_F0_Disable 342
D29_F1_Disable 342
D29_F2_Disable 342
D29_F3_Disable 342
D29_F7_Disable 342
D31_F1_Disable 343
D31_F2_Disable 343
D31_F3_Disable 343
D31_F5_Disable 343
D31_F6_Disable 342
Data 293, 450, 452
Data Message Byte 0 527
Data Message Byte 1 527
Data Mode 371
Data Parity Error Detected 278, 320, 418, 436, 463, 517, 536, 562
Data Scale 285, 488
Data Select 285, 488
DATA_HIGH_BYTE 531
DATA_LEN_CNT 513
DATA_LOW_BYTE 531
Data0/Count 525
Data1 525
DATABUFFER(63-0) 514
Date Alarm 372
Daylight Savings Enable 371
Debug Port Capability ID 488
Debug Port Number 497
Debug Port Offset 489

Deep Power-Down on Link Down Enable 297
Delayed Transaction Enable 331
Delivery Mode 368
Delivery Status 367
Destination 367
Destination Mode 368
Detected Parity Error 278, 304, 308, 320, 418, 436, 482, 517, 536, 562
DEV(7-4)_TRAP_EN 400
DEV(7-4)_TRAP_STS 400
DEV_ERR 523
Device Connects 224
Device ID 276, 302, 318, 416, 434, 462, 480, 516, 534, 560
Device Monitor Status 398
Device Partial/Slumber Request Port 0 453
Device Partial/Slumber Request Port 1 453
Device Specific Initialization 284, 448, 487, 544, 567
DEVICE_ADDRESS 530
DEVSEL# Timing Status 278, 304, 308, 320, 418, 436, 463, 482, 517, 536, 562
Diagnose Result 292
Discard Timer SERR# Enable 311
Discard Timer Status 311
Division Chain Select 370
DMA Channel Group Enable 347
DMA Channel Select 348, 349
DMA Collection Buffer Enable 331
DMA Controller Halted 551, 572
DMA Group Arbitration Priority 347
DMA Low Page 346
DMA Transfer Mode 349
DMA Transfer Type 349
DONE_STS 512
DRAM Initialization Bit 378
Drive 0 DMA Capable 432, 458
Drive 0 DMA Timing Enable 425, 443
Drive 0 Fast Timing Bank 425, 443
Drive 0 IORDY Sample Point Enable 425, 443
Drive 0 Prefetch/Posting Enable 425, 443
Drive 1 DMA Capable 432, 458
Drive 1 DMA Timing Enable 424, 442
Drive 1 Fast Timing Bank 425, 443
Drive 1 IORDY Sample Point Enable 424, 442
Drive 1 Prefetch/Posting Enable 424, 442
Drive 1 Timing Register Enable 424, 442
Duplex Mode 297
Dynamic Data 285

**E**

Early Receive 288
Early Receive Count 294
Edge/Level Bank Select (LTIM) 356
EEPROM Chip Select 292
EEPROM Serial Clock 292
EEPROM Serial Data In 292
EEPROM Serial Data Out 292
EHC Initialization 215
EHC Resets 216
EHCI Extended Capabilities Pointer 498
Enable 32-Byte Buffer 528
Enable Special Mask Mode 360
ENABLED_CNT 512
Enables CPU BIST 333
End of SMI 397
Enter Global Suspend Mode 471
ER Mask 289
Error 432, 458
ERROR_GOOD#_STS 513
EXCEPTION_STS 513
Extended Destination ID 367

**F**

FAILED 522
Fast Back to Back Capable 278, 304, 308, 320, 418, 436, 463, 482, 517, 536, 562
Fast Back to Back Enable 277, 303, 311, 319, 417, 435, 462, 516, 535, 561
Fast Primary Drive 0 Base Clock 429, 447
Fast Primary Drive 1 Base Clock 429, 447
Fast Secondary Drive 0 Base Clock 429, 447
Fast Secondary Drive 1 Base Clock 429, 447
FB_40_EN 341
FB_40_IDSEL 340
FB_50_EN 341
FB_50_IDSEL 340
FB_60_EN 341
FB_60_IDSEL 340
FB_70_EN 341
FB_70_IDSEL 340
FB_C0_EN 336
FB_C0_IDSEL 339
FB_C8_EN 336
FB_C8_IDSEL 339
FB_D0_EN 336
FB_D0_IDSEL 339
FB_D8_EN 336
FB_D8_IDSEL 339
FB_E0_EN 336

