



# Intel® 82801BA I/O Controller Hub 2 (ICH2)

Datasheet

---

*June 2000*



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.

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

The Intel 82801BA I/O Controller Hub 2 (ICH2) 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.

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

Alert on LAN and Wake on LAN are results of the IBM/Intel Advanced Manageability Alliance and are trademarks of IBM Corporation.

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by:

calling 1-800-548-4725 or

by visiting Intel's website at <http://www.intel.com>.

Copyright © Intel Corporation, 2000

\*Third-party brands and names are the property of their respective owners.

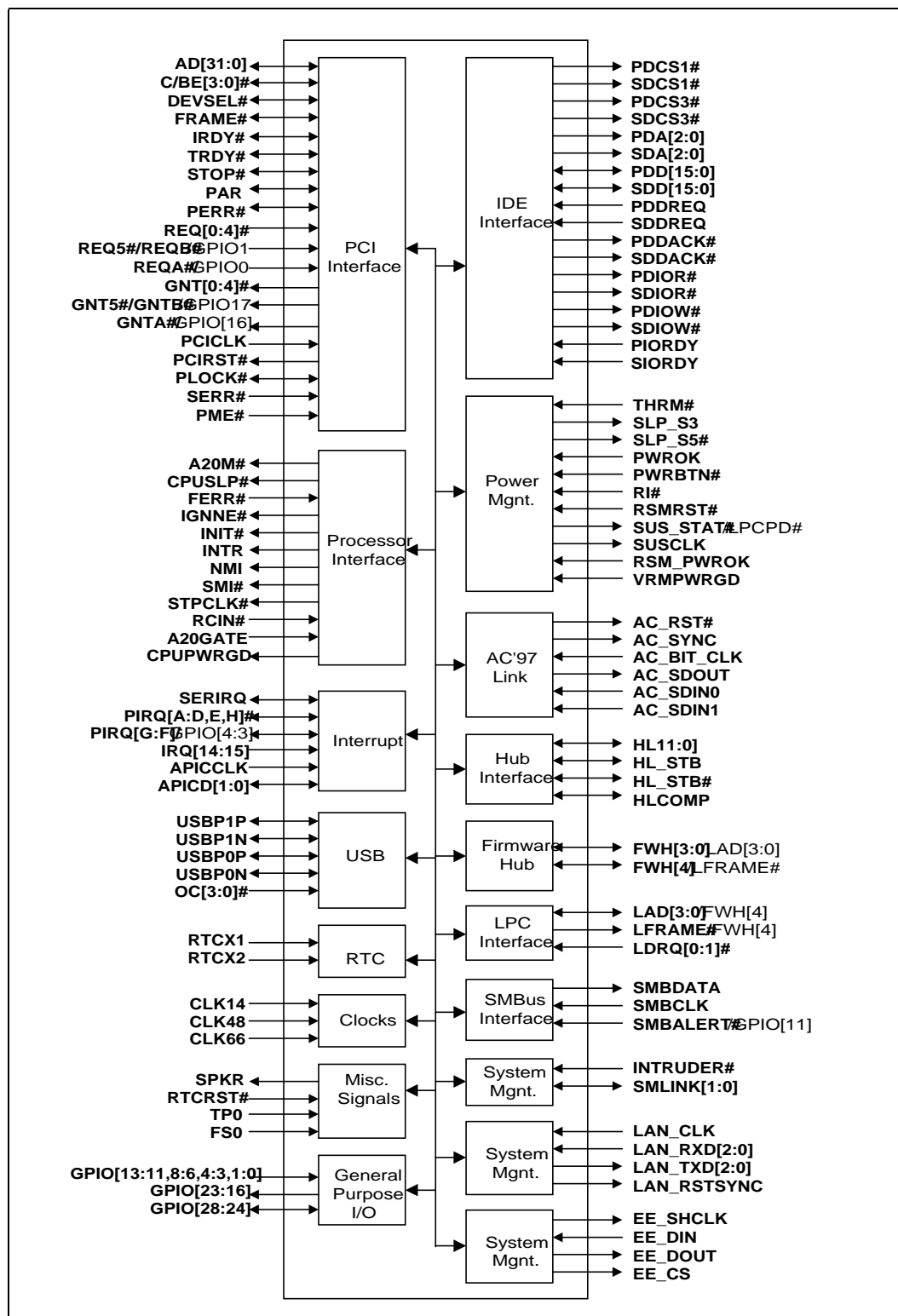


## Intel® 82801BA ICH2 Features

- n PCI Bus I/F
  - Supports PCI at 33 MHz
  - Supports PCI Rev 2.2 Specification
  - 133 MByte/sec maximum throughput
  - Supports up to 6 master devices on PCI
  - One PCI REQ/GNT pair can be given higher arbitration priority (intended for external 1394 host controller)
- n Integrated LAN Controller
  - WfM 2.0 Compliant
  - Interface to discrete LAN Connect component
  - 10/100 Mbit/sec Ethernet support
  - 1 Mbit/sec HomePNA\* support
- n Integrated IDE Controller
  - Independent timing of up to 4 drives
  - Ultra ATA/100/66/33, BMIDE and PIO modes
  - Read transfers up to 100MB/s, Writes to 89 MB/s
  - Separate IDE connections for Primary and Secondary cables
  - Implements Write Ping-Pong Buffer for faster write performance
- n USB
  - 2 UHCI Host Controllers with a total of 4 ports
  - USB 1.1 compliant
  - Supports wake-up from sleeping states S1–S4
  - Supports legacy Keyboard/Mouse software
- n AC'97 Link for Audio and Telephony CODECs
  - AC'97 2.1 compliant
  - Independent bus master logic for 5 channels (PCM In/Out, Mic Input, Modem In/Out)
  - Separate independent PCI functions for Audio and Modem
  - Support for up to six channels of PCM audio output (full AC3 decode)
  - Supports wake-up events
- n Interrupt Controller
  - Support up to 8 PCI interrupt pins
  - Supports PCI 2.2 Message-Based Interrupts
  - Two cascaded 82C59
  - Integrated I/O APIC capability
  - 15 interrupts supported in 8259 mode, 24 supported in I/O APIC mode
  - Supports Serial Interrupt Protocol
  - Supports Front-Side Bus interrupt delivery
- n 1.8 V operation with 3.3 V I/O
  - 5V tolerant buffers on IDE, PCI, USB Over-current and Legacy signals
- n 360-pin EBGA package
- n Power Management Logic
  - ACPI 1.0 compliant
  - ACPI-defined power states (C1–C2, S3–S5)
  - ACPI Power Management Timer
  - PCI PME# support
  - SMI# generation
  - All registers readable/restorable for proper resume from 0V suspend states
  - Support for APM-based legacy power management for non-ACPI Desktop implementations
- n External Glue Integration
  - Integrated Pull-up, Pull-down and Series Termination resistors on IDE and processor interface
- n Enhanced Hub I/F buffers improve routing flexibility (Not available with all Memory Controller Hubs)
- n Firmware Hub (FWH) I/F supports BIOS memory size up to 8 MBs
- n Low Pin count (LPC) I/F
  - Allows connection of legacy ISA and X-Bus devices such as Super I/O
  - Supports two Master/DMA devices.
- n 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
- n Real-Time Clock
  - 256-byte battery-backed CMOS RAM
  - Hardware implementation to indicate century rollover
- n 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
- n SM Bus
  - Host interface allows processor to communicate via SM Bus
  - Slave interface allows an external Microcontroller to access system resources
  - Compatible with most 2-Wire components that are also I<sup>2</sup>C compatible
- n Supports ISA bus via external PCI-ISA Bridge
- n GPIO
  - TTL, Open-Drain, Inversion
- n Timers Based on 82C54
  - System timer, Refresh request, Speaker tone output

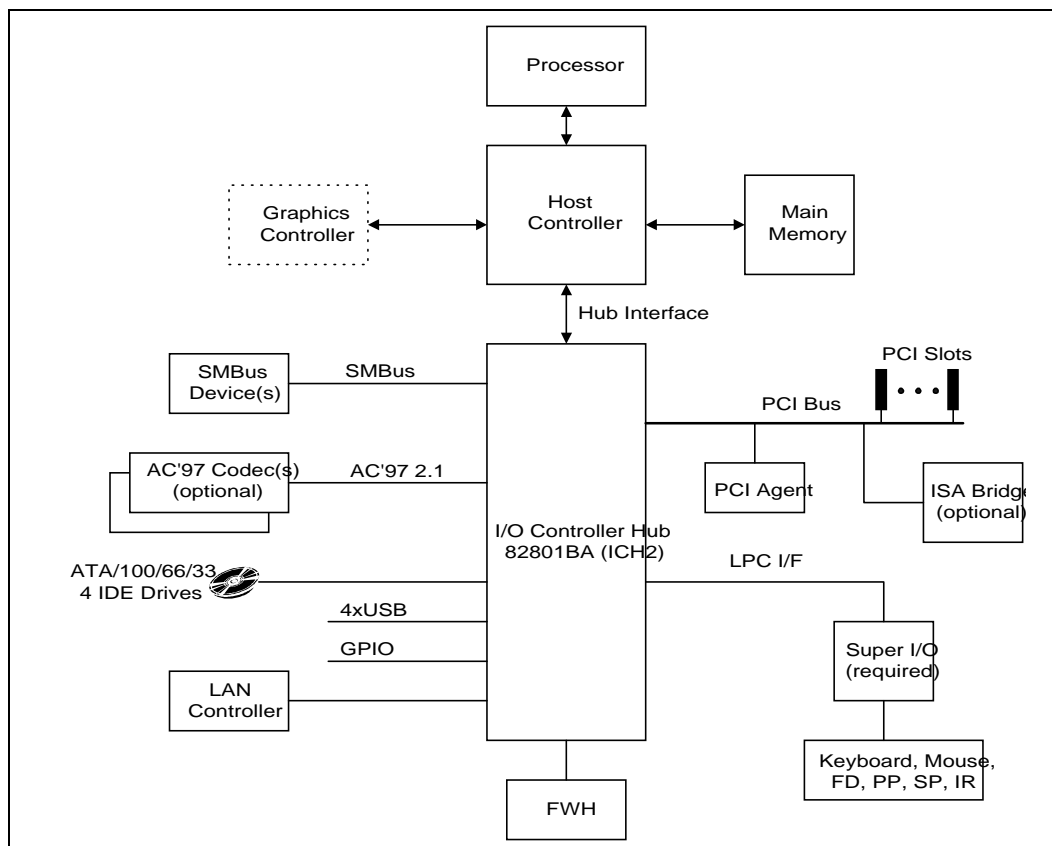
The Intel® 82801BA may contain design defects or errors known as errata which may cause the products to deviate from published specifications. Current characterized errata are available on request.

## Intel® 82801BA (ICH2) Simplified Block Diagram





## System Configuration



# Contents

1	Introduction .....	1-1
1.1	About this Document .....	1-1
1.2	Overview .....	1-3
2	Signal Description .....	2-1
2.1	Hub Interface to Host Controller .....	2-1
2.2	Link to LAN Connect .....	2-1
2.3	EEPROM Interface .....	2-2
2.4	Firmware Hub Interface .....	2-2
2.5	PCI Interface .....	2-2
2.6	IDE Interface .....	2-4
2.7	LPC Interface .....	2-6
2.8	Interrupt Interface .....	2-6
2.9	USB Interface .....	2-7
2.10	Power Management Interface .....	2-7
2.11	Processor Interface .....	2-8
2.12	SMBus Interface .....	2-9
2.13	System Management Interface .....	2-9
2.14	Real Time Clock Interface .....	2-10
2.15	Other Clocks .....	2-10
2.16	Miscellaneous Signals .....	2-10
2.17	AC'97 Link .....	2-11
2.18	General Purpose I/O .....	2-11
2.19	Power and Ground .....	2-12
2.20	Pin Straps .....	2-12
2.20.1	Functional Straps .....	2-12
2.20.2	Test Signals .....	2-13
2.20.2.1	Test Mode Selection .....	2-13
2.20.2.2	Test Straps .....	2-13
2.20.3	External RTC Circuitry .....	2-14
2.20.4	V5REF / Vcc3_3 Sequencing Requirements .....	2-14
3	ICH2 Power Planes and Pin States .....	3-1
3.1	Power Planes .....	3-1
3.2	Integrated Pull-Ups and Pull-Downs .....	3-1
3.3	IDE Integrated Series Termination Resistors .....	3-2
3.4	Output and I/O Signals Planes and States .....	3-2
3.5	Power Planes for Input Signals .....	3-5
4	ICH2 and System Clock Domains .....	4-1
5	Functional Description .....	5-1
5.1	Hub Interface to PCI Bridge (D30:F0) .....	5-1
5.1.1	PCI Bus Interface .....	5-1
5.1.2	PCI-to-PCI Bridge Model .....	5-2
5.1.3	IDSEL to Device Number Mapping .....	5-2
5.1.4	SERR# Functionality .....	5-2
5.1.5	Parity Error Detection .....	5-4



5.1.6	Standard PCI Bus Configuration Mechanism .....	5-4
5.1.7	PCI Dual Address Cycle (DAC) Support.....	5-5
5.2	LAN Controller (B1:D8:F0).....	5-5
5.2.1	LAN Controller Architectural Overview .....	5-6
5.2.2	LAN Controller PCI Bus Interface .....	5-8
5.2.2.1	Bus Slave Operation.....	5-8
5.2.2.2	Bus Master Operation.....	5-10
5.2.2.3	PCI Power Management .....	5-12
5.2.2.4	PCI Reset Signal .....	5-14
5.2.2.5	Wake-up Events .....	5-14
5.2.2.6	Wake on LAN (Preboot Wake-up) .....	5-15
5.2.3	Serial EEPROM Interface .....	5-16
5.2.4	CSMA/CD Unit .....	5-17
5.2.5	Media Management Interface .....	5-18
5.3	LPC Bridge (w/ System and Management Functions) (D31:F0).....	5-19
5.3.1	LPC Interface .....	5-19
5.3.1.1	LPC Cycle Types.....	5-20
5.3.1.2	Start Field Definition .....	5-20
5.3.1.3	Cycle Type / Direction (CYCTYPE + DIR).....	5-21
5.3.1.4	Size.....	5-21
5.3.1.5	SYNC.....	5-21
5.3.1.6	SYNC Time-out .....	5-22
5.3.1.7	SYNC Error Indication .....	5-22
5.3.1.8	LFRAME# Usage.....	5-22
5.3.1.9	I/O Cycles .....	5-23
5.3.1.10	Bus Master Cycles.....	5-24
5.3.1.11	LPC Power Management .....	5-24
5.3.1.12	Configuration and ICH2 Implications .....	5-24
5.4	DMA Operation (D31:F0) .....	5-25
5.4.1	Channel Priority .....	5-25
5.4.2	Address Compatibility Mode .....	5-26
5.4.3	Summary of DMA Transfer Sizes .....	5-26
5.4.4	Autoinitialize.....	5-27
5.4.5	Software Commands .....	5-28
5.5	PCI DMA.....	5-29
5.5.1	PCI DMA Expansion Protocol .....	5-29
5.5.2	PCI DMA Expansion Cycles .....	5-31
5.5.3	DMA Addresses .....	5-31
5.5.4	DMA Data Generation.....	5-31
5.5.5	DMA Byte Enable Generation.....	5-32
5.5.6	DMA Cycle Termination .....	5-32
5.5.7	LPC DMA .....	5-32
5.5.8	Asserting DMA Requests.....	5-32
5.5.9	Abandoning DMA Requests.....	5-33
5.5.10	General Flow of DMA Transfers .....	5-34
5.5.11	Terminal Count .....	5-34
5.5.12	Verify Mode.....	5-34
5.5.13	DMA Request Deassertion .....	5-35
5.5.14	SYNC Field / LDRQ# Rules .....	5-36
5.6	8254 Timers (D31:F0).....	5-37
5.6.1	Timer Programming .....	5-37
5.6.2	Reading from the Interval Timer .....	5-38



5.7	8259 Interrupt Controllers (PIC) (D31:F0) .....	5-40
5.7.1	Interrupt Handling .....	5-41
5.7.1.1	Generating Interrupts .....	5-41
5.7.1.2	Acknowledging Interrupts .....	5-41
5.7.1.3	Hardware/Software Interrupt Sequence .....	5-42
5.7.2	Initialization Command Words (ICWx) .....	5-42
5.7.3	Operation Command Words (OCW) .....	5-43
5.7.4	Modes of Operation .....	5-44
5.7.5	Masking Interrupts .....	5-46
5.7.6	Steering PCI Interrupts .....	5-46
5.8	Advanced Interrupt Controller (APIC) (D31:F0) .....	5-47
5.8.1	Interrupt Handling .....	5-47
5.8.2	Interrupt Mapping .....	5-48
5.8.3	APIC Bus Functional Description .....	5-49
5.8.3.1	Physical Characteristics of APIC .....	5-49
5.8.3.2	APIC Bus Arbitration .....	5-49
5.8.3.3	Bus Message Formats .....	5-50
5.8.4	PCI Message-Based Interrupts .....	5-55
5.8.4.1	Theory of Operation .....	5-55
5.8.4.2	Registers and Bits Associated with PCI Interrupt Delivery .....	5-55
5.8.5	Front-Side Interrupt Delivery .....	5-56
5.8.5.1	Theory of Operation .....	5-56
5.8.5.2	Edge-Triggered Operation .....	5-56
5.8.5.3	Level-Triggered Operation .....	5-56
5.8.5.4	Registers Associated with Front-Side Bus Interrupt Delivery .....	5-56
5.8.5.5	Interrupt Message Format .....	5-57
5.9	Serial Interrupt (D31:F0) .....	5-58
5.9.1	Start Frame .....	5-58
5.9.2	Data Frames .....	5-58
5.9.3	Stop Frame .....	5-59
5.9.4	Specific Interrupts not Supported via SERIRQ .....	5-59
5.9.5	Data Frame Format .....	5-60
5.10	Real Time Clock (D31:F0) .....	5-61
5.10.1	Update Cycles .....	5-61
5.10.2	Interrupts .....	5-62
5.10.3	Lockable RAM Ranges .....	5-62
5.10.4	Century Rollover .....	5-62
5.10.5	Clearing Battery-Backed RTC RAM .....	5-63
5.11	Processor Interface (D31:F0) .....	5-64
5.11.1	Processor Interface Signals .....	5-64
5.11.1.1	A20M# .....	5-64
5.11.1.2	INIT# .....	5-64
5.11.1.3	FERR#/IGNNE# (Coprocessor Error) .....	5-65
5.11.1.4	NMI .....	5-65
5.11.1.5	STPCLK# and CPUSLP# Signals .....	5-66
5.11.1.6	CPUPWRGOOD Signal .....	5-66
5.11.2	Dual Processor Issues .....	5-66
5.11.2.1	Signal Differences .....	5-66
5.11.2.2	Power Management .....	5-66
5.11.3	Speed Strapping for Processor .....	5-67
5.12	Power Management (D31:F0) .....	5-68