FB_E0_IDSEL 339
FB_E8_EN 336
FB_E8_IDSEL 339
FB_F0_EN 336
FB_F0_IDSEL 339
FB_F8_EN 336
FB_F8_IDSEL 339
FC Full 295
FC Paused 295
FC Paused Low 295
FCP Mask 289
FDD Decode Range 335
FDD_LPC_EN 338
FIFO Error 551, 572
FIFO Error Interrupt Enable 553, 574
Flow Control Pause 288
Flow Control Threshold 295
Force Global Resume 471
Force Port Resume 510
Force Thermal Throttling 390
FR Mask 289
Frame Length Timing Value 490
Frame List Current Index/Frame Number 475, 505
Frame List Rollover 503
Frame List Rollover Enable 504
Frame List Size 501
Frame Received 288
Full Reset 375

**G**

GAMEH_LPC_EN 337
GAMEL_LPC_EN 337
GBL_SMI_EN 397
General Self-Test Result 292
Generic Decode Range 1 Enable 337
Generic I/O Decode Range 1 Base Address 337
Generic I/O Decode Range 2 Base Address 340
Generic I/O Decode Range 2 Enable 340
Global Enable 388
Global Release 389
Global Reset 472
Global Status 387
GO_CNT 513
GP_BLINK(n) 411
GP_INV(n) 412
GP_IO_SEL2(32) 413
GP_IO_SEL2(34) 413
GP_LVL(32) 414
GP_LVL(34) 414
GP_LVL(41-40) 414
GP_LVL(49-48) 414
GP_LVL(n) 411
GPE0_STS 399
GPI Interrupt Enable 555, 575
GPI Status Change Interrupt 557, 577
GPI0 Route 381
GPI1 Route 381
GPI15 Route 381
GPI2 Route 381
GPIn_EN 394
GPIn_STS 392
GPIO Enable 325
GPIO(n)_SEL 410
GPIO_USE_SEL(23-21, 15-14, 11-9, 5-0) 410
GPIO_USE_SEL2(49-48, 41-40) 412
GPIO11_ALERT_DISABLE 408

**H**

HC BIOS Owned Semaphore 491
HC OS Owned Semaphore 491
HCHalted 474, 502
Header Type 280, 306, 322, 538, 563
Hide Device 0 313
Hide Device 1 313
Hide Device 2 313
Hide Device 3 313
Hide Device 4 313
Hide Device 5 313
Hide Device 8 313
Hide ISA Bridge 330
High Priority PCI Enable 314
Hole Enable 314
Host Controller Process Error 474
Host Controller Reset 472, 501
Host System Error 474, 502
Host System Error Enable 504
HOST_BUSY 523
HOST_NOTIFY_INTREN 530
HOST_NOTIFY_STS 529
HOST_NOTIFY_WKEN 530
Hour Format 371
HP Unsupported 312
HPET Address Enable 330
HPET Address Select 330
Hub Interface Rate 312
Hub Interface Rate Valid 312
Hub Interface Timeslice 312
Hub Interface Width 312
HUBNMI_STS 405
HUBSCI_STS 405