5.12.1	ICH2 and System Power States .....	5-68
5.12.2	System Power Planes.....	5-70
5.12.3	ICH2 Power Planes.....	5-70
5.12.4	SMI#/SCI Generation.....	5-70
5.12.5	Dynamic Processor Clock Control .....	5-72
5.12.5.1	Throttling Using STPCLK# .....	5-72
5.12.5.2	Transition Rules Among S0/Cx and Throttling States ....	5-73
5.12.6	Sleep States.....	5-73
5.12.6.1	Initiating Sleep State.....	5-74
5.12.6.2	Exiting Sleep States .....	5-74
5.12.6.3	Sx–G3–Sx, Handling Power Failures .....	5-76
5.12.7	Thermal Management.....	5-77
5.12.7.1	THRM# Signal .....	5-77
5.12.7.2	THRM# Initiated Passive Cooling.....	5-77
5.12.7.3	THRM# Override Software Bit.....	5-77
5.12.7.4	Processor-Initiated Passive Cooling (Via Programmed Duty Cycle on STPCLK#) .....	5-78
5.12.7.5	Active Cooling.....	5-78
5.12.8	Event Input Signals and Their Usage .....	5-78
5.12.8.1	PWRBTN# — Power Button.....	5-78
5.12.8.2	RI# — Ring Indicate .....	5-79
5.12.8.3	PME# — PCI Power Management Event.....	5-79
5.12.9	Alt Access Mode .....	5-80
5.12.9.1	Write Only Registers with Read Paths in Alternate Access Mode .....	5-80
5.12.9.2	PIC Reserved Bits .....	5-82
5.12.9.3	Read Only Registers with Write Paths in Alternate Access Mode .....	5-82
5.12.10	System Power Supplies, Planes, and Signals .....	5-83
5.12.11	Clock Generators .....	5-84
5.12.12	Legacy Power Management Theory of Operation .....	5-84
5.13	System Management (D31:F0).....	5-85
5.13.1	Theory of Operation .....	5-85
5.13.2	Alert on LAN* .....	5-86
5.14	General Purpose I/O.....	5-88
5.15	IDE Controller (D31:F1) .....	5-89
5.15.1	PIO Transfers.....	5-89
5.15.2	Bus Master Function .....	5-91
5.15.3	Ultra ATA/33 Protocol .....	5-95
5.15.4	Ultra ATA/66 Protocol .....	5-97
5.15.5	Ultra ATA/100 Protocol .....	5-97
5.15.6	Ultra ATA/33/66/100 Timing.....	5-97
5.16	USB Controller (Device 31:Functions 2 and 4) .....	5-98
5.16.1	Data Structures in Main memory .....	5-98
5.16.1.1	Frame List Pointer .....	5-98
5.16.1.2	Transfer Descriptor (TD).....	5-99
5.16.1.3	Queue Head (QH) .....	5-103
5.16.2	Data Transfers To/From Main Memory.....	5-104
5.16.2.1	Executing the Schedule.....	5-104
5.16.2.2	Processing Transfer Descriptors .....	5-104
5.16.2.3	Command Register, Status Register, and TD Status Bit Interaction .....	5-105
5.16.2.4	Transfer Queuing.....	5-106



5.16.3	Data Encoding and Bit Stuffing .....	5-109
5.16.4	Bus Protocol .....	5-110
5.16.4.1	Bit Ordering .....	5-110
5.16.4.2	SYNC Field.....	5-110
5.16.4.3	Packet Field Formats .....	5-110
5.16.4.4	Address Fields.....	5-111
5.16.4.5	Frame Number Field .....	5-112
5.16.4.6	Data Field .....	5-112
5.16.4.7	Cyclic Redundancy Check (CRC) .....	5-112
5.16.5	Packet Formats.....	5-113
5.16.5.1	Token Packets.....	5-113
5.16.5.2	Start of Frame Packets.....	5-113
5.16.5.3	Data Packets .....	5-114
5.16.5.4	Handshake Packets .....	5-114
5.16.5.5	Handshake Responses .....	5-115
5.16.6	USB Interrupts .....	5-115
5.16.6.1	Transaction Based Interrupts .....	5-115
5.16.6.2	Non-Transaction Based Interrupts .....	5-117
5.16.7	USB Power Management .....	5-117
5.16.8	USB Legacy Keyboard Operation.....	5-118
5.17	SMBus Controller Functional Description (D31:F3) .....	5-120
5.17.1	Host Controller .....	5-120
5.17.1.1	Command Protocols .....	5-121
5.17.1.2	I <sup>2</sup> C Behavior.....	5-126
5.17.1.3	Heartbeat for Use With the External LAN Controller ....	5-126
5.17.2	Bus Arbitration .....	5-127
5.17.3	Interrupts / SMI# .....	5-127
5.17.4	SMBALERT# .....	5-128
5.17.5	SMBus Slave Interface .....	5-128
5.18	AC'97 Controller Functional Description (Audio D31:F5, Modem D31:F6).....	5-132
5.18.1	AC-link Overview .....	5-133
5.18.2	AC-Link Low Power Mode .....	5-141
5.18.3	AC'97 Cold Reset .....	5-142
5.18.4	AC'97 Warm Reset .....	5-142
5.18.5	System Reset .....	5-143
5.19	Firmware Hub Interface .....	5-144
5.19.1	Field Definitions .....	5-144
5.19.2	Protocol.....	5-145
6	Register and Memory Mapping.....	6-1
6.1	PCI Devices and Functions .....	6-1
6.2	PCI Configuration Map .....	6-2
6.3	I/O Map .....	6-2
6.3.1	Fixed I/O Address Ranges.....	6-3
6.3.2	Variable I/O Decode Ranges .....	6-5
6.4	Memory Map.....	6-6
6.4.1	Boot-Block Update Scheme.....	6-7



LAN Controller Registers (B1:D8:F0).....	7-1
7.1 PCI Configuration Registers (B1:D8:F0).....	7-1
7.1.1 VID—Vendor ID Register (LAN Controller—B1:D8:F0) .....	7-2
7.1.2 DID—Device ID Register (LAN Controller—B1:D8:F0) .....	7-2
7.1.3 PCICMD—PCI Command Register (LAN Controller—B1:D8:F0) .....	7-2
7.1.4 PCISTS—PCI Status Register (LAN Controller—B1:D8:F0) .....	7-3
7.1.5 REVID—Revision ID Register (LAN Controller—B1:D8:F0) .....	7-3
7.1.6 SCC—Sub-Class Code Register (LAN Controller—B1:D8:F0) .....	7-4
7.1.7 BCC—Base-Class Code Register (LAN Controller—B1:D8:F0) .....	7-4
7.1.8 CLS—Cache Line Size Register (LAN Controller—B1:D8:F0) .....	7-4
7.1.9 PMLT—PCI Master Latency Timer Register (LAN Controller—B1:D8:F0) .....	7-4
7.1.10 HEADTYP—Header Type Register (LAN Controller—B1:D8:F0) .....	7-5
7.1.11 CSR_MEM_BASE CSR—Memory-Mapped Base Address Register (LAN Controller—B1:D8:F0) .....	7-5
7.1.12 CSR_IO_BASE—CSR I/O-Mapped Base Address Register (LAN Controller—B1:D8:F0) .....	7-5
7.1.13 SVID—Subsystem Vendor ID(LAN Controller—B1:D8:F0) .....	7-6
7.1.14 SID—Subsystem ID(LAN Controller—B1:D8:F0) .....	7-6
7.1.15 CAP_PTR—Capabilities Pointer (LAN Controller—B1:D8:F0) .....	7-6
7.1.16 INT_LN—Interrupt Line Register (LAN Controller—B1:D8:F0) .....	7-7
7.1.17 INT_PN—Interrupt Pin Register (LAN Controller—B1:D8:F0) .....	7-7
7.1.18 MIN_GNT—Minimum Grant Register (LAN Controller—B1:D8:F0) .....	7-7
7.1.19 MAX_LAT—Maximum Latency Register (LAN Controller—B1:D8:F0) .....	7-7
7.1.20 CAP_ID—Capability ID Register (LAN Controller—B1:D8:F0) .....	7-8
7.1.21 NXT_PTR—Next Item Pointer (LAN Controller—B1:D8:F0) .....	7-8
7.1.22 PM_CAP—Power Management Capabilities (LAN Controller—B1:D8:F0) .....	7-8
7.1.23 PMCSR—Power Management Control/Status Register (LAN Controller—B1:D8:F0) .....	7-9
7.1.24 DATA—Data Register (LAN Controller—B1:D8:F0) .....	7-9
7.2 LAN Control / Status Registers (CSR) .....	7-10
7.2.1 System Control Block Status Word Register .....	7-11
7.2.2 System Control Block Command Word Register .....	7-12
7.2.3 System Control Block General Pointer Register .....	7-14
7.2.4 PORT Register.....	7-14
7.2.5 EEPROM Control Register .....	7-15
7.2.6 Management Data Interface (MDI) Control Register .....	7-16
7.2.7 Receive DMA Byte Count Register .....	7-16
7.2.8 Early Receive Interrupt Register .....	7-17
7.2.9 Flow Control Register .....	7-18
7.2.10 Power Management Driver (PMDR) Register .....	7-19
7.2.11 General Control Register .....	7-19
7.2.12 General Status Register.....	7-20
7.2.13 Statistical Counters .....	7-20

8	Hub Interface to PCI Bridge Registers (D30:F0) .....	8-1
8.1	PCI Configuration Registers (D30:F0) .....	8-1
8.1.1	VID—Vendor ID Register (HUB-PCI—D30:F0) .....	8-2
8.1.2	DID—Device ID Register (HUB-PCI—D30:F0).....	8-2
8.1.3	CMD—Command Register (HUB-PCI—D30:F0).....	8-3
8.1.4	PD_STS—Primary Device Status Register (HUB-PCI—D30:F0) .....	8-4
8.1.5	REVID—Revision ID Register (HUB-PCI—D30:F0).....	8-4
8.1.6	SCC—Sub-Class Code Register (HUB-PCI—D30:F0).....	8-5
8.1.7	BCC—Base-Class Code Register (HUB-PCI—D30:F0).....	8-5
8.1.8	PMLT—Primary Master Latency Timer Register (HUB-PCI—D30:F0) .....	8-5
8.1.9	HEADTYP—Header Type Register (HUB-PCI—D30:F0).....	8-5
8.1.10	PBUS_NUM—Primary Bus Number Register (HUB-PCI—D30:F0) .....	8-6
8.1.11	SBUS_NUM—Secondary Bus Number Register (HUB-PCI—D30:F0) .....	8-6
8.1.12	SUB_BUS_NUM—Subordinate Bus Number Register (HUB-PCI—D30:F0) .....	8-6
8.1.13	SMLT—Secondary Master Latency Timer Register (HUB-PCI—D30:F0) .....	8-6
8.1.14	IOBASE—I/O Base Register (HUB-PCI—D30:F0).....	8-7
8.1.15	IOLIM—I/O Limit Register (HUB-PCI—D30:F0) .....	8-7
8.1.16	SECSTS—Secondary Status Register (HUB-PCI—D30:F0).....	8-8
8.1.17	MEMBASE—Memory Base Register (HUB-PCI—D30:F0) .....	8-9
8.1.18	MEMLIM—Memory Limit Register (HUB-PCI—D30:F0) .....	8-9
8.1.19	PREF_MEM_BASE—Prefetchable Memory Base Register (HUB-PCI—D30:F0) .....	8-9
8.1.20	PREF_MEM_MLT—Prefetchable Memory Limit Register (HUB-PCI—D30:F0) .....	8-10
8.1.21	IOBASE_HI—I/O Base Upper 16 Bits Register (HUB-PCI—D30:F0) .....	8-10
8.1.22	IOLIM_HI—I/O Limit Upper 16 Bits Register (HUB-PCI—D30:F0) .....	8-10
8.1.23	INT_LINE—Interrupt Line Register (HUB-PCI—D30:F0) .....	8-10
8.1.24	BRIDGE_CNT—Bridge Control Register (HUB-PCI—D30:F0) .....	8-11
8.1.25	BRIDGE_CNT2—Bridge Control Register 2 (HUB-PCI—D30:F0) .....	8-11
8.1.26	CNF—ICH2 Configuration Register (HUB-PCI—D30:F0) .....	8-12
8.1.27	MTT—Multi-Transaction Timer Register (HUB-PCI—D30:F0) .....	8-12
8.1.28	PCI_MAST_STS—PCI Master Status Register (HUB-PCI—D30:F0) .....	8-13
8.1.29	ERR_CMD—Error Command Register (HUB-PCI—D30:F0) .....	8-13
8.1.30	ERR_STS—Error Status Register (HUB-PCI—D30:F0).....	8-14



LPC Interface Bridge Registers (D31:F0) .....	9-1
9.1 PCI Configuration Registers (D31:F0) .....	9-1
9.1.1 VID—Vendor ID Register (LPC I/F—D31:F0) .....	9-2
9.1.2 DID—Device ID Register (LPC I/F—D31:F0) .....	9-2
9.1.3 PCICMD—PCI COMMAND Register (LPC I/F—D31:F0) .....	9-3
9.1.4 PCISTS—PCI Device Status (LPC I/F—D31:F0) .....	9-4
9.1.5 REVID—Revision ID Register (LPC I/F—D31:F0) .....	9-4
9.1.6 PI—Programming Interface (LPC I/F—D31:F0) .....	9-5
9.1.7 SCC—Sub-Class Code Register (LPC I/F—D31:F0) .....	9-5
9.1.8 BCC—Base-Class Code Register (LPC I/F—D31:F0) .....	9-5
9.1.9 HEADTYP—Header Type Register (LPC I/F—D31:F0) .....	9-5
9.1.10 PMBASE—ACPI Base Address (LPC I/F—D31:F0) .....	9-6
9.1.11 ACPI_CNTL—ACPI Control (LPC I/F—D31:F0) .....	9-6
9.1.12 BIOS_CNTL (LPC I/F—D31:F0) .....	9-7
9.1.13 TCO_CNTL—TCO Control (LPC I/F—D31:F0) .....	9-7
9.1.14 GPIOBASE—GPIO Base Address (LPC I/F—D31:F0) .....	9-8
9.1.15 GPIO_CNTL—GPIO Control (LPC I/F—D31:F0) .....	9-8
9.1.16 PIRQ[n]_ROUT—PIRQ[A,B,C,D] Routing Control (LPC I/F—D31:F0) .....	9-8
9.1.17 SERIRQ_CNTL—Serial IRQ Control (LPC I/F—D31:F0) .....	9-9
9.1.18 PIRQ[n]_ROUT—PIRQ[E,F,G,H] Routing Control (LPC I/F—D31:F0) .....	9-9
9.1.19 D31_ERR_CFG—Device 31 Error Configuration Register (LPC I/F—D31:F0) .....	9-10
9.1.20 D31_ERR_STS—Device 31 Error Status Register (LPC I/F—D31:F0) .....	9-10
9.1.21 PCI_DMA_CFG—PCI DMA Configuration (LPC I/F—D31:F0) .....	9-11
9.1.22 GEN_CNTL—General Control Register (LPC I/F—D31:F0) .....	9-11
9.1.23 GEN_STS—General Status (LPC I/F—D31:F0) .....	9-13
9.1.24 RTC_CONF—RTC Configuration Register (LPC I/F—D31:F0) .....	9-14
9.1.25 COM_DEC—LPC I/F Communication Port Decode Ranges (LPC I/F—D31:F0) .....	9-14
9.1.26 FDD/LPT_DEC—LPC I/F FDD & LPT Decode Ranges (LPC I/F—D31:F0) .....	9-15
9.1.27 SND_DEC—LPC I/F Sound Decode Ranges (LPC I/F—D31:F0) .....	9-15
9.1.28 FWH_DEC_EN1—FWH Decode Enable 1 Register (LPC I/F—D31:F0) .....	9-16
9.1.29 GEN1_DEC—LPC I/F Generic Decode Range 1 (LPC I/F—D31:F0) .....	9-17
9.1.30 LPC_EN—LPC I/F Enables (LPC I/F—D31:F0) .....	9-17
9.1.31 FWH_SEL1—FWH Select 1 Register (LPC I/F—D31:F0) .....	9-19
9.1.32 GEN2_DEC—LPC I/F Generic Decode Range 2 (LPC I/F—D31:F0) .....	9-20
9.1.33 FWH_SEL2—FWH Select 2 Register (LPC I/F—D31:F0) .....	9-20
9.1.34 FWH_DEC_EN2—FWH Decode Enable 2 Register (LPC I/F—D31:F0) .....	9-21
9.1.35 FUNC_DIS—Function Disable Register (LPC I/F—D31:F0) .....	9-22



9.2	DMA I/O Registers .....	9-23
9.2.1	DMABASE_CA—DMA Base and Current Address Registers .....	9-24
9.2.2	DMABASE_CC—DMA Base and Current Count Registers.....	9-25
9.2.3	DMAMEM_LP—DMA Memory Low Page Registers .....	9-25
9.2.4	DMACMD—DMA Command Register .....	9-26
9.2.5	DMASTS—DMA Status Register.....	9-26
9.2.6	DMA_WRMSK—DMA Write Single Mask Register.....	9-27
9.2.7	DMACH_MODE—DMA Channel Mode Register.....	9-27
9.2.8	DMA Clear Byte Pointer Register .....	9-28
9.2.9	DMA Master Clear Register.....	9-28
9.2.10	DMA_CLMSK—DMA Clear Mask Register .....	9-28
9.2.11	DMA_WRMSK—DMA Write All Mask Register .....	9-29
9.3	Timer I/O Registers.....	9-30
9.3.1	TCW—Timer Control Word Register .....	9-30
9.3.1.1	RDBK_CMD—Read Back Command .....	9-31
9.3.1.2	LTCH_CMD—Counter Latch Command .....	9-31
9.3.2	SBYTE_FMT—Interval Timer Status Byte Format Register .....	9-32
9.3.3	Counter Access Ports Register.....	9-32
9.4	8259 Interrupt Controller (PIC) Registers .....	9-33
9.4.1	Interrupt Controller I/O MAP .....	9-33
9.4.2	ICW1—Initialization Command Word 1 Register .....	9-34
9.4.3	ICW2—Initialization Command Word 2 Register .....	9-35
9.4.4	ICW3—Master Controller Initialization Command Word 3 Register .....	9-35
9.4.5	ICW3—Slave Controller Initialization Command Word 3 Register .....	9-36
9.4.6	ICW4—Initialization Command Word 4 Register .....	9-36
9.4.7	OCW1—Operational Control Word 1 (Interrupt Mask) Register ....	9-36
9.4.8	OCW2—Operational Control Word 2 Register .....	9-37
9.4.9	OCW3—Operational Control Word 3 Register .....	9-38
9.4.10	ELCR1—Master Controller Edge/Level Triggered Register .....	9-39
9.4.11	ELCR2—Slave Controller Edge/Level Triggered Register .....	9-40
9.5	Advanced Interrupt Controller (APIC) .....	9-41
9.5.1	APIC Register Map .....	9-41
9.5.2	IND—Index Register .....	9-41
9.5.3	DAT—Data Register .....	9-42
9.5.4	IRQPA—IRQ Pin Assertion Register .....	9-42
9.5.5	EOIR—EOI Register .....	9-43
9.5.6	ID—Identification Register .....	9-43
9.5.7	VER—Version Register .....	9-44
9.5.8	ARBID—Arbitration ID Register .....	9-44
9.5.9	BOOT_CONFIG—Boot Configuration Register .....	9-44
9.5.10	Redirection Table.....	9-45
9.6	Real Time Clock Registers .....	9-47
9.6.1	I/O Register Address Map .....	9-47
9.6.2	Indexed Registers.....	9-47
9.6.2.1	RTC_REGA—Register A .....	9-48
9.6.2.2	RTC_REGB—Register B (General Configuration) .....	9-49
9.6.2.3	RTC_REGC—Register C (Flag Register) .....	9-50
9.6.2.4	RTC_REGD—Register D (Flag Register) .....	9-50