HUBSERR_STS 405
HUBSMI_STS 405

**I**

I/O Address Base Bits 307
I/O Address Base Upper 16 bits (31-16) 310
I/O Address Limit Bits 307
I/O Address Limit Upper 16 bits (31-16) 310
I/O Addressing Capability 307
I/O Space Enable 277, 303, 319, 417, 435, 462, 481, 516, 535, 543, 561
I/O Space Indicator 281
I2C 240, 242
I2C_EN 520
i64_EN 377
ICW/OCW Select 356
ICW4 Write Required 356
IDE Decode Enable 424, 442
IDEP0_ACT_STS 401
IDEP0_TRP_EN 402
IDEP1_ACT_STS 401
IDEP1_TRP_EN 402
IDES0_ACT_STS 401
IDES0_TRP_EN 402
IDES1_ACT_STS 401
IDES1_TRP_EN 402
IN_USE_CNT 513
Index 452
INIT_NOW 374
Integrated SATA RAID Configuration 380
INTEL_USB2_EN 396
INTEL_USB2_STS 398
Interesting Packet 296
Internal LAN Master Request Status 315
Internal PCI Master Request Status 315
Interrupt 432, 458
Interrupt Disable 277, 417, 435, 462, 481, 516, 535, 561
Interrupt Enable 293
Interrupt Input Pin Polarity 367
Interrupt Level Select 359
Interrupt Line 282, 423, 441, 466, 485, 519, 541, 566
Interrupt Line Routing 310
Interrupt Mask 289
Interrupt on Async Advance 502
Interrupt on Async Advance Doorbell 500
Interrupt on Async Advance Enable 504
Interrupt on Complete Enable 475
Interrupt on Completion Enable 553, 574
Interrupt PIN 519
Interrupt Pin 283, 423, 441, 467, 485, 566
Interrupt Request Flag 372
Interrupt Request Level 357
Interrupt Request Mask 359
Interrupt Rout 583
Interrupt Routing Enable 326, 328
Interrupt Status 278, 418, 436, 463, 482, 517, 536, 562
Interrupt Threshold Control 500
Interrupt Vector Base Address 357
INTR 523
INTRD_SEL 408
INTREN 524
Intruder Detect 406
INUSE_STS 522
IO Space Indicator 518
IOCHK# NMI Enable 373
IOCHK# NMI Source Status 373
IORDY Sample Point 424, 442
IRQ Number 364
IRQ Routing 326, 328
IRQ1_CAUSE 409
IRQ10 ECL 362
IRQ11 ECL 362
IRQ12 ECL 362
IRQ12_CAUSE 409
IRQ14 ECL 362
IRQ15 ECL 362
IRQ3 ECL 361
IRQ4 ECL 361
IRQ5 ECL 361
IRQ6 ECL 361
IRQ7 ECL 361
IRQ9 ECL 362
ISA Enable 311

**K**

KBC_ACT_STS 401
KBC_LPC_EN 337
KBC_TRP_EN 402
Keyboard IRQ1 Latch Enable 330
KILL 524

**L**

LAN Connect Address 293
LAN Connect Register Address 293
LAN Connect Software Reset 297
Last Codec Read Data Input 558
Last Valid Buffer Completion Interrupt 551, 572
Last Valid Buffer Interrupt Enable 553, 574

Last Valid Index (4-0) 550, 571
LAST_BYTE 523
Latch Count of Selected Counters 353
Latch Status of Selected Counters 353
LEG_ACT_STS 401
LEG_IO_TRP_EN 402
Legacy Rout 581
Legacy Rout Capable 581
LEGACY_USB_EN 397
LEGACY_USB_STS 399
LEGACY_USB2_EN 396
LEGACY_USB2_STS 398
Light Host Controller Reset 500
Line Status 478, 509
Link Pointer Low 507
Link Status Change Indication 296
Link Status Indication 297
LINK_ID_STS 513
Loop Back Test Mode 471
Low Speed Device Attached 477
Lower 128-byte Lock 334
LPC Bridge Disable 342
LPT Decode Range 335
LPT_LPC_EN 338

**M**

Magic Packet 296
Main Counter Tick Period 581
Management Data Interrupt 288
Map Value 451
Mask 367
Master Abort Mode 311
Master Abort Status 278, 320, 536, 562
Master Clear 350
Master Data Parity Error Detected 304, 308, 482
Master Latency Timer Count 280, 305, 307, 420, 438, 484
Master/Slave in Buffered Mode 358
Max Data 312
Max Packet 471
Maximum Latency 283
Maximum Redirection Entries 366
MC_LPC_EN 337
MCSMI_ENMicrocontroller SMI Enable 396
MD3 556, 576
Memory Address Base 309
Memory Address Limit 309
Memory Space Enable 277, 303, 319, 417, 435, 462, 481, 516, 535, 561
Memory Write and Invalidate Enable 277, 303, 535, 561
Memory-Space Indicator 281
Microcontroller SMI# Status 399
Microphone 2 In Interrupt 556, 576
Microphone In Interrupt 557, 577
Microprocessor Mode 358
Minimum Grant 283
Minimum SLP_S4# Assertion Width Violation Status 378
Mode Selection Status 354
Modem In Interrupt 557, 577
Modem Out Interrupt 557, 577
MON(n)_TRAP_BASE 383
MON4_FWD_EN 382
MON4_MASK 383
MON5_FWD_EN 382
MON5_MASK 383
MON6_FWD_EN 382
MON6_MASK 383
MON7_FWD_EN 382
MON7_MASK 383
Mouse IRQ12 Latch Enable 330
MSI Enable 450
Multichannel Capabilities 556, 576
Multi-Function Device 280, 306, 322, 465
Multiple Message Capable 450
Multiple Message Enable 450
Multi-Transaction Timer Count Value 315