9.7	Processor Interface Registers.....	9-51
9.7.1	NMI_SC—NMI Status and Control Register .....	9-51
9.7.2	NMI_EN—NMI Enable (and Real Time Clock Index) .....	9-52
9.7.3	PORT92—Fast A20 and Init Register.....	9-52
9.7.4	COPROC_ERR—Coprocessor Error Register .....	9-52
9.7.5	RST_CNT—Reset Control Register .....	9-53
9.8	Power Management Registers (D31:F0) .....	9-53
9.8.1	Power Management PCI Configuration Registers (D31:F0) .....	9-53
9.8.1.1	GEN_PMCON_1—General PM Configuration 1 Register (PM—D31:F0) .....	9-54
9.8.1.2	GEN_PMCON_2—General PM Configuration 2 Register (PM—D31:F0) .....	9-55
9.8.1.3	GEN_PMCON_3—General PM Configuration 3 Register (PM—D31:F0) .....	9-55
9.8.1.4	GPI_ROUT—GPI Routing Control Register (PM—D31:F0) .....	9-56
9.8.1.5	TRP_FWD_EN—IO Monitor Trap Forwarding Enable Register (PM—D31:F0).....	9-56
9.8.1.6	MON[n]_TRP_RNG—I/O Monitor [4:7] Trap Range Register for Devices 4–7 (PM—D31:F0) .....	9-57
9.8.1.7	MON_TRP_MSK—I/O Monitor Trap Range Mask Register for Devices 4–7 (PM—D31:F0) .....	9-57
9.8.2	APM I/O Decode .....	9-58
9.8.2.1	APM_CNT—Advanced Power Management Control Port Register.....	9-58
9.8.2.2	APM_STS—Advanced Power Management Status Port Register.....	9-58
9.8.3	Power Management I/O Registers.....	9-59
9.8.3.1	PM1_STS—Power Management 1 Status Register .....	9-60
9.8.3.2	PM1_EN—Power Management 1 Enable Register.....	9-61
9.8.3.3	PM1_CNT—Power Management 1 Control Register .....	9-62
9.8.3.4	PM1_TMR—Power Management 1 Timer Register .....	9-62
9.8.3.5	PROC_CNT—Processor Control Register .....	9-63
9.8.3.6	LV2—Level 2 Register.....	9-64
9.8.3.7	GPE0_STS—General Purpose Event 0 Status Register .....	9-64
9.8.3.8	GPE0_EN—General Purpose Event 0 Enables Register .....	9-66
9.8.3.9	GPE1_STS—General Purpose Event 1 Status Register .....	9-67
9.8.3.10	GPE1_EN—General Purpose Event 1 Enable Register .....	9-68
9.8.3.11	SMI_EN—SMI Control and Enable Register .....	9-68
9.8.3.12	SMI_STS—SMI Status Register.....	9-70
9.8.3.13	MON_SMI—Device Monitor SMI Status and Enable Register .....	9-71
9.8.3.14	DEVACT_STS—Device Activity Status Register .....	9-72
9.8.3.15	DEVTRAP_EN—Device Trap Enable Register .....	9-73
9.8.3.16	BUS_ADDR_TRACK—Bus Address Tracker Register .....	9-74
9.8.3.17	BUS_CYC_TRACK—Bus Cycle Tracker Register .....	9-74

9.9	System Management TCO Registers (D31:F0) .....	9-75
9.9.1	TCO Register I/O Map .....	9-75
9.9.2	TCO1_RLD—TCO Timer Reload and Current Value Register .....	9-75
9.9.3	TCO1_TMR—TCO Timer Initial Value Register .....	9-76
9.9.4	TCO1_DAT_IN—TCO Data In Register .....	9-76
9.9.5	TCO1_DAT_OUT—TCO Data Out Register .....	9-76
9.9.6	TCO1_STS—TCO1 Status Register .....	9-76
9.9.7	TCO2_STS—TCO2 Status Register .....	9-78
9.9.8	TCO1_CNT—TCO1 Control Register .....	9-79
9.9.9	TCO2_CNT—TCO2 Control Register .....	9-80
9.9.10	TCO_MESSAGE1 and TCO_MESSAGE2 Registers .....	9-80
9.9.11	TCO_WDSTATUS—TCO2 Control Register .....	9-80
9.9.12	SW_IRQ_GEN—Software IRQ Generation Register .....	9-81
9.10	General Purpose I/O Registers (D31:F0) .....	9-81
9.10.1	GPIO Register I/O Address Map .....	9-83
9.10.2	GPIO_USE_SEL—GPIO Use Select Register .....	9-83
9.10.3	GP_IO_SEL—GPIO Input/Output Select Register .....	9-84
9.10.4	GP_LVL—GPIO Level for Input or Output Register .....	9-84
9.10.5	GPO_BLINK—GPO Blink Enable Register .....	9-85
9.10.6	GPI_INV—GPIO Signal Invert Register .....	9-86
10	IDE Controller Registers (D31:F1) .....	10-1
10.1	PCI Configuration Registers (IDE—D31:F1) .....	10-1
10.1.1	CMD—Command Register (IDE—D31:F1) .....	10-2
10.1.2	STS—Device Status Register (IDE—D31:F1) .....	10-3
10.1.3	PI—Programming Interface (IDE—D31:F1) .....	10-3
10.1.4	SCC—Sub Class Code (IDE—D31:F1) .....	10-3
10.1.5	BCC—Base Class Code (IDE—D31:F1) .....	10-4
10.1.6	MLT—Master Latency Timer (IDE—D31:F1) .....	10-4
10.1.7	BM_BASE—Bus Master Base Address Register (IDE—D31:F1) .....	10-4
10.1.8	IDE_SVID—Subsystem Vendor ID (IDE—D31:F1) .....	10-5
10.1.9	IDE_SID—Subsystem ID (IDE—D31:F1) .....	10-5
10.1.10	IDE_TIM—IDE Timing Register (IDE—D31:F1) .....	10-6
10.1.11	SLV_IDETIM—Slave (Drive 1) IDE Timing Register (IDE—D31:F1) .....	10-7
10.1.12	SDMA_CNT—Synchronous DMA Control Register (IDE—D31:F1) .....	10-8
10.1.13	SDMA_TIM—Synchronous DMA Timing Register (IDE—D31:F1) .....	10-8
10.1.14	IDE_CONFIG—IDE I/O Configuration Register .....	10-9
10.2	Bus Master IDE I/O Registers (D31:F1) .....	10-11
10.2.1	BMIC[P,S]—Bus Master IDE Command Register .....	10-11
10.2.2	BMIS[P,S]—Bus Master IDE Status Register .....	10-12
10.2.3	BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register .....	10-12



11	USB Controller Registers.....	11-1
11.1	PCI Configuration Registers (D31:F2/F4) .....	11-1
11.1.1	VID—Vendor Identification Register (USB—D31:F2/F4) .....	11-1
11.1.2	DID—Device Identification Register (USB—D31:F2/F4) .....	11-2
11.1.3	CMD—Command Register (USB—D31:F2/F4).....	11-2
11.1.4	STA—Device Status Register (USB—D31:F2/F4) .....	11-3
11.1.5	RID—Revision Identification Register (USB—D31:F2/F4) .....	11-3
11.1.6	PI—Programming Interface (USB—D31:F2/F4) .....	11-3
11.1.7	SCC—Sub Class Code Register (USB—D31:F2/F4) .....	11-4
11.1.8	BCC—Base Class Code Register (USB—D31:F2/F4) .....	11-4
11.1.9	BASE—Base Address Register (USB—D31:F2/F4).....	11-4
11.1.10	SVID—Subsystem Vendor ID (USB—D31:F2/F4).....	11-4
11.1.11	SID—Subsystem ID (USB—D31:F2/F4).....	11-5
11.1.12	INTR_LN—Interrupt Line Register (USB—D31:F2/F4) .....	11-5
11.1.13	INTR_PN—Interrupt Pin Register (USB—D31:F2/F4).....	11-5
11.1.14	SB_RELNUM—Serial Bus Release Number Register (USB—D31:F2/F4).....	11-5
11.1.15	uSB_LEGKEY—USB Legacy Keyboard/Mouse Control Register (USB—D31:F2/F4) .....	11-6
11.1.16	USB_RES—USB Resume Enable Register (USB—D31:F2/F4).....	11-7
11.2	USB I/O Registers.....	11-8
11.2.1	USBCMD—USB Command Register .....	11-8
11.2.2	USBSTA—USB Status Register .....	11-11
11.2.3	USBINTR—Interrupt Enable Register.....	11-12
11.2.4	FRNUM—Frame Number Register .....	11-12
11.2.5	FRBASEADD—Frame List Base Address .....	11-13
11.2.6	SOFMOD—Start of Frame Modify Register.....	11-13
11.2.7	PORTSC[0,1]—Port Status and Control Register.....	11-14
12	SMBus Controller Registers (D31:F3) .....	12-1
12.1	PCI Configuration Registers (SMBUS—D31:F3).....	12-1
12.1.1	VID—Vendor Identification Register (SMBUS—D31:F3).....	12-1
12.1.2	DID—Device Identification Register (SMBUS—D31:F3) .....	12-1
12.1.3	CMD—Command Register (SMBUS—D31:F3).....	12-2
12.1.4	STA—Device Status Register (SMBUS—D31:F3) .....	12-2
12.1.5	SCC—Sub Class Code Register (SMBUS—D31:F3).....	12-3
12.1.6	BCC—Base Class Code Register (SMBUS—D31:F3) .....	12-3
12.1.7	SMB_BASE—SMBus Base Address Register (SMBUS—D31:F3) .....	12-3
12.1.8	SVID—Subsystem Vendor ID (SMBUS—D31:F2/F4) .....	12-3
12.1.9	SID—Subsystem ID (SMBUS—D31:F2/F4) .....	12-4
12.1.10	INTR_LN—Interrupt Line Register (SMBUS—D31:F3) .....	12-4
12.1.11	INTR_PN—Interrupt Pin Register (SMBUS—D31:F3) .....	12-4
12.1.12	HOSTC—Host Configuration Register (SMBUS—D31:F3) .....	12-4
12.2	SMBus I/O Registers .....	12-5
12.2.1	HST_STS—Host Status Register .....	12-6
12.2.2	HST_CNT—Host Control Register .....	12-7
12.2.3	HST_CMD—Host Command Register.....	12-8
12.2.4	XMIT_SLVA—Transmit Slave Address Register .....	12-8
12.2.5	HST_D0—Data 0 Register.....	12-8
12.2.6	HST_D1—Data 1 Register.....	12-8

	12.2.7 BLOCK_DB—Block Data Byte Register .....	12-9
	12.2.8 RCV_SLVA—Receive Slave Address Register .....	12-9
	12.2.9 SLV_DATA—Receive Slave Data Register .....	12-9
	12.2.10 SMLINK_PIN_CTL—SMLINK Pin Control Register .....	12-10
	12.2.11 SMBUS_PIN_CTL SMBus Pin Control Register .....	12-10
13	AC'97 Audio Controller Registers (D31:F5) .....	13-1
13.1	AC'97 Audio PCI Configuration Space (D31:F5) .....	13-1
13.1.1	VID—Vendor Identification Register (Audio—D31:F5) .....	13-1
13.1.2	DID—Device Identification Register (Audio—D31:F5) .....	13-2
13.1.3	PCICMD—PCI Command Register (Audio—D31:F5) .....	13-2
13.1.4	PCISTS—PCI Device Status Register (Audio—D31:F5) .....	13-3
13.1.5	RID—Revision Identification Register (Audio—D31:F5) .....	13-3
13.1.6	PI—Programming Interface Register (Audio—D31:F5) .....	13-3
13.1.7	SCC—Sub Class Code Register (Audio—D31:F5) .....	13-4
13.1.8	BCC—Base Class Code Register (Audio—D31:F5) .....	13-4
13.1.9	HEDT—Header Type Register (Audio—D31:F5) .....	13-4
13.1.10	NAMBAR—Native Audio Mixer Base Address Register (Audio—D31:F5) .....	13-5
13.1.11	NABMBAR—Native Audio Bus Mastering Base Address Register (Audio—D31:F5) .....	13-5
13.1.12	SVID—Subsystem Vendor ID Register (Audio—D31:F5) .....	13-6
13.1.13	SID—Subsystem ID Register (Audio—D31:F5) .....	13-6
13.1.14	INTR_LN—Interrupt Line Register (Audio—D31:F5) .....	13-6
13.1.15	INTR_PN—Interrupt Pin Register (Audio—D31:F5) .....	13-7
13.2	AC'97 Audio I/O Space (D31:F5) .....	13-7
13.2.1	x_BDBAR—Buffer Descriptor Base Address Register .....	13-9
13.2.2	x_CIV—Current Index Value Register .....	13-10
13.2.3	x_LVI—Last Valid Index Register .....	13-10
13.2.4	x_SR—Status Register .....	13-11
13.2.5	x_PICB—Position In Current Buffer Register .....	13-12
13.2.6	x_PIV—Prefetched Index Value Register .....	13-12
13.2.7	x_CR—Control Register .....	13-13
13.2.8	GLOB_CNT—Global Control Register .....	13-14
13.2.9	GLOB_STA—Global Status Register .....	13-15
13.2.10	CAS—Codec Access Semaphore Register .....	13-16
14	AC'97 Modem Controller Registers (D31:F6) .....	14-1
14.1	AC'97 Modem PCI Configuration Space (D31:F6) .....	14-1
14.1.1	VID—Vendor Identification Register (Modem—D31:F6) .....	14-1
14.1.2	DID—Device Identification Register (Modem—D31:F6) .....	14-2
14.1.3	PCICMD—PCI Command Register (Modem—D31:F6) .....	14-2
14.1.4	PCISTA—Device Status Register (Modem—D31:F6) .....	14-3
14.1.5	RID—Revision Identification Register (Modem—D31:F6) .....	14-3
14.1.6	PI—Programming Interface Register (Modem—D31:F6) .....	14-3
14.1.7	SCC—Sub Class Code Register (Modem—D31:F6) .....	14-4
14.1.8	BCC—Base Class Code Register (Modem—D31:F6) .....	14-4
14.1.9	HEDT—Header Type Register (Modem—D31:F6) .....	14-4
14.1.10	MMBAR—Modem Mixer Base Address Register (Modem—D31:F6) .....	14-4
14.1.11	MBAR—Modem Base Address Register (Modem—D31:F6) .....	14-5
14.1.12	SVID—Subsystem Vendor ID (Modem—D31:F6) .....	14-5



	14.1.13 SID—Subsystem ID (Modem—D31:F6) .....	14-6
	14.1.14 INTR_LN—Interrupt Line Register (Modem—D31:F6) .....	14-6
	14.1.15 INT_PIN—Interrupt Pin (Modem—D31:F6) .....	14-6
14.2	AC'97 Modem I/O Space (D31:F6) .....	14-7
	14.2.1 x_BDBAR—Buffer Descriptor List Base Address Register .....	14-8
	14.2.2 x_CIV—Current Index Value Register .....	14-9
	14.2.3 x_LVI—Last Valid Index Register .....	14-9
	14.2.4 x_SR—Status Register .....	14-10
	14.2.5 x_PICB—Position In Current Buffer Register .....	14-11
	14.2.6 x_PIV—Prefetch Index Value Register .....	14-11
	14.2.7 x_CR—Control Register .....	14-11
	14.2.8 GLOB_CNT—Global Control Register .....	14-12
	14.2.9 GLOB_STA—Global Status Register .....	14-13
	14.2.10 CAS—Codec Access Semaphore Register .....	14-14
15	Pinout and Package Information .....	15-1
	15.1 Pinout .....	15-1
	15.2 Package Information .....	15-9
16	Electrical Characteristics .....	16-1
	16.1 Absolute Maximum Ratings .....	16-1
	16.2 Functional Operating Range .....	16-1
	16.3 D.C. Characteristics .....	16-2
	16.4 A.C. Characteristics .....	16-7
	16.5 Timing Diagrams .....	16-16
17	Testability .....	17-1
	17.1 Test Mode Description .....	17-1
	17.2 Tri-state Mode .....	17-2
	17.3 XOR Chain Mode .....	17-2
	17.3.1 XOR Chain Testability Algorithm Example .....	17-2
	17.3.1.1 Test Pattern Consideration for XOR Chain 4 .....	17-3
A	I/O Register Index .....	A-1
B	Register Bit Index .....	B-1

## Figures

2-1	Required External RTC Circuit .....	2-14
2-2	Example V5REF Sequencing Circuit .....	2-14
4-1	ICH2 and System Clock Domains .....	4-1
4-2	Conceptual System Clock Diagram .....	4-2
5-1	Primary Device Status Register Error Reporting Logic.....	5-2
5-2	Secondary Status Register Error Reporting Logic.....	5-3
5-3	NMI# Generation Logic.....	5-3
5-4	Integrated LAN Controller Block Diagram.....	5-6
5-5	64-Word EEPROM Read Instruction Waveform.....	5-16
5-6	LPC Interface Diagram .....	5-19
5-7	Typical Timing for LFRAME#.....	5-23
5-8	Abort Mechanism.....	5-23
5-9	ICH2 DMA Controller.....	5-25
5-10	DMA Serial Channel Passing Protocol .....	5-29
5-11	DMA Request Assertion Through LDRQ#.....	5-33
5-12	Coprocessor Error Timing Diagram .....	5-65
5-13	Signal Strapping .....	5-67
5-14	Physical Region Descriptor Table Entry .....	5-91
5-15	Transfer Descriptor .....	5-99
5-16	Example Queue Conditions .....	5-106
5-17	USB Data Encoding.....	5-109
5-18	USB Legacy Keyboard Flow Diagram .....	5-118
5-19	ICH2 Based AC'97 2.1.....	5-133
5-20	AC'97 2.1 Controller-Codec Connection.....	5-134
5-21	AC-link Protocol .....	5-135
5-22	AC-link Powerdown Timing.....	5-141
5-23	SDIN Wake Signaling .....	5-142
5-24	FWH Memory Cycle Preamble .....	5-145
5-25	Single Byte Read .....	5-145
5-26	Single Byte Write .....	5-146
15-1	ICH2 Ballout (Top view — Left side).....	15-2
15-2	ICH2 Ballout (Top view — Right side) .....	15-3
15-3	ICH2 Package (Top and Side Views) .....	15-9
15-4	ICH2 Package (Bottom View) .....	15-10
16-1	Clock Timing .....	16-16
16-2	Valid Delay From Rising Clock Edge.....	16-16
16-3	Setup And Hold Times.....	16-16
16-4	Float Delay.....	16-16
16-5	Pulse Width.....	16-17
16-6	Output Enable Delay.....	16-17
16-7	IDE PIO Mode.....	16-17
16-8	IDE Multiword DMA .....	16-18
16-9	Ultra ATA Mode (Drive Initiating a Burst Read) .....	16-18
16-10	Ultra ATA Mode (Sustained Burst) .....	16-19
16-11	Ultra ATA Mode (Pausing a DMA Burst) .....	16-19
16-12	Ultra ATA Mode (Terminating a DMA Burst) .....	16-20
16-13	USB Rise and Fall Times.....	16-20
16-14	USB Jitter.....	16-20
16-15	USB EOP Width.....	16-21



16-16	SMBus Transaction.....	16-21
16-17	SMBus Time-out .....	16-21
16-18	Power Sequencing and Reset Signal Timings.....	16-22
16-19	1.8V/3.3V Power Sequencing .....	16-22
16-20	G3 (Mechanical Off) to S0 Timings.....	16-23
16-21	S0 to S1 to S0 Timings .....	16-23
16-22	S0 to S5 to S0 Timings .....	16-24
16-23	C0 to C2 to C0 Timings.....	16-24
17-1	Test Mode Entry (XOR Chain Example) .....	17-1
17-2	Example XOR Chain Circuitry.....	17-2

## Tables

1-1	Industry Specifications.....	1-1
1-2	PCI Devices and Functions .....	1-3
2-1	Hub Interface Signals .....	2-1
2-2	LAN Connect Interface Signals.....	2-1
2-3	EEPROM Interface Signals .....	2-2
2-4	Firmware Hub Interface Signals .....	2-2
2-5	PCI Interface Signals .....	2-2
2-6	IDE Interface Signals .....	2-4
2-7	LPC Interface Signals .....	2-6
2-8	Interrupt Signals.....	2-6
2-9	USB Interface Signals.....	2-7
2-10	Power Management Interface Signals .....	2-7
2-11	Processor Interface Signals.....	2-8
2-12	SM Bus Interface Signals .....	2-9
2-13	System Management Interface Signals .....	2-9
2-14	Real Time Clock Interface .....	2-10
2-15	Other Clocks .....	2-10
2-16	Miscellaneous Signals .....	2-10
2-17	AC'97 Link Signals.....	2-11
2-18	General Purpose I/O Signals .....	2-11
2-19	Power and Ground Signals.....	2-12
2-20	Functional Strap Definitions.....	2-12
2-21	Test Mode Selection .....	2-13
3-1	ICH2 Power Planes .....	3-1
3-2	Integrated Pull-Up and Pull-Down Resistors .....	3-1
3-3	IDE Series Termination Resistors.....	3-2
3-4	Power Plane and States for Output and I/O Signals.....	3-3
3-5	Power Plane for Input Signals .....	3-5
5-1	Type 0 Configuration Cycle Device Number Translation.....	5-4
5-2	EEPROM Format.....	5-16
5-3	EEPROM Word Field Descriptions .....	5-17
5-4	LPC Cycle Types Supported .....	5-20
5-5	Start Field Bit Definitions .....	5-20
5-6	Cycle Type Bit Definitions.....	5-21
5-7	Transfer Size Bit Definition .....	5-21
5-8	SYNC Bit Definition.....	5-21
5-9	ICH2 Response to Sync Failures.....	5-22
5-10	DMA Transfer Size .....	5-27
5-11	Address Shifting in 16-bit I/O DMA Transfers .....	5-27
5-12	DMA Cycle vs. I/O Address .....	5-31
5-13	PCI Data Bus vs. DMA I/O Port Size .....	5-31
5-14	DMA I/O Cycle Width vs. BE[3:0]# .....	5-32
5-15	Counter Operating Modes .....	5-38
5-16	Interrupt Controller Core Connections .....	5-40
5-17	Interrupt Status Registers .....	5-41
5-18	Content of Interrupt Vector Byte .....	5-41
5-19	APIC Interrupt Mapping .....	5-48
5-20	Arbitration Cycles.....	5-49
5-21	APIC Message Formats.....	5-50