**N**

N_PORTS 497
NEWCENTURY_STS 405
Next Capability 448, 543, 566
Next EHCI Capability Pointer 491
Next Item Pointer 284
Next Item Pointer 1 Value 486
Next Item Pointer 2 Capability 489
Next Pointer 449
NMI Enable 374
NMI_NOW 407
NMI2SMI_EN 407
NMI2SMI_STS 406
No Reboot 332
Number of Companion Controllers 497
Number of Ports per Companion Controller 497
Number of Timer Capability 581

**O**

OCW2 Select 359
OCW3 Select 360
Opcode 293

Overall Enable 581
Overcurrent Active 477, 510
Overcurrent Change 510
Overcurrent Indicator 477
OWNER_CNT 512

**P**

Parity Error Response 277, 303, 319, 417, 435, 462, 481, 516, 535, 561
Parity Error Response Enable 311
Pass Through State 469
PCI Interrupt Enable 468
PCI Master Request Status 315
PCI Parity Inversion State 316
PCI SERR# Enable 373
PCISML_SCLI 298
PCISML_SCLO 298
PCISML_SDAI 298
PCISML_SDAO 298
PCISML_SGNT 298
PCISML_SREQ 298
PCM 4/6 Enable 554
PCM In 1, Microphone In 1 Data In Line 558
PCM In 2 Interrupt 556, 576
PCM In 2, Microphone In 2 Data In Line 558
PCM In Interrupt 557, 577
PCM Out Interrupt 557, 577
PCM Out Mode 554
PEC_DATA 526
PEC_EN 523
Periodic Interrupt Capable 583
Periodic Interrupt Enable 371
Periodic Interrupt Flag 372
Periodic List Execution 216
Periodic Schedule Enable 500
Periodic Schedule Status 502
Periodic SMI# Rate Select 377
PERIODIC_EN 396
PERIODIC_STS 398
PIRQAE_ACT_STS 401
PIRQBF_ACT_STS 401
PIRQCG_ACT_STS 401
PIRQDH_ACT_STS 401
PM1_STS_REG 399
PME Clock 284, 448, 487, 544, 567
PME Enable 285, 449, 488, 567
PME Status 285, 296, 449, 488, 545, 567
PME Support 284, 448, 487, 544, 567
PME_B0_EN 394
PME_B0_STS 392
PME_EN 395
PME_STS 392
Pointer Field 291
Poll Mode Command 360
POP_MODE_CAP 419, 437
POP_MODE_SEL 419, 437
Port 0 BIST FIS Initiate 454
Port 0 Enabled 451
Port 0 Present 451
Port 1 BIST FIS Initiate 454
Port 1 Enabled 451
Port 1 Present 451
Port Change Detect 503
Port Change Interrupt Enable 504
Port Enable/Disable Change 478, 510
Port Enabled/Disabled 478, 511
PORT Function Selection 291
Port Owner 509
Port Power 509
Port Reset 477, 509
Port Test Control 509
Port Wake Implemented 491
Port Wake Up Capability Mask 491
PORT0EN 470
PORT1EN 470
Position In Current Buffer (15-0) 552
Positive Decode Enable 331
Postable Memory Write Enable 319, 417, 435, 462, 481, 516
Power Button Enable 388
Power Button Override Status 386
Power Button Status 387
Power Failure 379
Power Management Capability ID 486
Power Management Data 286
Power Management Event Enable 545
Power Sequencing 617
Power State 285, 449, 488, 545, 567
PRD Interrupt Status 458
Prefetch Flush Enable 314
Prefetchable 281, 484
Prefetchable Memory Address Base 309
Prefetchable Memory Address Limit 310
Prefetched Index Value (4-0) 552, 573
PRIM_SIG_MODE 429, 447
Primary Bus Number 306
Primary Discard Timer 311
Primary Drive 0 Base Clock 430, 448
Primary Drive 0 Cycle Time 428, 446
Primary Drive 0 Synchronous DMA Mode Enable

427, 445
Primary Drive 1 Base Clock 430, 448
Primary Drive 1 Cycle Time 428, 446
Primary Drive 1 IORDY Sample Point 426, 444
Primary Drive 1 Recovery Time 426, 444
Primary Drive 1 Synchronous DMA Mode Enable 427, 445
Primary Master Channel Cable Reporting 429
Primary Slave Channel Cable Reporting 429
Programmable Frame List Flag 498
Programming Interface 321, 464, 483, 537, 563
PRQ 366
PWRBTN_LVL 377
PWROK Failure 378

**R**

Rate Select 370
Read / Write Control 431, 457
Read Back Command 353
Read Completion Status 557, 577
Read Policies for Periodic DMA 217
Read/Write Select 352
Read/Write Selection Status 354
Ready 293
Real Time Clock Index Address 374
Receive DMA Byte Count 293
Receive Not Ready 288
Receive Unit Command 290
Receive Unit Status 289
Received Master Abort 304, 308, 418, 436, 463, 482, 517
Received System Error 308
Received Target Abort 278, 304, 308, 320, 482, 517
RECEIVED_PID_STS(23-16) 514
Reclamation 502
Recovery Time 424, 442
Redirection Entry Clear 365
Refresh Cycle Toggle 373
Register Read Command 360
Register Result 292
Remote IRR 367
REQ(5)#/GNT(5)# PC/PCI Protocol Select 330
Reset CPU 375
Reset Registers 553, 574
Resource Type Indicator 322, 325, 420, 421, 422, 438, 439, 440, 465, 484, 538, 539, 540, 564
Resume Detect 474, 477
Resume Interrupt Enable 475
Revision ID 279, 305, 321, 419, 437, 463, 483, 537, 562
Revision Identificaiton 581
RI_EN 395
RI_STS 393
RNR Mask 289
ROM Content Result 292
Rotate and EOI Codes 359
RTC Event Enable 388
RTC Status 386
RTC_PWR_STSRTC Power Status 379
Run/Pause Bus Master 553, 574
Run/Stop 472, 501
RW 525

**S**

S/PDIF Interrupt 556, 576
S/PDIF Slot Map 554
Safe Mode 332
Sample Capabilities 556, 576
SATA Setup Data A 452
SATA Setup Data B 453
SCB General Pointer 291
SCI Enable 389
SCI IRQ Select 323
Scratchpad Bit 365
SEC_SIG_MODE 429, 447
SECOND_TO_STS 406
Secondary Bus Number 306
Secondary Bus Reset 311
Secondary Codec ID 542
Secondary Discard Timer 311
Secondary Drive 0 Base Clock 430, 448
Secondary Drive 0 Cycle Time 428, 446
Secondary Drive 0 Synchronous DMA Mode Enable 427, 445
Secondary Drive 1 Base Clock 429, 447
Secondary Drive 1 Cycle Time 428, 446
Secondary Drive 1 IORDY Sample Point 425, 444
Secondary Drive 1 Recovery Time 426, 444
Secondary Drive 1 Synchronous DMA Mode Enable 427, 445
Secondary Master Channel Cable Reporting 429
Secondary Slave Channel Cable Reporting 429
SEND_NOW 407
SEND_PID_CNT(15-8) 514
Serial Bus Release Number 467
Serial IRQ Enable 327
Serial IRQ Frame Size 327
Serial IRQ Mode Select 327