5-22	EOI Message .....	5-50
5-23	Short Message.....	5-51
5-24	APIC Bus Status Cycle Definition .....	5-52
5-25	Lowest Priority Message (Without Focus Processor) .....	5-53
5-26	Remote Read Message .....	5-54
5-27	Interrupt Message Address Format .....	5-57
5-28	Interrupt Message Data Format.....	5-57
5-29	Stop Frame Explanation .....	5-59
5-30	Data Frame Format.....	5-60
5-31	Configuration Bits Reset By RTCRST# Assertion .....	5-63
5-32	INIT# Going Active.....	5-65
5-33	NMI Sources .....	5-65
5-34	DP Signal Differences.....	5-66
5-35	Frequency Strap Behavior Based on Exit State.....	5-67
5-36	Frequency Strap Bit Mapping .....	5-67
5-37	General Power States for Systems using ICH2.....	5-68
5-38	State Transition Rules for ICH2 .....	5-69
5-39	System Power Plane.....	5-70
5-40	Causes of SMI# and SCI .....	5-71
5-41	Break Events.....	5-72
5-42	Sleep Types.....	5-74
5-43	Causes of Wake Events.....	5-74
5-44	GPI Wake Events.....	5-75
5-45	Sleep State Exit Latencies .....	5-75
5-46	Transitions Due To Power Failure .....	5-76
5-47	Transitions Due to Power Button .....	5-78
5-48	Transitions Due to RI# signal.....	5-79
5-49	Write Only Registers with Read Paths in Alternate Access Mode .....	5-80
5-50	PIC Reserved Bits Return Values .....	5-82
5-51	Register Write Accesses in Alternate Access Mode .....	5-82
5-52	ICH2 Clock Inputs .....	5-84
5-53	Alert on LAN* Message Data .....	5-87
5-54	IDE Transaction Timings (PCI Clocks) .....	5-90
5-55	Interrupt/Active Bit Interaction Definition.....	5-93
5-56	UltraATA/33 Control Signal Redefinitions .....	5-95
5-57	Frame List Pointer Bit Description .....	5-98
5-58	TD Link Pointer .....	5-99
5-59	TD Control and Status .....	5-100
5-60	TD Token .....	5-102
5-61	TD Buffer Pointer .....	5-102
5-62	Queue Head Block.....	5-103
5-63	Queue Head Link Pointer.....	5-103
5-64	Queue Element Link Pointer .....	5-103
5-65	Command Register, Status Register and TD Status Bit Interaction .....	5-105
5-66	Queue Advance Criteria.....	5-107
5-67	USB Schedule List Traversal Decision Table .....	5-108
5-68	PID Format.....	5-110
5-69	PID Types .....	5-111
5-70	Address Field.....	5-111
5-71	Endpoint Field .....	5-112
5-72	Token Format.....	5-113



5-73	SOF Packet .....	5-113
5-74	Data Packet Format .....	5-114
5-75	Bits maintained in low power states .....	5-117
5-76	USB Legacy Keyboard State Transitions .....	5-119
5-77	Quick Protocol .....	5-121
5-78	Send / Receive Byte Protocol .....	5-121
5-79	Write Byte/Word Protocol .....	5-122
5-80	Read Byte/Word Protocol .....	5-122
5-81	Process Call Protocol .....	5-123
5-82	Block Read/Write Protocol .....	5-125
5-83	I <sup>2</sup> C Block Read .....	5-126
5-84	Slave Write Cycle Format .....	5-128
5-85	Slave Write Registers .....	5-129
5-86	Command Types .....	5-129
5-87	Read Cycle Format .....	5-130
5-88	Data Values for Slave Read Registers .....	5-131
5-89	Featured Supported by ICH2 .....	5-132
5-90	AC'97 Signals .....	5-134
5-91	Input Slot 1 Bit Definitions .....	5-139
5-92	Output Tag Slot 0 .....	5-140
5-93	AC-link state during PCIRST# .....	5-143
6-1	PCI Devices and Functions .....	6-2
6-2	Fixed I/O Ranges Decoded by ICH2 .....	6-3
6-3	Variable I/O Decode Ranges .....	6-5
6-4	Memory Decode Ranges from Processor Perspective .....	6-6
7-1	PCI Configuration Map (LAN Controller—B1:D8:F0) .....	7-1
7-2	Configuration of Subsystem ID and Subsystem Vendor ID via EEPROM ....	7-6
7-3	Data Register Structure .....	7-10
7-4	ICH2 Integrated LAN Controller CSR Space .....	7-10
7-5	Self-Test Results Format .....	7-15
7-6	Statistical Counters .....	7-20
8-1	PCI Configuration Map (HUB-PCI—D30:F0) .....	8-1
9-1	PCI Configuration Map (LPC I/F—D31:F0) .....	9-1
9-2	DMA Registers .....	9-23
9-3	PIC Registers .....	9-33
9-4	APIC Direct Registers .....	9-41
9-5	APIC Indirect Registers .....	9-41
9-6	RTC I/O Registers .....	9-47
9-7	RTC (Standard) RAM Bank .....	9-47
9-8	PCI Configuration Map (PM—D31:F0) .....	9-53
9-9	APM Register Map .....	9-58
9-10	ACPI and Legacy I/O Register Map .....	9-59
9-11	TCO I/O Register Map .....	9-75
9-12	Summary of GPIO Implementation .....	9-81
9-13	Registers to Control GPIO .....	9-83
10-1	PCI Configuration Map (IDE—D31:F1) .....	10-1
10-2	Bus Master IDE I/O Registers .....	10-11
11-1	PCI Configuration Map (USB—D31:F2/F4) .....	11-1
11-2	USB I/O Registers .....	11-8
11-3	Run/Stop, Debug Bit Interaction SWDBG (Bit 5), Run/Stop (Bit 0) Operation .....	11-10



12-1	PCI Configuration Registers (SMBUS—D31:F3).....	12-1
12-2	SMB I/O Registers .....	12-5
13-1	PCI Configuration Map (Audio—D31:F5).....	13-1
13-2	ICH2 Audio Mixer Register Configuration .....	13-7
13-3	Native Audio Bus Master Control Registers.....	13-9
14-1	PCI Configuration Map (Modem—D31:F6).....	14-1
14-2	ICH2 Modem Mixer Register Configuration .....	14-7
14-3	Modem Registers .....	14-8
15-1	ICH2 EBGA Ball List .....	15-4
16-1	D.C. Current Characteristics (ICH2 Power Consumption Measurements) .	16-2
16-2	DC Characteristic Input Signal Association .....	16-2
16-3	DC Input Characteristics .....	16-3
16-4	DC Characteristic Output Signal Association.....	16-4
16-5	DC Output Characteristics .....	16-5
16-6	Other DC Characteristics .....	16-6
16-7	Clock Timings .....	16-7
16-8	PCI Interface Timing .....	16-9
16-9	IDE PIO & Multiword DMA Mode Timing .....	16-10
16-10	Ultra ATA Timing (Mode 0, Mode 1, Mode 2) .....	16-11
16-11	Ultra ATA Timing (Mode 3, Mode 4, Mode 5) .....	16-11
16-12	Universal Serial Bus Timing.....	16-12
16-13	IOAPIC Bus Timing.....	16-13
16-14	SMBus Timing.....	16-13
16-15	AC'97 Timing .....	16-13
16-16	LPC Timing .....	16-13
16-17	Miscellaneous Timings.....	16-14
16-18	Power Sequencing and Reset Signal Timings.....	16-14
16-19	Power Management Timings .....	16-15
17-1	Test Mode Selection .....	17-1
17-2	XOR Test Pattern Example.....	17-2
17-3	XOR Chain #1 (RTCRST# Asserted for 4 PCI Clocks while PWROK Active) .....	17-4
17-4	XOR Chain #2 (RTCRST# Asserted for 5 PCI clocks while PWROK Active) .....	17-5
17-5	XOR Chain #3 (RTCRST# Asserted for 6 PCI Clocks while PWROK Active) .....	17-6
17-6	XOR Chain #4 (RTCRST# Asserted for 7 PCI Clocks while PWROK Active) .....	17-7
17-7	Signals Not in XOR Chain.....	17-8
A-1	ICH2 Fixed I/O Registers .....	A-1
A-2	ICH2 Variable I/O Registers.....	A-6



## Revision History

Revision	Description	Date
-001	Initial Release.	June 2000

# Introduction

# 1

The Intel® 82801BA ICH2 is a highly integrated multifunctional I/O Controller Hub that provides the interface to the PCI Bus and integrates many of the functions needed in today's PC platforms. The ICH2 communicates with the host controller over a dedicated hub interface. This datasheet provides a detailed description of the ICH2's functions and capabilities including, signals, registers, on-chip functional units, interfaces, pinout, packaging, electrical characteristics, and testability.

## 1.1 About this Document

This datasheet is intended for Original Equipment Manufacturers and BIOS vendors creating ICH2-based products. This document assumes a working knowledge of the vocabulary and principles of USB, IDE, AC'97, SMBus, PCI, ACPI, LAN, and LPC. Although some details of these features are described within this document, refer to the individual industry specifications listed in [Table 1-1](#) for the complete details.

**Table 1-1. Industry Specifications**

Specification	Location
LPC	<a href="http://developer.intel.com/design/pcisets/lpc/">http://developer.intel.com/design/pcisets/lpc/</a>
AC'97	<a href="http://developer.intel.com/pc-supply/platform/ac97/">http://developer.intel.com/pc-supply/platform/ac97/</a>
WfM	<a href="http://developer.intel.com/ial/WfM/usesite.htm">http://developer.intel.com/ial/WfM/usesite.htm</a>
SMBus	<a href="http://www.sbs-forum.org/specs.htm">http://www.sbs-forum.org/specs.htm</a>
PCI	<a href="http://pcisig.com/specs.htm">http://pcisig.com/specs.htm</a>
USB	<a href="http://www.usb.org">http://www.usb.org</a>
ACPI	<a href="http://www.teleport.com/~acpi/">http://www.teleport.com/~acpi/</a>

### Chapter 1. Introduction

Chapter 1 introduces the ICH2 and provides information on document organization. This chapter also describes the key features of the ICH2 and provides a brief description of the major functions.

### Chapter 2. Signal Description

Chapter 2 provides a detailed description of each ICH2 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. ICH2 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. ICH2 and System Clock Domains

Chapter 4 provides a list of each clock domain associated with the ICH2 in an ICH2-based system.

### Chapter 5. Functional Description

Chapter 5 provides a detailed description of the functions in the ICH2. All PCI buses, devices and functions in this manual are abbreviated using the following nomenclature; Bus:Device:Function. This datasheet abbreviates buses as B0 and B1, devices as D8, D30 and D31 and functions as F0, F1, F2, F3, F4, F5 and F6. 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 ICH2's external PCI bus is typically Bus 1; however, it may be assigned a different number depending on system configuration.

**Chapter 6. Register, Memory and I/O Address Maps**

Chapter 6 provides an overview of the registers, fixed I/O ranges, variable I/O ranges and memory ranges decoded by the ICH2.

**Chapter 7. LAN Controller Registers**

Chapter 7 provides a detailed description of all registers that reside in the ICH2's integrated LAN Controller. The integrated LAN Controller resides on the ICH2'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 ICH2 including DMA, Timers, Interrupts, CPU 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. USB Controller Registers**

Chapter 11 provides a detailed description of all registers that reside in the two USB controllers. These controllers reside at Device 31, Functions 2 and 4 (D31:F2/F4).

**Chapter 12. SMBus Controller Registers**

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

**Chapter 13. AC'97 Audio Controller Registers**

Chapter 13 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 section 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 14. AC'97 Modem Controller Registers**

Chapter 14 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 section 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 15. Pinout and Package Information**

Chapter 15 provides the ball assignment for the 360 EBGA package. The chapter also provides the physical dimensions and characteristics of the 360 EBGA package.

**Chapter 16. Electrical Characteristics**

Chapter 16 provides the AC and DC characteristics including timing diagrams.

**Chapter 17. Testability**

Chapter 17 provides details about the implementation of test modes on the ICH2.

**Index**

There are indexes listing registers and register bits.

## 1.2 Overview

The ICH2 (82801BA) provides extensive I/O support. Functions and capabilities include:

- PCI Rev 2.2 compliant 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)
- USB host interface with support for 4 USB ports; 2 host controllers
- Integrated LAN Controller
- System Management Bus (SMBus) with additional support for I<sup>2</sup>C devices
- AC'97 2.1 Compliant Link for Audio and Telephony codecs (up to 6 channels)
- Low Pin Count (LPC) interface
- Firmware Hub (FWH) interface support
- Alert On LAN\* (AOL) and Alert On LAN 2 (AOL2)\*

The ICH2 incorporates a variety of PCI functions that are divided into two logical devices (30 and 31) on PCI Bus 0 and one device on Bus 1. Device 30 is the Hub Interface-To-PCI bridge. Device 31 contains all the other PCI functions, except the LAN Controller as shown in [Table 1-2](#). The LAN controller is located on Bus 1.

**Table 1-2. 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 (includes: DMA, Timers, compatible interrupt controller, APIC, RTC, processor interface control, power management control, System Management control, and GPIO control)
Bus 0:Device 31:Function 1	IDE Controller
Bus 0:Device 31:Function 2	USB Controller #1
Bus 0:Device 31:Function 3	SMBus Controller
Bus 0:Device 31:Function 4	USB Controller #2
Bus 0:Device 31:Function 5	AC'97 Audio Controller
Bus 0:Device 31:Function 6	AC'97 Modem Controller
Bus 1:Device 8:Function 0	LAN Controller

The following sub-sections provide an overview of the ICH2 capabilities.

## Hub Architecture

As I/O speeds increase, the demand placed on the PCI bus by the I/O bridge has become significant. With the addition of AC'97 and Ultra ATA/100, coupled with the existing USB, I/O requirements could impact PCI bus performance. The chipset's *hub interface architecture* ensures that the I/O subsystem; both PCI and the integrated I/O features (IDE, AC'97, USB, etc.), will receive adequate bandwidth. By placing the I/O bridge on the hub interface (instead of PCI), the hub architecture ensures that both the I/O functions integrated into the ICH2 and the PCI peripherals obtain the bandwidth necessary for peak performance.

## PCI Interface

The ICH2 PCI interface provides a 33 MHz, Rev. 2.2 compliant implementation. All PCI signals are 5V tolerant, except PME#. The ICH2 integrates a PCI arbiter that supports up to six external PCI bus masters in addition to the internal ICH2 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 CD ROMs. Each IDE device can have independent timings. The IDE interface supports PIO IDE transfers up to 14 Mbytes/sec and Bus Master IDE transfers up to 100 Mbytes/sec. It does not consume any ISA DMA resources. The IDE interface integrates 16x32-bit buffers for optimal transfers.

The ICH2'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 5.15, "IDE Controller \(D31:F1\)"](#) on page 5-89 for details).

## Low Pin Count (LPC) Interface

The ICH2 implements an LPC Interface as described in the LPC 1.0 specification. The Low Pin Count (LPC) Bridge function of the ICH2 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.

Note that in the current chipset platform, the Super I/O (SIO) component has migrated to the Low Pin Count (LPC) interface. Migration to the LPC interface allows for lower cost Super I/O designs.

## 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 ICH2 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 ICH2'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 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.

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 ICH2 provides an ISA-Compatible interrupt controller 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 ICH2 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 circuit.

## Advanced Programmable Interrupt Controller (APIC)

In addition to the standard ISA compatible interrupt controller (PIC) described in the previous section, the ICH2 incorporates the Advanced Programmable Interrupt Controller (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.

## Enhanced Universal Serial Bus (USB) Controller

The USB controller provides enhanced support for the Universal Host Controller Interface (UHCI). This includes support that allows legacy software to use a USB-based keyboard and mouse. The ICH2 is USB Revision 1.1 compliant. The ICH2 contains two USB Host Controllers. Each Host Controller includes a root hub with two separate USB ports each, for a total of 4 USB ports. See [Section 5.16, “USB Controller \(Device 31:Functions 2 and 4\)” on page 5-98](#) for details.

## LAN Controller

The ICH2'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, "LAN Controller \(B1:D8:F0\)" on page 5-5](#) for details.

## RTC

The ICH2 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 3V lithium battery that provides up to 7 years of protection.

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 ICH2 configuration.

## Enhanced Power Management

The ICH2'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 ICH2 contains full support for the Advanced Configuration and Power Interface (ACPI) Specification.

## System Management Bus (SMBus)

The ICH2 contains an SMBus Host interface that allows the processor to communicate with SMBus slaves. This interface is compatible with most I<sup>2</sup>C devices. Special I<sup>2</sup>C commands are implemented (e.g., the I<sup>2</sup>C Read that allows the ICH2 to perform block reads of I<sup>2</sup>C devices).

The ICH2's SMBus host controller provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). The host controller supports seven SMBus interface command protocols for communicating with SMBus slave devices (see System Management Bus Specifications, Rev 1.0): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Process Call, and Block Read/Write.

## Manageability

The ICH2 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 ICH2'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 ICH2 looks for the processor to fetch the first instruction after reset. If the processor does not fetch the first instruction, the ICH2 will reboot the system at the safe-mode frequency multiplier.
- **ECC Error Reporting.** When detecting an ECC error, the host controller has the ability to send one of several messages to the ICH2. The host controller can instruct the ICH2 to generate either an SMI#, NMI, SERR#, or TCO interrupt.
- **Function Disable.** The ICH2 provides the ability to disable the following functions: AC'97 Modem, AC'97 Audio, IDE, USB, 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 ICH2 provides an input signal (INTRUDER#) that can be attached to a switch that is activated by the system case being opened. The ICH2 can be programmed to generate an SMI# or TCO interrupt due to an active INTRUDER# signal.
- **SMBus.** The ICH2 integrates an SMBus controller that provides an interface to manage peripherals (e.g., serial presence detection (SPD) or RIMMs and thermal sensors).
- **Alert-On-LAN\*.** The ICH2 supports Alert-On-LAN\* and Alert-On-LAN\* 2. In response to a TCO event (intruder detect, thermal event, processor not booting) the ICH2 sends a message over the SMBus. A LAN controller can decode this SMBus message and send a message over the network to alert the network manager.

## AC'97 2.1 Controller

The *Audio Codec '97* (AC'97) 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 both an AC and an MC. The AC'97 specification defines the interface between the system logic and the audio or modem codec, known as the *AC'97 Digital Link*.

The ICH2's AC'97 (with the appropriate codecs) not only replaces ISA audio and modem functionality, but also improves overall platform integration by incorporating the AC'97 digital link. The use of the ICH2-integrated AC'97 digital link reduces cost and eases migration from ISA.

By using an audio codec, the AC'97 digital 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 ICH2-integrated digital link allows several external codecs to be connected to the ICH2. 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 two audio codecs or a combination of an audio and 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 codec can be on-board and the modem codec can be placed on a riser. Intel is developing an AC'97 digital link connector. With a single integrated codec, or AMC, both audio and modem can be routed to a connector near the rear panel, where the external ports can be located.