SERIRQ_SMI_STS 398
SERR# Due to Delayed Transaction Timeout 329
SERR# Due to Received Target Abort 316, 329
SERR# Enable 277, 303, 311, 319, 417, 435, 462, 481, 516, 535, 561
SERR# Enable on Receiving Target Abort 316
SERR# NMI Source Status 373
SERR# on Delayed Transaction Timeout Enable 328
SERR# on Received Target Abort Enable 328
Short Packet Interrupt Enable 475
Signaled System Error 278, 304, 320, 418, 436, 482, 517, 536, 562
Signaled Target Abort 278, 304, 308, 320, 418, 436, 463, 482, 517, 536, 562
Single or Cascade 356
Slave Identification Code 358
SLAVE_ADDR 527
Sleep Enable 389
Sleep Type 389
SLP_S4# Assertion Stretch Enable 379
SLP_S4# Minimum Assertion Width 379
SLP_SMI_EN 396
SLP_SMI_STS 399
SMB_CMD 524
SMB_FOR_BIOS 343
SMB_SMI_EN 520
SMBALERT_DIS 530
SMBALERT_STS 522
SMBCLK_CTL 529
SMBCLK_CUR_STS 529
SMBDATA_CUR_STS 529
SMBus Host Enable 520
SMBus SMI Status 398
SMBus TCO Mode 527
SMBus Wake Status 393
SMI at End of Pass-Through Enable 469
SMI Caused by End of Pass-Through 468
SMI Caused by Port 60 Read 468
SMI Caused by Port 60 Write 468
SMI Caused by Port 64 Read 468
SMI Caused by Port 64 Write 468
SMI Caused by USB Interrupt 468
SMI on Async 493
SMI on Async Advance 492
SMI on Async Advance Enable 493
SMI on Async Enable 494
SMI on BAR 492
SMI on BAR Enable 492
SMI on CF 494
SMI on CF Enable 494
SMI on Frame List Rollover 492
SMI on Frame List Rollover Enable 493
SMI on HCHalted 494
SMI on HCHalted Enable 494
SMI on HCReset 494
SMI on HCReset Enable 494
SMI on Host System Error 492
SMI on Host System Error Enable 493
SMI on OS Ownership Change 492
SMI on OS Ownership Enable 493
SMI on PCI Command 492
SMI on PCI Command Enable 492
SMI on Periodic 494
SMI on Periodic Enable 494
SMI on PMCSR 493
SMI on PMSCR Enable 494
SMI on Port 60 Reads Enable 469
SMI on Port 60 Writes Enable 469
SMI on Port 64 Reads Enable 469
SMI on Port 64 Writes Enable 469
SMI on Port Change Detect 492
SMI on Port Change Enable 493
SMI on PortOwner 493
SMI on PortOwner Enable 494
SMI on USB Complete 492
SMI on USB Complete Enable 493
SMI on USB Error 492
SMI on USB Error Enable 493
SMI on USB IRQ Enable 469
SMI_LOCK 377
SMLink Slave SMI Status 406
SMLINK_CLK_CTL 528
SMLINK0_CUR_STS 528
SMLINK1_CUR_STS 528
SOF Timing Value 476
Software Debug 471
Software Generated Interrupt 289
Software Interrupt 288
Software SMI# Timer Enable 396
SOP_MODE_CAP 419, 437
SOP_MODE_SEL 419, 437
Speaker Data Enable 373
Special Cycle Enable 277, 303, 319, 417, 435, 462, 481, 516, 535, 561
Special Fully Nested Mode 358
Special Mask Mode 360
Speed 297
Square Wave Enable 371
START 523
Start Frame Pulse Width 327

Start/Stop Bus Master 431, 457
Steer Enable 558
STPCLK_DEL 380
Sub Class Code 279, 305, 321, 419, 437, 464, 483, 518, 537, 563
Subordinate Bus Number 306
Subsystem ID 282, 422, 440, 466, 485, 519, 541, 565
Subsystem Vendor ID 281, 422, 440, 466, 484, 518, 540, 565
Suspend 477, 510
SW_TCO_SMI 406
SWSMI_RATE_SEL 379
SWSMI_TMR_STS 399
System Bus Message Disable 331
System Reset 375
System Reset Status 378

**T**

TCO Data In Value 404
TCO Data Out Value 404
TCO Interrupt Enable 324
TCO Interrupt Select 324
TCO Request 296
TCO Timer Halt 407
TCO Timer Initial Value 404
TCO Timer Value 403
TCO_EN 396
TCO_INT_STS 406
TCO_LOCK 407
TCO_MESSAGE(n) 408
TCO_STS 398
TCOSCI_EN 395
TCOSCI_STS 393
Tertiary Codec ID 542
Thermal Interrupt Override Status 394
Thermal Interrupt Status 394
THRM#_POL 395
THRM_DTY 391
THRM_EN 395
Throttle Status 390
THTL_DTY 391
THTL_EN 391
TIMEOUT 405
Timeout/CRC Interrupt Enable 475
Timer 0 Interrupt Active 582
Timer 1 Interrupt Active 582
Timer 2 Interrupt Active 582
Timer Counter 2 Enable 373
Timer Counter 2 OUT Status 373
Timer Interrupt Rout Capability 583
Timer Interrupt Type 584
Timer n 32-bit Mode 583
Timer n Interrupt Enable 584
Timer n Size 583
Timer n Type 584
Timer n Value Set 583
Timer Overflow Interrupt Enable 388
Timer Overflow Statu 387
Timer Value 390
TOKEN_PID_CNT(7-0) 514
Top_Swap Lock-Down 331
Top-Block Swap Mode 333
Transient Disconnect Detect 380
Trigger Mode 367
Type 281, 484, 539, 540

**U**

UDF Supported 536
UHCI v/s EHCI 215
Unimplemented Asynchronous Park Mode Bits 500
Update Cycle Inhibit 371
Update In Progress 370
Update-Ended Flag 372
Update-Ended Interrupt Enable 371
Upper 128-byte Enable 334
Upper 128-byte Lock 334
Upper Address(43-32) 506
Upper Address(63-44) 506
USB Error Interrupt 474, 503
USB Error Interrupt Enable 504
USB Interrupt 474, 503
USB Interrupt Enable 504
USB Release Number 489
USB_ADDRESS_CNF 514
USB_ENDPOINT_CNF 514
USB1_EN 395
USB1_STS 393
USB2_EN 395
USB2_STS 393
USB3_EN 394
USB3_STS 392
USB4_EN 394
USB4_STS 392
User Definable Features 278, 320, 418, 436, 463, 482, 517, 562

**V**

Valid RAM and Time Bit 372
Vector 368
Vendor ID 276, 302, 318, 416, 434, 462, 480,

515, 534, 560
Vendor ID Capability 581
Version 284, 366, 448, 487, 544, 567
VGA 16-Bit Decode 311
VGA Enable 311
VGA Palette Snoop 277, 303, 319, 417, 435, 462, 481, 516, 535, 561

**W**

Wait Cycle Control 277, 303, 319, 417, 435, 462, 481, 516, 535, 561
Wake on Connect Enable 508
Wake on Disconnect Enable 508
Wake on Overcurrent Enable 508
Wake Status 386
Watchdog Status 408
Write Policies for Periodic DMA 218
WRITE_READ#_CNT 513
WRT_RDONLY 495

**X**

Xoff 295
Xon 295

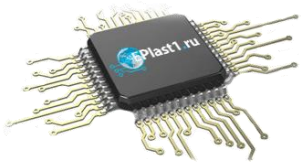
Компания «ЭлектроПласт» предлагает заключение долгосрочных отношений при поставках импортных электронных компонентов на взаимовыгодных условиях!

Наши преимущества:

- Оперативные поставки широкого спектра электронных компонентов отечественного и импортного производства напрямую от производителей и с крупнейших мировых складов;
- Поставка более 17-ти миллионов наименований электронных компонентов;
- Поставка сложных, дефицитных, либо снятых с производства позиций;
- Оперативные сроки поставки под заказ (от 5 рабочих дней);
- Экспресс доставка в любую точку России;
- Техническая поддержка проекта, помощь в подборе аналогов, поставка прототипов;
- Система менеджмента качества сертифицирована по Международному стандарту ISO 9001;
- Лицензия ФСБ на осуществление работ с использованием сведений, составляющих государственную тайну;
- Поставка специализированных компонентов (Xilinx, Altera, Analog Devices, Intersil, Interpoint, Microsemi, Aeroflex, Peregrine, Syfer, Eurofarad, Texas Instrument, Miteq, Cobham, E2V, MA-COM, Hittite, Mini-Circuits,General Dynamics и др.);

Помимо этого, одним из направлений компании «ЭлектроПласт» является направление «Источники питания». Мы предлагаем Вам помощь Конструкторского отдела:

- Подбор оптимального решения, техническое обоснование при выборе компонента;
- Подбор аналогов;
- Консультации по применению компонента;
- Поставка образцов и прототипов;
- Техническая поддержка проекта;
- Защита от снятия компонента с производства.




**Как с нами связаться**

**Телефон:** 8 (812) 309 58 32 (многоканальный)
**Факс:** 8 (812) 320-02-42
**Электронная почта:** org@eplast1.ru
**Адрес:** 198099, г. Санкт-Петербург, ул. Калинина, дом 2, корпус 4, литера А.