The digital link in the ICH2 is compliant with revision 2.1 of the AC'97, so it supports two 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 ICH2 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 Woofer, for a complete surround-sound effect. ICH2 has expanded support for two audio codecs on the AC'97 digital link.

# 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:

<b>I</b>	Input Pin
<b>O</b>	Output Pin
<b>OD</b>	Open Drain Output Pin.
<b>I/O</b>	Bi-directional Input / Output Pin.

## 2.1 Hub Interface to Host Controller

Table 2-1. Hub Interface Signals

Name	Type	Description
HL[11:0]	I/O	<b>Hub Interface Signals</b>
HL_STB	I/O	<b>Hub Interface Strobe:</b> One of two differential strobe signals used to transmit and receive data through the hub interface.
HL_STB#	I/O	<b>Hub Interface Strobe Complement:</b> Second of the two differential strobe signals.
HLCOMP	I/O	<b>Hub Interface Compensation:</b> Used for hub interface buffer compensation.

## 2.2 Link to LAN Connect

Table 2-2. LAN Connect Interface Signals

Name	Type	Description
LAN_CLK	I	<b>LAN Interface Clock:</b> This signal is driven by the LAN Connect component. The frequency range is 0.8 MHz to 50 MHz.
LAN_RXD[2:0]	I	<b>Received Data:</b> The 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	<b>Transmit Data:</b> The integrated LAN Controller uses these signals to transfer data and control information to the LAN Connect component.
LAN_RSTSYNC	O	<b>LAN Reset/Sync:</b> The LAN Connect component's Reset and Sync signals are multiplexed onto this pin.

## 2.3 EEPROM Interface

Table 2-3. EEPROM Interface Signals

Name	Type	Description
EE_SHCLK	O	<b>EEPROM Shift Clock:</b> EE_SHCLK is the serial shift clock output to the EEPROM.
EE_DIN	I	<b>EEPROM Data In:</b> EE_DIN transfers data from the EEPROM to the ICH2. This signal has an integrated pull-up resistor.
EE_DOUT	O	<b>EEPROM Data Out:</b> EE_DOUT transfers data from the ICH2 to the EEPROM.
EE_CS	O	<b>EEPROM Chip Select:</b> EE_CS is a chip-select signal to the EEPROM.

## 2.4 Firmware Hub Interface

Table 2-4. Firmware Hub Interface Signals

Name	Type	Description
FWH[3:0] / LAD[3:0]	I/O	<b>Firmware Hub Signals.</b> These signals are muxed with LPC address signals.
FWH[4] / LFRAME#	I/O	<b>Firmware Hub Signals.</b> This signal is muxed with LPC LFRAME# signal.

## 2.5 PCI Interface

Table 2-5. PCI Interface Signals

Name	Type	Description																								
AD[31:0]	I/O	<b>PCI Address/Data:</b> AD[31:0] is a multiplexed address and data bus. During the first clock of a transaction, AD[31:0] contain a physical address (32 bits). During subsequent clocks, AD[31:0] contain data. The ICH2 drives all 0s on AD[31:0] during the address phase of all PCI Special Cycles.																								
C/BE[3:0]#	I/O	<b>Bus Command and Byte Enables:</b> 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. <table><thead><tr><th>C/BE[3:0]#</th><th>Command Type</th></tr></thead><tbody><tr><td>0000</td><td>Interrupt Acknowledge</td></tr><tr><td>0001</td><td>Special Cycle</td></tr><tr><td>0010</td><td>I/O Read</td></tr><tr><td>0011</td><td>I/O Write</td></tr><tr><td>0110</td><td>Memory Read</td></tr><tr><td>0111</td><td>Memory Write</td></tr><tr><td>1010</td><td>Configuration Read</td></tr><tr><td>1011</td><td>Configuration Write</td></tr><tr><td>1100</td><td>Memory Read Multiple</td></tr><tr><td>1110</td><td>Memory Read Line</td></tr><tr><td>1111</td><td>Memory Write and Invalidate</td></tr></tbody></table> <p>All command encodings not shown are reserved. The ICH2 does not decode reserved values, and therefore will not respond if a PCI master generates a cycle using one of the reserved values.</p>	C/BE[3:0]#	Command Type	0000	Interrupt Acknowledge	0001	Special Cycle	0010	I/O Read	0011	I/O Write	0110	Memory Read	0111	Memory Write	1010	Configuration Read	1011	Configuration Write	1100	Memory Read Multiple	1110	Memory Read Line	1111	Memory Write and Invalidate
C/BE[3:0]#	Command Type																									
0000	Interrupt Acknowledge																									
0001	Special Cycle																									
0010	I/O Read																									
0011	I/O Write																									
0110	Memory Read																									
0111	Memory Write																									
1010	Configuration Read																									
1011	Configuration Write																									
1100	Memory Read Multiple																									
1110	Memory Read Line																									
1111	Memory Write and Invalidate																									

Table 2-5. PCI Interface Signals (Continued)

Name	Type	Description
DEVSEL#	I/O	<b>Device Select:</b> The ICH2 asserts DEVSEL# to claim a PCI transaction. As an output, the ICH2 asserts DEVSEL# when a PCI master peripheral attempts an access to an internal ICH2 address or an address destined for the hub interface (main memory or AGP). As an input, DEVSEL# indicates the response to an ICH2-initiated transaction on the PCI bus. DEVSEL# is tri-stated from the leading edge of PCIRST#. DEVSEL# remains tri-stated by the ICH2 until driven by a target device.
FRAME#	I/O	<b>Cycle Frame:</b> 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 ICH2 when the ICH2 is the target, and FRAME# is an output from the ICH2 when the ICH2 is the Initiator. FRAME# remains tri-stated by the ICH2 until driven by an Initiator.
IRDY#	I/O	<b>Initiator Ready:</b> IRDY# indicates the ICH2'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 ICH2 has valid data present on AD[31:0]. During a read, it indicates the ICH2 is prepared to latch data. IRDY# is an input to the ICH2 when the ICH2 is the Target and an output from the ICH2 when the ICH2 is an Initiator. IRDY# remains tri-stated by the ICH2 until driven by an Initiator.
TRDY#	I/O	<b>Target Ready:</b> TRDY# indicates the ICH2'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 ICH2, as a Target, has placed valid data on AD[31:0]. During a write, TRDY# indicates the ICH2, as a Target is prepared to latch data. TRDY# is an input to the ICH2 when the ICH2 is the Initiator and an output from the ICH2 when the ICH2 is a Target. TRDY# is tri-stated from the leading edge of PCIRST#. TRDY# remains tri-stated by the ICH2 until driven by a target.
STOP#	I/O	<b>Stop:</b> STOP# indicates that the ICH2, as a Target, is requesting the Initiator to stop the current transaction. STOP# causes the ICH2, as an Initiator, to stop the current transaction. STOP# is an output when the ICH2 is a target and an input when the ICH2 is an Initiator. STOP# is tri-stated from the leading edge of PCIRST#. STOP# remains tri-stated until driven by the ICH2.
PAR	I/O	<b>Calculated/Checked Parity:</b> PAR uses "even" parity calculated on 36 bits, AD[31:0] plus C/BE[3:0]#. "Even" parity means that the ICH2 counts the number of 1s within the 36 bits plus PAR and the sum is always even. The ICH2 always calculates PAR on 36 bits, regardless of the valid byte enables. The ICH2 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 ICH2 drives and tri-states PAR identically to the AD[31:0] lines except that the ICH2 delays PAR by exactly one PCI clock. PAR is an output during the address phase (delayed one clock) for all ICH2 initiated transactions. PAR is an output during the data phase (delayed one clock) when the ICH2 is the Initiator of a PCI write transaction, and when it is the target of a read transaction. ICH2 checks parity when it is the target of a PCI write transaction. If a parity error is detected, the ICH2 sets the appropriate internal status bits, and has the option to generate an NMI# or SMI#.
PERR#	I/O	<b>Parity Error:</b> An external PCI device drives PERR# when it receives data that has a parity error. The ICH2 drives PERR# when it detects a parity error. The ICH can either generate an NMI# or SMI# upon detecting a parity error (either detected internally or reported via the PERR# signal).
REQ[0:4]# REQ[5]# / REQ[B]# / GPIO[1]	I	<b>PCI Requests:</b> The ICH2 supports up to 6 masters on the PCI bus. REQ[5]# is muxed with PC/PCI REQ[B]# (must choose one or the other, but not both). If not used for PCI or PC/PCI, REQ[5]#/REQ[B]# can instead be used as GPIO[1].  Note: REQ[0]# is programmable to have improved arbitration latency for supporting PCI-based 1394 controllers.

Table 2-5. PCI Interface Signals (Continued)

Name	Type	Description
<b>GNT[0:4]#</b> <b>GNT[5]# /</b> <b>GNT[B]# /</b> <b>GPIO[17]#</b>	O	<b>PCI Grants:</b> Supports up to 6 masters on the PCI bus. GNT[5]# is muxed with PC/PCI GNT[B]# (must choose one or the other, but not both). If not needed for PCI or PC/PCI, GNT[5]# can instead be used as a GPIO.  Pull-up resistors are not required on these signals. If pullups are used, they should be tied to the Vcc3_3 power rail. GNT[B]#/GNT[5]#/GPIO[17] has an internal pull-up.
<b>PCICLK</b>	I	<b>PCI Clock:</b> 33 MHz clock. PCICLK provides timing for all transactions on the PCI Bus.
<b>PCIRST#</b>	O	<b>PCI Reset:</b> ICH2 asserts PCIRST# to reset devices that reside on the PCI bus. The ICH2 asserts PCIRST# during power-up and when S/W initiates a hard reset sequence through the RC (CF9h) register. The ICH2 drives PCIRST# inactive a minimum of 1 ms after PWROK is driven active. The ICH2 drives PCIRST# active a minimum of 1 ms when initiated through the RC register.
<b>PLOCK#</b>	I/O	<b>PCI Lock:</b> PLOCK# indicates an exclusive bus operation and may require multiple transactions to complete. ICH2 asserts PLOCK# when it performs non-exclusive transactions on the PCI bus. PLOCK# is ignored when PCI masters are granted the bus.
<b>SERR#</b>	I	<b>System Error:</b> SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, the ICH2 has the ability to generate an NMI, SMI#, or interrupt.
<b>PME#</b>	I	<b>PCI Power Management Event:</b> 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 ICH2 may drive PME# active due to an internal wake event. The ICH2 will not drive PME# high, but it will be pulled up to VccSus3_3 by an internal pull-up resistor.
<b>REQ[A]# /</b> <b>GPIO[0]</b> <b>REQ[B]# /</b> <b>REQ[5]# /</b> <b>GPIO[1]</b>	I	<b>PC/PCI DMA Request [A:B]:</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.  When not used for PC/PCI requests, these signals can be used as General Purpose Inputs. Instead, REQ[B]# can be used as the 6th PCI bus request.
<b>GNT[A]# /</b> <b>GPIO[16]</b> <b>GNT[B]# /</b> <b>GNT[5]# /</b> <b>GPIO[17]</b>	O	<b>PC/PCI DMA Acknowledges [A:B]:</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 I/O or audio codecs which need to perform legacy 8237 DMA but have no ISA bus.  When not used for PC/PCI, these signals can be used as General Purpose Outputs. GNT[B]# can also be used as the 6th PCI bus master grant output. These signal have internal pull-up resistors.

## 2.6 IDE Interface

Table 2-6. IDE Interface Signals

Name	Type	Description
<b>PDCS1#,</b> <b>SDCS1#</b>	O	<b>Primary and Secondary IDE Device Chip Selects for 100 Range:</b> These signals are for the ATA command register block. This output signal is connected to the corresponding signal on the primary or secondary IDE connector.
<b>PDCS3#,</b> <b>SDCS3#</b>	O	<b>Primary and Secondary IDE Device Chip Select for 300 Range:</b> These signals are for the ATA control register block. This output signal is connected to the corresponding signal on the primary or secondary IDE connector.

Table 2-6. IDE Interface Signals (Continued)

Name	Type	Description
PDA[2:0], SDA[2:0]	O	<b>Primary and Secondary IDE Device Address:</b> 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	<b>Primary and Secondary IDE Device Data:</b> These signals directly drive the corresponding signals on the primary or secondary IDE connector. There is a weak internal pull-down resistor on PDD[7] and SDD[7].
PDDREQ, SDDREQ	I	<b>Primary and Secondary IDE Device DMA Request:</b> 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. They are not associated with any AT-compatible DMA channel. There is a weak internal pull-down resistor on these signals.
PDDACK#, SDDACK#	O	<b>Primary and Secondary IDE Device DMA Acknowledge:</b> These signals directly drive the DAK# signals on the primary and secondary IDE connectors. Each signal is asserted by the ICH2 to indicate to the 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.
PDIOR#  SDIOR#	O	<b>Primary and Secondary Disk I/O Read (PIO and Non-Ultra DMA):</b> This is the command to the IDE device that it may drive data on the PDD or SDD lines. Data is latched by the ICH2 on the deassertion edge of PDIOR# or SDIOR#. The IDE device is selected either by the ATA register file chip selects (PDSCS1# or SDSCS1#, PDSCS3# or SDSCS3#) and the PDA or SDA lines, or the IDE DMA acknowledge (PDDAK# or SDDAK#).  <b>Primary and Secondary Disk Write Strobe (Ultra DMA Writes to Disk):</b> This is the data write strobe for writes to disk. When writing to disk, ICH2 drives valid data on rising and falling edges of PDWSTB or SDWSTB.  <b>Primary and Secondary Disk DMA Ready (Ultra DMA Reads from Disk):</b> This is the DMA ready for reads from disk. When reading from disk, ICH2 deasserts PRDMARDY# or SRDMARDY# to pause burst data transfers.
PDIOW#  SDIOW#	O	<b>Primary and Secondary Disk I/O Write (PIO and Non-Ultra DMA):</b> 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 (PDSCS1# or SDSCS1#, PDSCS3# or SDSCS3#) and the PDA or SDA lines, or the IDE DMA acknowledge (PDDAK# or SDDAK#).  <b>Primary and Secondary Disk Stop (Ultra DMA):</b> ICH2 asserts this signal to terminate a burst.
PIORDY  SIORDY	I	<b>Primary and Secondary I/O Channel Ready (PIO):</b> This signal keeps 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.  <b>Primary and Secondary Disk Read Strobe (Ultra DMA Reads from Disk):</b> When reading from disk, ICH2 latches data on rising and falling edges of this signal from the disk.  <b>Primary and Secondary Disk DMA Ready (Ultra DMA Writes to Disk):</b> When writing to disk, this is deasserted by the disk to pause burst data transfers.

## 2.7 LPC Interface

Table 2-7. LPC Interface Signals

Name	Type	Description
<b>LAD[3:0] / FWH[3:0]</b>	I/O	<b>LPC Multiplexed Command, Address, Data:</b> Internal pull-ups are provided.
<b>LFRAME# / FWH[4]</b>	O	<b>LPC Frame:</b> Indicates the start of an LPC cycle, or an abort.
<b>LDRQ[1:0]#</b>	I	<b>LPC Serial DMA/Master Request Inputs:</b> These signals are used to request DMA or bus master access. Typically, they are connected to external Super I/O device. An internal pull-up resistor is provided on these signals.

## 2.8 Interrupt Interface

Table 2-8. Interrupt Signals

Name	Type	Description
<b>SERIRQ</b>	I/O	<b>Serial Interrupt Request:</b> This pin implements the serial interrupt protocol.
<b>PIRQ[D:A]#</b>	I/OD	<b>PCI Interrupt Requests:</b> In Non-APIC Mode the PIRQx# signals can be routed to interrupts 3:7, 9:12, 14, or 15 as described in the Interrupt Steering section. Each PIRQx# line has a separate Route Control Register. In APIC mode, these signals are connected to the internal I/O APIC in the following fashion: PIRQ[A]# is connected to IRQ16, PIRQ[B]# to IRQ17, PIRQ[C]# to IRQ18, and PIRQ[D]# to IRQ19. This frees the ISA interrupts.
<b>PIRQ[H]#, PIRQ[G:F]# / GPIO[4:3], PIRQ[E]#</b>	I/OD	<b>PCI Interrupt Requests:</b> In Non-APIC Mode the PIRQx# signals can be routed to interrupts 3:7, 9:12, 14 or 15 as described in the Interrupt Steering section. Each PIRQx# line has a separate Route Control Register. In APIC mode, these signals are connected to the internal I/O APIC in the following fashion: PIRQ[E]# is connected to IRQ20, PIRQ[F]# to IRQ21, PIRQ[G]# to IRQ22, and PIRQ[H]# to IRQ23. This frees the ISA interrupts. If not needed for interrupts, PIRQ[G:F] can be used as GPIO.
<b>IRQ[14:15]</b>	I	<b>Interrupt Request 14:15:</b> 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.
<b>APICCLK</b>	I	<b>APIC Clock:</b> The APIC clock runs at 33.333 MHz.
<b>APICD[1:0]</b>	I/OD	<b>APIC Data:</b> These bi-directional open drain signals are used to send and receive data over the APIC bus. As inputs, the data is valid on the rising edge of APICCLK. As outputs, new data is driven from the rising edge of the APICCLK.

## 2.9 USB Interface

Table 2-9. USB Interface Signals

Name	Type	Description
USBP0P, USBP0N, USBP1P, USBP1N	I/O	<b>Universal Serial Bus Port 1:0 Differential:</b> These differential pairs are used to transmit Data/Address/Command signals for ports 0 and 1 (USB Controller 1).
USBP2P, USBP2N, USBP3P, USBP3N	I/O	<b>Universal Serial Bus Port 3:2 Differential:</b> These differential pairs are used to transmit Data/Address/Command signals for ports 2 and 3 (USB Controller 2).
OC[3:0]#	I	<b>Overcurrent Indicators:</b> These signals set corresponding bits in the USB controllers to indicate that an overcurrent condition has occurred.

## 2.10 Power Management Interface

Table 2-10. Power Management Interface Signals

Name	Type	Description
THRM#	I	<b>Thermal Alarm:</b> THRM# is an active low signal generated by external hardware to start the hardware clock throttling mode. This signal can also generate an SMI# or an SCI.
SLP_S3#	O	<b>S3 Sleep Control:</b> Power plane control. This signal is used to shut off power to all non-critical systems when in S3 (Suspend To RAM), S4 (Suspend To Disk) or S5 (Soft Off) states.
SLP_S5#	O	<b>S5 Sleep Control:</b> Power plane control. This signal is used to shut power off to all non-critical systems when in the S4 (Suspend To Disk) or S5 (Soft Off) states.
PWROK	I	<b>Power OK:</b> When asserted, PWROK is an indication to the ICH2 that core power and PCICLK have been stable for at least 1 ms. PWROK can be driven asynchronously. When PWROK is negated, the ICH2 asserts PCIRST#.
PWRBTN#	I	<b>Power Button:</b> The Power Button will cause SMI# or SCI to indicate a system request to go to a sleep state. If the system is already in a sleep state, this signal will cause a wake event. If PWRBTN# is pressed for more than 4 seconds, this will cause an unconditional transition (power button override) to the S5 state with only the PWRBTN# available as a wake event. Override will occur even if the system is in the S1-S4 states. This signal has an internal pull-up resistor.
RI#	I	<b>Ring Indicate:</b> From the modem interface. This signal can be enabled as a wake event; this is preserved across power failures.
RSMRST#	I	<b>Resume Well Reset:</b> Used for resetting the resume power plane logic.
RSM_PWROK	I	<b>Resume Well Power OK:</b> When asserted, this signal is an indication to the ICH2 that the resume well power (VccSus3_3, VccSus1_8) has been stable for at least 10 ms.
SUS_STAT# / LPCPD#	O	<b>Suspend Status:</b> This signal is asserted by the ICH2 to indicate that the system will be entering a low power state soon. This can be monitored by devices with memory that need to switch from normal refresh to suspend refresh mode. It can also be used by other peripherals as an indication that they should isolate their outputs that may be going to powered-off planes. This signal is called LPCPD# on the LPC interface.
SUSCLK	O	<b>Suspend Clock:</b> This signal is an output of the RTC generator circuit and is used by other chips for the refresh clock.
VRMPWRGD	I	<b>VRM Power Good:</b> This should be connected to be the processor's VRM Power Good.

## 2.11 Processor Interface

Table 2-11. Processor Interface Signals

Name	Type	Description
A20M#	O	<p><b>Mask A20:</b> A20M# will go active based on either setting the appropriate bit in the Port 92h register, or based on the A20GATE input being active.</p> <p><b>Speed Strap:</b> During the reset sequence, ICH2 drives A20M# high if the corresponding bit is set in the FREQ_STRP register.</p>
CPUSLP#	O	<p><b>Processor Sleep:</b> 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 ICH2 can optionally assert the CPUSLP# signal when going to the S1 state.</p>
FERR#	I	<p><b>Numeric Coprocessor Error:</b> This signal is tied to the coprocessor error signal on the processor. FERR# is only used if the ICH2 coprocessor error reporting function is enabled in the General Control Register (Device 31:Function 0, Offset D0, bit 13). If FERR# is asserted, the ICH2 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.</p>
IGNNE#	O	<p><b>Ignore Numeric Error:</b> This signal is connected to the ignore error pin on the processor. IGNNE# is only used if the ICH2 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.</p> <p><b>Speed Strap:</b> During the reset sequence, ICH2 drives IGNNE# high if the corresponding bit is set in the FREQ_STRP register.</p>
INIT#	O	<p><b>Initialization:</b> INIT# is asserted by the ICH2 for 16 PCI clocks to reset the processor. ICH2 can be configured to support processor BIST. In that case, INIT# will be active when PCIRST# is active.</p>
INTR	O	<p><b>Processor Interrupt:</b> INTR is asserted by the ICH2 to signal the processor that an interrupt request is pending and needs to be serviced. It is an asynchronous output and normally driven low.</p> <p><b>Speed Strap:</b> During the reset sequence, ICH2 drives INTR high if the corresponding bit is set in the FREQ_STRP register.</p>
NMI	O	<p><b>Non-Maskable Interrupt:</b> NMI is used to force a non-maskable interrupt to the processor. The ICH2 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.</p> <p><b>Speed Strap:</b> During the reset sequence, ICH2 drives NMI high if the corresponding bit is set in the FREQ_STRP register.</p>
SMI#	O	<p><b>System Management Interrupt:</b> SMI# is an active low output synchronous to PCICLK. It is asserted by the ICH2 in response to one of many enabled hardware or software events.</p>
STPCLK#	O	<p><b>Stop Clock Request:</b> STPCLK# is an active low output synchronous to PCICLK. It is asserted by the ICH2 in response to one of many hardware or software events. When the processor samples STPCLK# asserted, it responds by stopping its internal clock.</p>

Table 2-11. Processor Interface Signals (Continued)

Name	Type	Description
RCIN#	I	<b>Keyboard Controller Reset Processor:</b> The keyboard controller can generate INIT# to the processor. This saves the external OR gate with the ICH2's other sources of INIT#. When the ICH2 detects the assertion of this signal, INIT# is generated for 16 PCI clocks. Note that the ICH2 ignores RCIN# assertion during transitions to the S3, S4 and S5 states.
A20GATE	I	<b>A20 Gate:</b> This signal is from the keyboard controller. It acts as an alternative method to force the A20M# signal active. A20GATE saves the external OR gate needed with various other PCIsets.
CPUPWRGD	OD	<b>Processor Power Good:</b> This signal should be connected to the processor's PWRGD input. This is an open-drain output signal (external pull-up resistor required) that represents a logical AND of the ICH2's PWROK and VRMPWRGD signals.

## 2.12 SMBus Interface

Table 2-12. SM Bus Interface Signals

Name	Type	Description
SMBDATA	I/OD	<b>SMBus Data:</b> External pull-up is required.
SMBCLK	I/OD	<b>SMBus Clock:</b> External pull-up is required.
SMBALERT#/GPIO[11]	I	<b>SMBus Alert:</b> This signal is used to wake the system or generate an SMI#. If not used for SMBALERT#, it can be used as a GPI.

## 2.13 System Management Interface

Table 2-13. System Management Interface Signals

Name	Type	Description
INTRUDER#	I	<b>Intruder Detect:</b> 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	<b>System Management Link:</b> These signals are an SMBus link to an optional external system management ASIC or LAN controller. External pull-ups are required. Note that SMLINK[0] corresponds to an SMBus Clock signal and SMLINK[1] corresponds to an SMBus Data signal.

## 2.14 Real Time Clock Interface

Table 2-14. Real Time Clock Interface

Name	Type	Description
RTCX1	Special	<b>Crystal Input 1:</b> This signal is connected to the 32.768 KHz crystal. If no external crystal is used, then RTCX1 can be driven with the desired clock rate.
RTCX2	Special	<b>Crystal Input 2:</b> This signal is connected to the 32.768 KHz crystal. If no external crystal is used, then RTCX2 should be left floating.

## 2.15 Other Clocks

Table 2-15. Other Clocks

Name	Type	Description
CLK14	I	<b>Oscillator Clock:</b> This signal is used for 8254 timers. Runs at 14.31818 MHz. This clock is permitted to stop during S3 (or lower) states.
CLK48	I	<b>48 MHz Clock:</b> This signal is used to run the USB controller. Runs at 48 MHz. This clock is permitted to stop during S3 (or lower) states.
CLK66	I	<b>66 MHz Clock:</b> This signal is used to run the hub interface. Runs at 66 MHz. This clock is permitted to stop during S3 (or lower) states.

## 2.16 Miscellaneous Signals

Table 2-16. Miscellaneous Signals

Name	Type	Description
SPKR	O	<b>Speaker:</b> 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 1.  Note: SPKR is sampled at the rising edge of PWROK as a functional strap. See <a href="#">Section 2.20.1</a> for more details.
RTCRST#	I	<b>RTC Reset:</b> When asserted, this signal resets register bits in the RTC well and sets the RTC_PWR_STS bit (bit 2 in GEN_PMCN3 register). This signal is also used to enter the test modes documented in <a href="#">Section 2.20.2</a> .  Note: Clearing CMOS in an ICH2-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.
TP[0]	I	<b>Test Point:</b> This signal must have an external pull-up to VccSus3_3.
FS[0]	I	<b>Functional Strap:</b> This signal is reserved for future use. There is an internal pull-up resistor on this signal.

## 2.17 AC'97 Link

Table 2-17. AC'97 Link Signals

Name	Type	Description
AC_RST#	O	<b>AC97 Reset:</b> Master H/W reset to external Codec(s)
AC_SYNC	O	<b>AC97 Sync:</b> 48 KHz fixed rate sample sync to the Codec(s)
AC_BIT_CLK	I	<b>AC97 Bit Clock:</b> 12.288 MHz serial data clock generated by the external Codec(s). See Note.
AC_SDOUT	O	<b>AC97 Serial Data Out:</b> Serial TDM data output to the Codec(s) Note: AC_SDOUT is sampled at the rising edge of PWROK as a functional strap. See <a href="#">Section 2.20.1</a> for more details.
AC_SDIN[1:0]	I	<b>AC97 Serial Data In 0:</b> Serial TDM data inputs from the Codecs. See Note.

**NOTE:** If the ACLINK Shutoff bit in the AC'97 Global Control Register (See [Section 13.2.8](#)) is set to 1, internal pull-down resistors will be enabled on AC\_BIT\_CLK and AC\_SDATA\_IN[1:0]. If ACLINK Shutoff is cleared to 0, these pull-down resistors are disabled. If there is no codec down on the system board, the two signals AC\_SDIN[1:0] should be pulled down externally with a resistor to ground.

## 2.18 General Purpose I/O

Table 2-18. General Purpose I/O Signals

Name	Type	Description
GPIO[31:29]	O	Not implemented.
GPIO[28:27]	I/O	Can be input or output. Resume power well. Unmuxed.
GPIO[26]	I/O	Not implemented.
GPIO[25]	I/O	Can be input or output. Resume power well. Not Muxed.
GPIO[24]	I/O	Can be input or output. Resume power well.
GPIO[23]	O	Fixed as Output only. Main power well.
GPIO[22]	OD	Fixed as Output only. Main power well. Open-drain output.
GPIO[21]	O	Fixed as Output only. Main power well.
GPIO[20:18]	O	Fixed as Output only. Main power well.
GPIO[17:16]	O	Fixed as Output only. Main Power Well. Can instead be used for PC/PCI GNT[A:B]#. GPIO[17] can also alternatively be used for PCI GNT[5]#. Integrated pull-up resistor.
GPIO[15:14]	I	Not implemented.
GPIO[13:12]	I	Fixed as Input only. Resume Power Well. Not muxed.
GPIO[11]	I	Fixed as Input only. Resume Power Well. Can instead be used for SMBALERT#.
GPIO[10:9]	I	Not implemented.
GPIO[8]	I	Fixed as Input only. Resume Power Well. Not muxed.
GPIO[7]	I	Fixed as Input only. Main power well. Not muxed.
GPIO[6]	I	Fixed as Input only. Main power well.
GPIO[5]		Not implemented.
GPIO[4:3]	I	Fixed as Input only. Main power well. Can be used instead as PIRQ[G:F]#.
GPIO[2]		Not implemented.
GPIO[1:0]	I	Fixed as Input only. Main Power Well. Can instead be used for PC/PCI REQ[A:B]#. GPIO[1] can also alternatively be used for PCI REQ[5]#.

## 2.19 Power and Ground

Table 2-19. Power and Ground Signals

Name	Description
<b>Vcc3_3</b>	3.3V supply for Core well I/O buffers. This power may be shut off in S3, S5 or G3 states.
<b>Vcc1_8</b>	1.8V supply for Core well logic. This power may be shut off in S3, S5 or G3 states.
<b>V5REF</b>	Reference for 5V tolerance on Core well inputs. This power may be shut off in S3, S5 or G3 states.
<b>HUBREF</b>	0.9V reference for the hub interface. This power may be shut off in S3, S5 or G3 states.
<b>VccSus3_3</b>	3.3V supply for Resume well I/O buffers. This power is not expected to be shut off unless the system is unplugged.
<b>VccSus1_8</b>	1.8V supply for Resume well logic. This power is not expected to be shut off unless the system is unplugged.
<b>V5REF_SUS</b>	Reference for 5V tolerance on Resume well inputs. This power is not expected to be shut off unless the system is unplugged. Note: V5REF_SUS only affects 5V tolerance for the USB OC[3:0]# pins and can be connected to VccSUS3_3 if 5V tolerance on these signals is not required.
<b>VccRTC</b>	3.3V (can drop to 2.0V min. in G3 state) supply for the RTC well. This power is not expected to be shut off unless the RTC battery is removed or completely drained. Note: Implementations should not attempt to clear CMOS by using a jumper to pull VccRTC low. Clearing CMOS in an ICH2-based platform can be done by using a jumper on RTCRST# or GPI, or using SAFEMODE strap.
<b>VBIAS</b>	RTC well bias voltage. The DC reference voltage applied to this pin sets a current that is mirrored throughout the oscillator and buffer circuitry. See <a href="#">Section 2.20.3</a> .
<b>V_CPU_IO</b>	Powered by the same supply as the processor I/O voltage. This supply is used to drive the processor interface outputs.
<b>Vss</b>	Grounds.

## 2.20 Pin Straps

### 2.20.1 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 4 PCI clocks prior to the time it is sampled.

Table 2-20. 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 ICH2 sets the processor speed strap pins for safe mode. Refer to processor specification for speed strapping definition. The status of this strap is readable via the SAFE_MODE bit (bit 2, D31: F0, Offset D4h).
EE_DOUT	Reserved		System designers should include a placeholder for a pull-down resistor on EE_DOUT but <i>do not populate the resistor</i> .

**Table 2-20. Functional Strap Definitions (Continued)**

Signal	Usage	When Sampled	Comment
FS[0]	Reserved		System designers should include a placeholder for a pull-down resistor on FS[0] but <i>do not populate the resistor</i> .
GNT[A]#	Top-Swap Override	Rising Edge of PWROK	The signal has a weak internal pull-up. If the signal is sampled low, the system is strapped to the “Top-Swap” mode (ICH2 will invert 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 GNT[A]# being pulled down.
HLCOMP	Enhanced Hub Interface Mode	During PCIRST# assertion	If this signal is sampled high (via an external pull-up to VCC1_8), the normal hub interface buffer mode will be selected. If this signal is sampled low (via an external pull-down), the enhanced hub interface buffer mode will be selected. See the specific platform design guide for resistor values and routing guidelines for each hub interface mode.
SPKR	No Reboot	Rising Edge of PWROK	The signal has a weak internal pull-up. If the signal is sampled low, the system is strapped to the “No Reboot” mode (ICH2 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).

## 2.20.2 Test Signals

### 2.20.2.1 Test Mode Selection

When PWROK is active (high), driving RTCRST# low for a number of PCI clocks (33 MHz) will activate a particular test mode as specified in [Table 2-21](#).

**Note:** RTCRST# may be driven low any time after PCIRST is inactive. Refer to [Chapter 17, “Testability”](#) for a detailed description of the ICH2 test modes.

**Table 2-21. 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”
9–24	Reserved. DO NOT ATTEMPT
>24	No Test Mode Selected

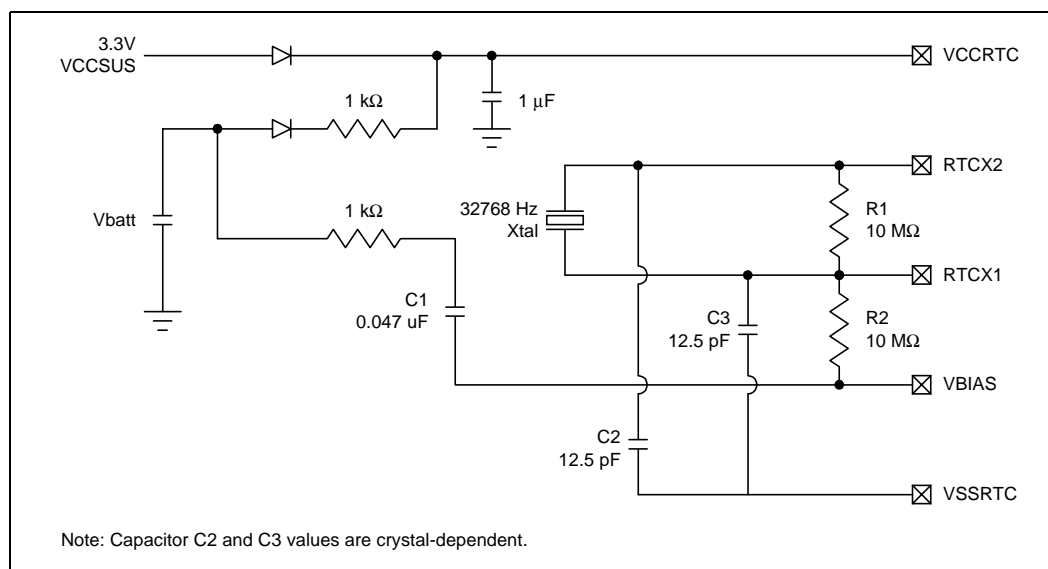
### 2.20.2.2 Test Straps

The ICH2’s TP[0] (Test Point) signal must be pulled to VccSus3\_3 with an external pull-up resistor.

### 2.20.3 External RTC Circuitry

To reduce RTC well power consumption, the ICH2 implements an internal oscillator circuit that is sensitive to step voltage changes in VccRTC and VBIAS. Figure 2-1 shows a schematic diagram of the circuitry required to condition these voltages to ensure correct operation of the ICH2 RTC.

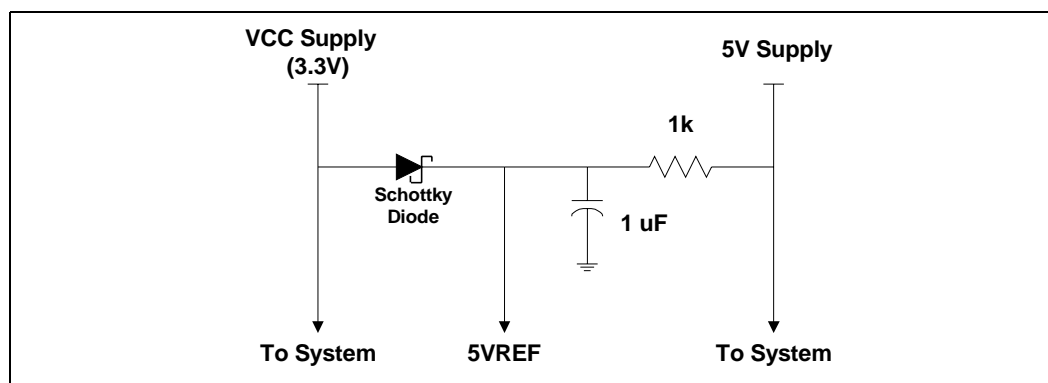
Figure 2-1. Required External RTC Circuit



### 2.20.4 V5REF / Vcc3\_3 Sequencing Requirements

V5REF and V5REF\_Sus are the reference voltages for 5V tolerance on inputs to the ICH2. V5REF and V5REF\_Sus must power up before or simultaneous to Vcc3\_3 and VccSus3\_3 respectively, and must power down after or simultaneous to Vcc3\_3 and VccSus3\_3 respectively. Refer to Figure 2-2 for an example circuit schematic that may be used to ensure proper V5REF sequencing. Note that separate circuits must be implemented for both the Core and Suspend well supplies.

Figure 2-2. Example V5REF Sequencing Circuit



# ICH2 Power Planes and Pin States 3

## 3.1 Power Planes

Table 3-1. ICH2 Power Planes

Plane	Description
Main I/O (3.3V)	<b>Vcc3_3:</b> 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 (1.8V)	<b>Vcc1_8:</b> 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 (3.3V Standby)	<b>VccSUS3_3:</b> Powered by the main power supply in S0 - S1 states. Powered by the trickle power supply when the system is in the S3, S4, S5, state. Assumed to be shut off only when in the G3 state (system is unplugged).
Resume Logic (1.8V Standby)	<b>VccSUS1_8:</b> Powered by the main power supply in S0 - S1 states. Powered by the trickle power supply when the system is in the S3, S4, S5, state. Assumed to be shut off only when in the G3 state (system is unplugged).
Processor Interface (1.3 ~ 2.5V)	<b>V_CPU_IO:</b> 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	<b>VccRTC:</b> When other power is available (from the main supply), external diode coupling will provide power to reduce the drain on the RTC battery. Assumed to operate from 3.3V down to 2.0V.

## 3.2 Integrated Pull-Ups and Pull-Downs

Table 3-2. Integrated Pull-Up and Pull-Down Resistors

Signal	Resistor Type	Nominal Value	Notes
EE_DIN	pull-up	24 K $\Omega$	1
EE_DOUT	pull-up	24 K $\Omega$	1
GNT[B:A]# / GNT[5]# / GPIO[17:16]	pull-up	24 K $\Omega$	1
LAD[3:0]# / FWH[3:0]#	pull-up	24 K $\Omega$	1
LDRQ[1:0]	pull-up	24 K $\Omega$	1
PME#	pull-up	24 K $\Omega$	1
PWRBTN#	pull-up	24 K $\Omega$	1
SPKR	pull-up	24 K $\Omega$	1, 5
AC_BITCLK	pull-down	20 K $\Omega$	2, 6
AC_SDIN[0]	pull-down	20 K $\Omega$	2, 6
AC_SDIN[1]	pull-down	20 K $\Omega$	2, 6
AC_SDOUT	pull-down	20 K $\Omega$	2, 6
AC_SYNC	pull-down	20 K $\Omega$	2, 6
LAN_RXD[2:0]	pull-down	20 K $\Omega$	2, 6

**Table 3-2. Integrated Pull-Up and Pull-Down Resistors (Continued)**

Signal	Resistor Type	Nominal Value	Notes
LAN_RXD[2:0]	pull-up	9 K $\Omega$	3
PDD[7] / SDD[7]	pull-down	5.9 K $\Omega$	4
PDDREQ / SDDREQ	pull-down	5.9 K $\Omega$	4

**NOTES:**

1. Simulation data shows that these resistor values can range from 18 K $\Omega$  to 42 K $\Omega$ .
2. Simulation data shows that these resistor values can range from 13 K $\Omega$  to 38 K $\Omega$ .
3. Simulation data shows that these resistor values can range from 6 K $\Omega$  to 14 K $\Omega$ .
4. Simulation data shows that these resistor values can range from 4.3 K $\Omega$  to 20 K $\Omega$ .
5. The pull-up or pull-down on this signal is only enabled at boot/reset for strapping function.
6. This pull-down is only enabled when the ACLINK Shut Off bit in the AC'97 Global Control Register is set to 1.

### 3.3 IDE Integrated Series Termination Resistors

Table 3-3 shows the ICH2 IDE signals that have integrated series termination resistors.

**Table 3-3. IDE Series Termination Resistors**

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

**NOTE:** Simulation data indicates that the integrated series termination resistors are a nominal 33  $\Omega$  but can range from 31  $\Omega$  to 43  $\Omega$ .

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

Table 3-4 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. ICH2 not driving the signal high or low.
“High”	ICH2 is driving the signal to a logic ‘1’
“Low”	ICH2 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”	ICH2 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 ICH2 is not driving

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

**Table 3-4. Power Plane and States for Output and I/O Signals**

Signal Name	Power Plane	During PCIRST#	Immediately after PCIRST#	S1	S3	S4/S5
<b>PCI Bus</b>						
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[0:5]#	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
<b>LPC Interface</b>						
LAD[3:0]	Main I/O	High	High	High	Off	Off
LFRAME#	Main I/O	High	High	High	Off	Off
<b>LAN Connect and EEPROM Interface</b>						
EE_CS	Resume I/O	High	Running	Defined	Defined	Defined
EE_DOUT	Resume I/O	High	Running	Defined	Defined	Defined
EE_SHCLK	Resume I/O		Running	Defined	Defined	Defined
LAN_RSTSYNC	Resume I/O	High	Defined	Defined	Defined	Defined
LAN_TXD[2:0]	Resume I/O	Low	Defined	Defined	Defined	Defined
<b>IDE Interface</b>						
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[15:0], SDD[15:0]	Main I/O	Undefined	Undefined	Undefined	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
<b>Interrupts</b>						
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
APICD[1:0]	Main I/O	High-Z	High-Z	High-Z	Off	Off

Table 3-4. Power Plane and States for Output and I/O Signals (Continued)

Signal Name	Power Plane	During PCIRST#	Immediately after PCIRST#	S1	S3	S4/S5
USB Interface						
USBP[3:0][P:N]	Resume I/O	High-Z	High-Z	High-Z	High-Z	High-Z
Power Management						
SLP_S3#	Resume I/O	High	High	High	Low	Low
SLP_S5#	Resume I/O	High	High	High	High	Low
SUS_STAT#	Resume I/O	High	High	High	Low	Low
SUSCLK	Resume I/O	Running				
Processor Interface						
A20M#	Processor I/O	See Note 1	High	High	Off	Off
CPUPWRGD	Main I/O	See Note 3	High-Z	High-Z	Off	Off
CPUSLP#	Processor I/O	High	High	Defined	Off	Off
IGNNE#	Processor I/O	See Note 1	High	High	Off	Off
INIT#	Processor I/O	High	High	High	Off	Off
INTR	Processor I/O	See Note 1	Low	Low	Off	Off
NMI	Processor I/O	See Note 1	Low	Low	Off	Off
SMI#	Processor I/O	High	High	High	Off	Off
STPCLK#	Processor I/O	High	High	Low	Off	Off
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
Miscellaneous Signals						
SPKR	Main I/O	High-Z with internal pull-up	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
Unmuxed GPIO Signals						
GPIO[18]	Main I/O	High	See Note 2	Defined	Off	Off
GPIO[19:20]	Main I/O	High	High	Defined	Off	Off
GPIO[21]	Main I/O	High	High	Defined	Off	Off
GPIO[22]	Main I/O	High-Z	High-Z	Defined	Off	Off

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

Signal Name	Power Plane	During PCIRST#	Immediately after PCIRST#	S1	S3	S4/S5
GPIO[23]	Main I/O	Low	Low	Defined	Off	Off
GPIO[24]	Resume I/O	Low	Low	Defined	Defined	Defined
GPIO[25]	Resume I/O	High	High	Defined	Defined	Defined
GPIO[27:28]	Resume I/O	High-Z	High-Z	Defined	Defined	Defined

**NOTES:**

1. ICH2 sets these signals at reset for processor frequency strap.
2. GPIO[18] will toggle at a frequency of approximately 1 Hz when the ICH2 comes out of reset
3. CPUPWRGD is an open-drain output that represents a logical AND of the ICH2's VRMPWRGD and PWROK signals, and thus will be driven low by ICH2 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.

## 3.5 Power Planes for Input Signals

Table 3-5 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 3-5. Power Plane for Input Signals**

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[1:0]	Resume I/O	AC'97 Codec	Low	Low	Low
APICCLK	Main I/O	Clock Generator	Running	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
EE_DIN	Resume I/O	EEPROM component	Driven	Driven	Driven
FERR#	Main I/O	Processor	Static	Low	Low
INTRUDER#	RTC	External Switch	Driven	Driven	Driven
IRQ[15:14]	Main I/O	IDE	Static	Low	Low
LAN_CLK	Resume I/O	LAN Connect component	Driven	Driven	Driven
LAN_RXD[2:0]	Resume I/O	LAN Connect component	Driven	Driven	Driven
LDRQ[0]#	Main I/O	LPC Devices	High	Low	Low
LDRQ[1]#	Main I/O	LPC Devices	High	Low	Low

Table 3-5. Power Plane for Input Signals (Continued)

Signal Name	Power Well	Driver During Reset	S1	S3	S5
OC[3:0]#	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
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[0:5]#	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
RSM_PWROK	Resume I/O	External RC circuit	High	High	High
RSMRST#	RTC	External RC circuit	High	High	High
RTCRST#	RTC	External RC circuit	High	High	High
SDDREQ	Main I/O	IDE Drive	Static	Low	Low
SERR#	Main I/O	PCI Bus Peripherals	High	Low	Low
SIORDY	Main I/O	IDE Drive	Static	Low	Low
SMBALERT#	Resume I/O	External pull-up	Driven	Driven	Driven
THRM#	Main I/O	Thermal Sensor	Driven	Low	Low
VRMPWRGD	Main I/O	Processor Voltage Regulator	High	Low	Low

# ICH2 and System Clock Domains

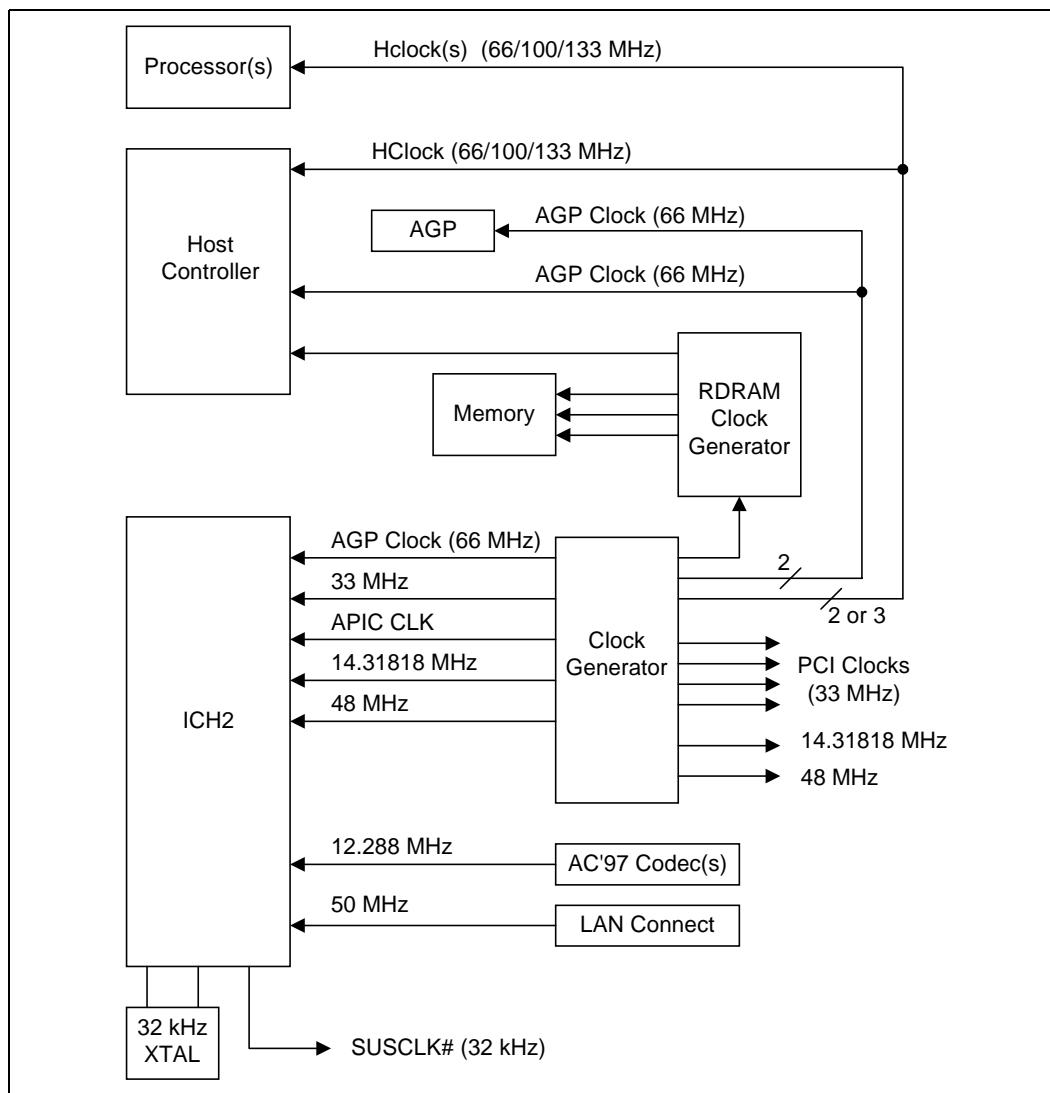
## 4

Table 4-1 shows the system clock domains. Figure 4-2 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.

**Figure 4-1. ICH2 and System Clock Domains**

Clock Domain	Frequency	Source	Usage
ICH2 CLK66	66 MHz	Main Clock Generator	Hub interface, processor interface. AGP. Shut off during S3 or below.
ICH2 PCICLK	33 MHz	Main Clock Generator	Free-running PCI Clock to ICH2. 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.
ICH2 CLK48	48 MHz	Main Clock Generator	Super I/O, USB Controller. Expected to be shut off during S3 or below.
ICH2 CLK14	14.31818 MHz	Main Clock Generator	Used for ACPI timer. Expected to be shut off during S3 or below.
ICH2 AC_BIT_CLK	12.288 MHz	AC'97 Codec	AC'97 Link. Generated by AC'97 CODEC. Can be shut off by codec in D3. Expected to be shut off during S3 or below.
RTC	32.768 kHz	ICH2	RTC, Power Management. ICH2 has its own oscillator. Always running, even in G3 state.
ICH2 APICCLK	33.33 MHz	Main Clock Generator	Used for ICH2 processor interrupt messages. Runs at 33.33 MHz. Expected to be shut off during S3 or below.
LAN_CLK	0.8 to 50 MHz	LAN Connect Component	Generated by the LAN Connect component. Expected to be shut off during S3 or below.

Figure 4-2. Conceptual System Clock Diagram



# Functional Description

# 5

## 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 ICH2 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.

### 5.1.1 PCI Bus Interface

The ICH2 PCI interface provides a 33 MHz, Rev. 2.2 compliant implementation. All PCI signals are 5V tolerant. The ICH2 integrates a PCI arbiter that supports up to six external PCI bus masters in addition to the internal ICH2 requests.

Note that most transactions targeted to the ICH2 will first appear on the external PCI bus before being claimed back by the ICH2. The exceptions are I/O cycles involving USB, IDE, and AC'97. These transactions will complete over the hub interface without appearing on the external PCI bus. *Configuration cycles* targeting USB, IDE or AC'97 will appear on the PCI bus. If the ICH2 is programmed for positive decode, the ICH2 will claim the cycles appearing on the external PCI bus in medium decode time. If the ICH2 is programmed for subtractive decode, the ICH2 will claim these cycles in subtractive time. If the ICH2 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. To boot off a PCI card it is necessary to keep the ICH2 in subtractive decode mode. When booting off a PCI card, the BOOT\_STS bit (bit 2, TCO2 Status Register) will be set.

**Note:** The ICH2's AC'97, IDE and USB Controllers can not access PCI address ranges.

**Note:** PCI devices that cause long latencies (numerous retries) to processor-to-PCI Locked cycles may starve isochronous transfers between USB or AC'97 devices and memory. This will result in overrun or underrun, causing reduced quality of the isochronous data (e.g., audio).

**Note:** PCI configuration write cycles, initiated by the processor, with the following characteristics will be 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

## 5.1.2 PCI-to-PCI Bridge Model

From a software perspective, the ICH2 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 ICH2 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 ICH2 asserts one address signal as an IDSEL. When accessing device 0, the ICH2 asserts AD16. When accessing Device 1, the ICH2 asserts AD17. This mapping continues up to device 15 where the ICH2 asserts AD31. Note that the ICH2's internal functions (AC'97, IDE, USB, 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 ICH2's PCI bus and, hence, uses AD24 for IDSEL.

## 5.1.4 SERR# Functionality

There are several internal and external sources that can cause SERR#. The ICH2 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 ICH2 does not drive the external PCI bus SERR# signal active onto the PCI bus. The external SERR# signal is an input into the ICH2 driven only by external PCI devices. The conceptual logic diagrams in Figure 5-1 and Figure 5-2 illustrate all sources of SERR#, along with their respective enable and status bits. Figure 5-3 shows how the ICH2 error reporting logic is configured for NMI# generation.

Figure 5-1. Primary Device Status Register Error Reporting Logic

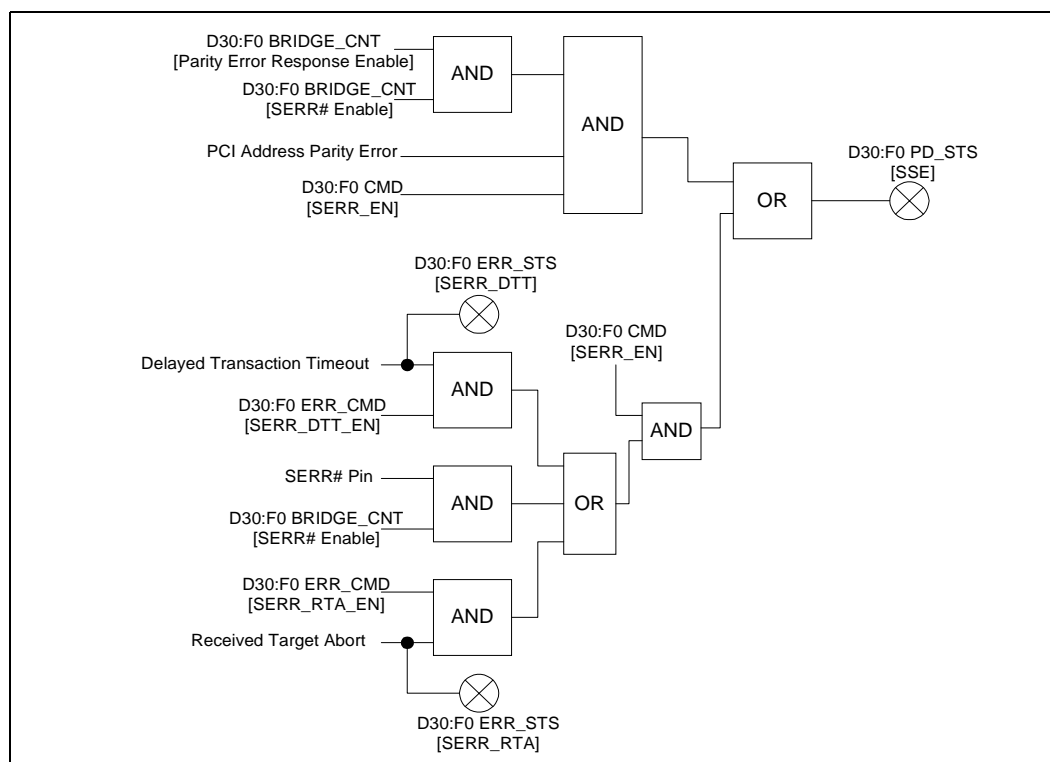


Figure 5-2. Secondary Status Register Error Reporting Logic

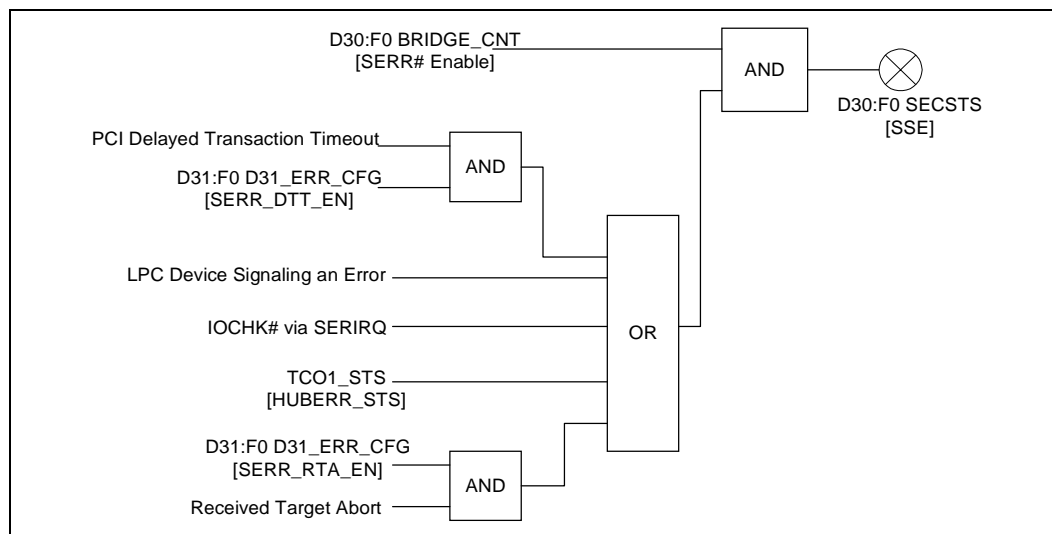
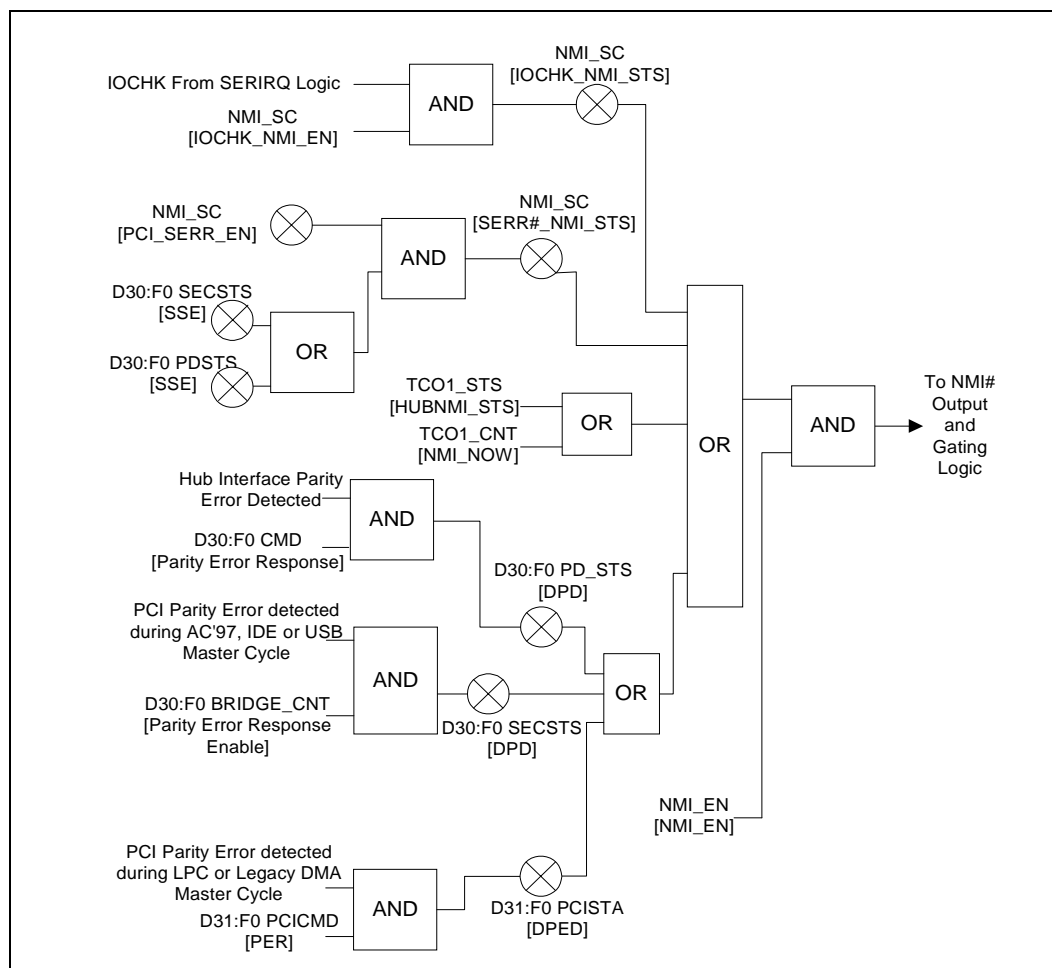


Figure 5-3. NMI# Generation Logic



## 5.1.5 Parity Error Detection

The ICH2 can detect and report different parity errors in the system. The ICH2 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 5-3](#) details all the parity errors that the ICH2 can detect, along with their respective enable bits, status bits, and the results.

**Note:** If NMIs are enabled and parity error checking on PCI is also enabled, then parity errors 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 8 functions with each function containing up to 256 8-bit configuration registers. The PCI specification 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 ICH2. The PCI specification defines two mechanisms to access configuration space (Mechanism #1 and Mechanism #2). The ICH2 only supports Mechanism #1.

Configuration cycles for PCI Bus #0 devices #2 through #31, and for PCI Bus numbers greater than 0 will be sent towards the ICH2 from the host controller. The ICH2 compares the non-zero Bus Number with the Secondary Bus Number and Subordinate Bus number registers of its P2P bridge to determine if the configuration cycle is meant for Primary PCI or a downstream PCI bus.

### Type 0 to Type 0 Forwarding

When a Type 0 configuration cycle is received on the hub interface, the ICH2 forwards these cycles to PCI and then reclaims them. The ICH2 uses address bits AD[15:14] to communicate the ICH2 device numbers in Type 0 configuration cycles. If the Type 0 cycle on the hub interface specifies any device number other than 30 or 31, the ICH2 will not set any address bits in the range AD[31:11] during the corresponding transaction on PCI. [Table 5-1](#) shows the device number translation.

**Table 5-1. 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 29	0000000000000000_00000b
30	0000000000000000_01000b
31	0000000000000000_10000b

The ICH2 logic generates single DWord configuration read and write cycles on the PCI bus. The ICH2 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 ICH2 runs a Type 1 cycle on the PCI bus.

## Type 1 to Type 0 Conversion

When the bus number for the Type 1 configuration cycle matches the PCI (Secondary) bus number, the ICH2 converts the address as follows:

- For device numbers 0 through 15, only one bit of the PCI address [31:16] is set. If the device number is 0, AD[16] is set; if the device number is 1, AD[17] is set; etc.
- The ICH2 always drives 0s on bits AD[15:11] when converting Type 1 configurations cycles to Type 0 configuration cycles on PCI.
- Address bits [10:1] are also passed unchanged to PCI.
- Address bit [0] is changed to 0.

## 5.1.7 PCI Dual Address Cycle (DAC) Support

The ICH2 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 will be determined by the memory controller and the processor.

The DAC mode is only supported for PCI adapters and is not supported for any of the internal PCI masters (IDE, LAN, USB, AC'97, 8237 DMA, etc.). ICH2 does not support DAC for processor-initiated cycles.

When a PCI master wants to initiate a cycle with an address above 4 GB, it follows the following behavioral rules (See PCI 2.2 Specification, 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 ICH2 will ignore 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 ICH2'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, which 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 ICH2 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 ICH2's virtual PCI-to-PCI Bridge (see [Section 5.1.2](#)). This is typically Bus 1; it may be assigned a different number depending on system configuration.

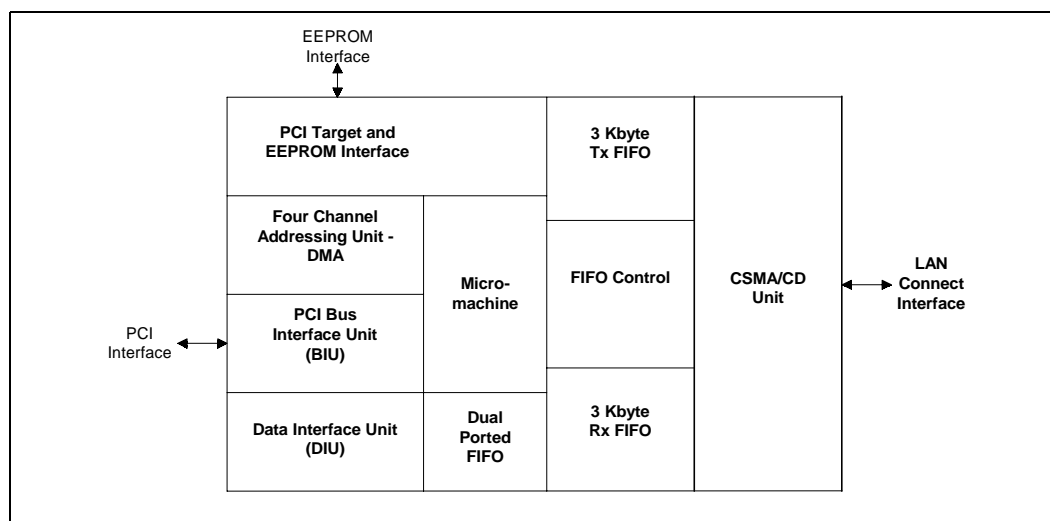
### Feature Summary

- 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) Rev 2.0
- Backward compatible software with 82557, 82558 and 82559
- TCP/UDP checksum offload capabilities
- Support for Intel's Adaptive Technology

## 5.2.1 LAN Controller Architectural Overview

Figure 5-4 is a high level block diagram of the ICH2 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 5-4. Integrated LAN Controller Block Diagram



### Parallel Subsystem Overview

The parallel subsystem is divided 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 (e.g., 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 Bus Specification, Revision 2.2. 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 (e.g., 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 that 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](#).

## FIFO Subsystem Overview

The ICH2 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 ICH2 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.

## Serial CSMA/CD Unit Overview

The CSMA/CD unit of the ICH2 LAN Controller allows it to be connected to the 82562ET/EM 10/100 Mbps Ethernet LAN Connect components or the 82562EH 1 Mbps HomePNA\*-compliant LAN Connect component. 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 which allows simultaneous transmission and reception of frames.

## 5.2.2 LAN Controller PCI Bus Interface

As a Fast Ethernet Controller, the role of the ICH2 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 ICH2 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
- PCI Configuration cycles

The size of the CSR memory space is 4 KB 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 4 KB of CSR space, which includes 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 Control/Status Registers, 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 to complete any delayed transaction. Any master access to the LAN Controller after the completion of the EEPROM read will be 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 terminates 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, will claim the cycle if it was the target of the transaction and continue 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 ICH2 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 the 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 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, 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 a 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 signal's 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 channel) 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:

1. Minimum transfer of one cache line
2. Active byte enable bits (or BE[3:0]# are all low) during MWI access
3. 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:

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

If any one of the above conditions does not hold, the LAN Controller will use 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 will be 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 (i.e., 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:** This feature is not recommended for use in non-cache line oriented systems since it may cause shorter bursts and lower performance.

**Note:** This feature should be used only when the CLS register in PCI Configuration space is set to 8 or 16.

**Note:** The LAN Controller reads all control data structures (including Receive Buffer Descriptors) from the first DWord (even if it is not required) 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 specification rev. 2.2, is provided in the ICH2 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 will wake 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 all four power states defined in the Power Management Network Device Class Reference Specification, Rev. 1.0. The four states, D0 through D3, 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 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.** As defined in the Network Device Class Reference Specification, 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 (PCI Configuration space) switches the LAN Controller from D0u state to 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.** For a device to meet the D1 power state requirements, as specified in the Advanced Configuration and Power Interface (ACPI) Specification, Revision 1.0, 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 such as 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 the PME\_EN 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 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 (if PME\_EN=0).

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 ICH2 resume well and, thus, is connected to an auxiliary power source (a separate LAN well). This 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 WOL 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 WOL 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 will classify the incoming packets as one of the following categories:

- **No Match:** The LAN Controller discards the packet and continues to process the incoming packets.
- **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 WOL 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 WOL mode after reset if the WOL bit in the EEPROM is set. At this point, the LAN Controller is in the D0u state.

When the LAN Controller is in WOL mode:

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

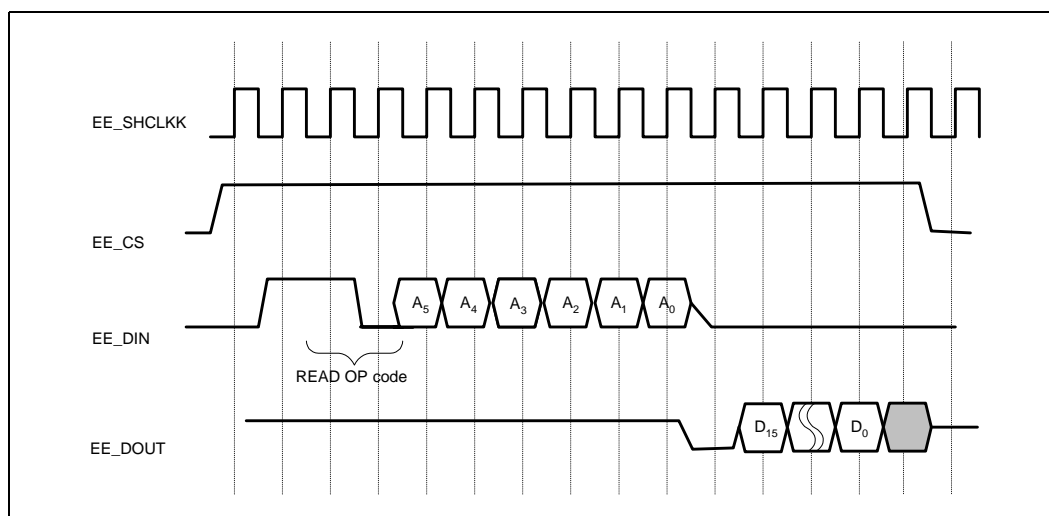
The LAN Controller switches from WOL mode to the D0a power state following a setup of the Memory or I/O Base Address Registers in the PCI configuration space.

### 5.2.3 Serial EEPROM Interface

The serial EEPROM stores configuration data for the ICH2 integrated LAN Controller and is a serial in/serial out device. The LAN Controller supports a 64 word size or 256 word size EEPROM and automatically detects the EEPROM's size. A 256 word EEPROM device is required for a Cardbus system and contains the CIS information. 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 word EEPROM or eight bits for a 256 word EEPROM. The end of the address field is indicated by a dummy zero 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 5-5.

**Figure 5-5. 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. The ICH2 integrated LAN Controller's EEPROM format is shown below in Table 5-2.

**Table 5-2. EEPROM Format**

Word	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0h	IA Byte 2								IA Byte 1							
1h	IA Byte 4								IA Byte 3							
2h	IA Byte 6								IA Byte 5							
Ah	Sig		ID	00b		Rev ID			1b	DPD	WOL	00b		Rsvd	STB Ena	0b
Bh	Subsystem ID															
Ch	Subsystem Vendor ID															
Dh	Reserved															

Note that word Ah contains several configuration bits. These bits are described in Table 5-3.

**Table 5-3. EEPROM Word Field Descriptions**

Bits	Name	Description
Word Ah 15:14	Signature	A signature of 01b indicates to the LAN Controller that there is a valid EEPROM present. If this field is not equal to 01b, the other bits are ignored and the default values are used.
Word Ah 13	ID	The ID bit indicates how the Subsystem ID and Subsystem Vendor ID fields are used as described in <a href="#">Section 7.1.14</a> . Default value is 0b.
Word Ah 12:11	Reserved	Should be set to 0b.
Word Ah 10:8	Revision ID	These three bits are used as the three least significant bits of the device revision, if bits [15, 14, and 13] = 011b and the ID was set as described in <a href="#">Section 7.1.14</a> . The default value depends on the silicon revision.
Word Ah 7	Power Management	Enables the LAN Controller's PCI power management capabilities. This bit is always set to 1.
Word Ah 6	Deep Power Down	Bit is used as the Deep Power Down enable/disable bit. When the DPD bit = 0, deep power down is enabled in the D3 power state while PME is disabled. If the DPD bit = 1, deep power down is disabled in the D3 power state while PME is disabled.
Word Ah 5	Wake on LAN	This bit sets the LAN Controller into WOL mode. When in this mode, the LAN Controller reads three additional words from the EEPROM, word addresses 0h, 1h, and 2h. These words are expected to hold the MAC Individual Address. After reading these words, the LAN Controller wakes the system by asserting PME# when a wake-up packet is received. Default value is 0b.
Word Ah 4:3	Reserved	Should be set to 00b.
Word Ah 2	Reserved	Should be set to 00b.
Word Ah 1	Standby Enable	Enables the LAN Controller to enter standby mode. When this bit = 1, the LAN Controller is able to recognize an idle state and can enter standby mode (some internal clocks are stopped for power saving purposes). If this bit = 0, the idle recognition circuit is disabled and the LAN Controller always remains in an active state.
Word Ah 0	Reserved	Should be set to 0.

## 5.2.4 CSMA/CD Unit

The ICH2 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 also supports the 1 Mbps Home Phoneline Networking Alliance (HomePNA\*) specification effort. It performs all the CSMA/CD protocol functions such as transmission, reception, collision handling, etc. The LAN Controller CSMA/CD unit interfaces to the 82562ET/EM 10/100 Mbps Ethernet or the 82562EH 1 Mbps HomePNA\*-compliant LAN Connect component through the ICH2's LAN Connect interface signals.

### 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 LAN Connect component detects a valid frame on its receive differential pair. When in Full Duplex mode, the ICH2 integrated LAN Controller also supports the IEEE 802.3x flow control standard.

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 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 LAN Connect component. Following reset, the CSMA will default to automatically track the 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.

### Flow Control

The LAN Controller supports IEEE 802.3x frame based flow control frames only in full 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.

### Address Filtering Modifications

The LAN Controller can be configured to ignore one 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.

### VLAN Support

The LAN Controller supports the IEEE 802.1 standard VLAN. All VLAN flows are 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 LAN Connect component via a control register in the ICH2 integrated LAN Controller. This allows the software driver to place the LAN Connect in specific modes such as 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 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 LAN Connect component, and are described in detail in the 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 LAN Connect component through the MDI register.

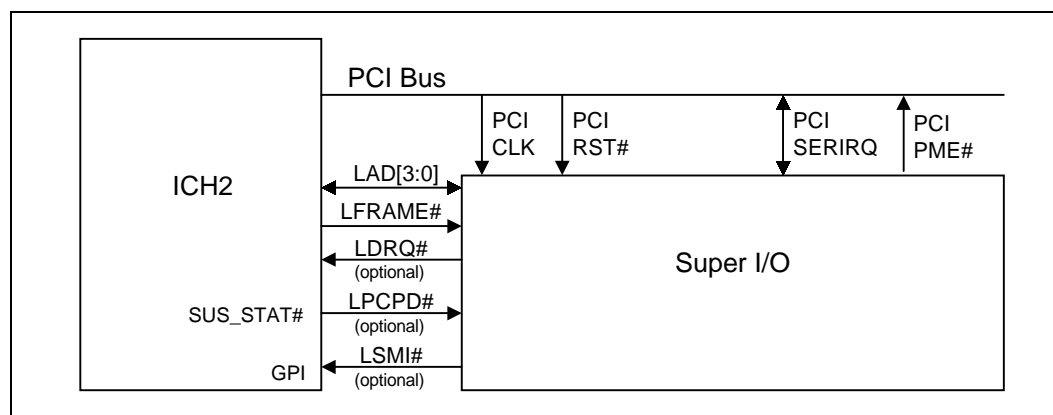
## 5.3 LPC Bridge (w/ System and Management Functions) (D31:F0)

The LPC Bridge function of the ICH2 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.

### 5.3.1 LPC Interface

The ICH2 implements an LPC interface as described in the LPC 1.0 specification. The LPC interface to the ICH2 is shown in Figure 5-6. Note that the ICH2 implements all of the signals that are shown as optional, but peripherals are not required to do so.

Figure 5-6. LPC Interface Diagram



### 5.3.1.1 LPC Cycle Types

The ICH2 implements all of the cycle types described in the LPC I/F 1.0 specification. [Table 5-4](#) shows the cycle types supported by the ICH2.

**Table 5-4. 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. ICH2 breaks up 16 and 32-bit processor cycles into multiple 8-bit transfers. See Note 1 below.
I/O Write	1 byte only. ICH2 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:**

- For memory cycles below 16 MB which do not target enabled FWH ranges, the ICH2 will perform standard LPC memory cycles. It will only attempt 8-bit transfers. If the cycle appears on PCI as a 16-bit transfer, it will appear as two consecutive 8-bit transfers on LPC. Likewise, if the cycle appears as a 32-bit transfer on PCI, it will appear as four consecutive 8-bit transfers on LPC. If the cycle is not claimed by any peripheral, it will be subsequently aborted, and the ICH2 will return 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.
- Bus Master Read or Write cycles must be naturally aligned. For example, a 1-byte transfer can be to any address. However, the 2-byte transfer must be word aligned (i.e. with an address where A0=0). A DWord transfer must be DWord aligned (i.e., with an address where A1 and A0 are both 0).

### 5.3.1.2 Start Field Definition

**Table 5-5. 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.

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

The ICH2 always drives bit 0 of this field to 0. Peripherals running bus master cycles must also drive bit 0 to 0. [Table 5-6](#) shows the valid bit encodings.

**Table 5-6. Cycle Type Bit Definitions**

Bits[3:2]	Bit[1]	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 ICH2 will abort the cycle.

### 5.3.1.4 Size

Bits[3:2] are reserved. The ICH2 always drives them to 00. Peripherals running bus master cycles are also supposed to drive 00 for bits 3:2; however, the ICH2 ignores those bits. [Table 5-7](#) shows the bit encodings for Bits[1:0].

**Table 5-7. Transfer Size Bit Definition**

Bits[1:0]	Size
00	8 bit transfer (1 byte)
01	16-bit transfer (2 bytes)
10	Reserved. The ICH2 never drives this combination. If a peripheral running a bus master cycle drives this combination, the ICH2 may abort the transfer.
11	32 bit transfer (4 bytes)

### 5.3.1.5 SYNC

Valid values for the SYNC field are listed in [Table 5-8](#).

**Table 5-8. SYNC Bit Definition**

Bits[3:0]	Indication
0000	<b>Ready:</b> SYNC achieved with no error. For DMA transfers, this also indicates DMA request deassertion and no more transfers desired for that channel.
0101	<b>Short Wait:</b> Part indicating wait states. For bus master cycles, the ICH2 does not use this encoding. It will instead use the Long Wait encoding (see next encoding below).
0110	<b>Long Wait:</b> Part indicating wait states; many wait states will be added. This encoding driven by the ICH2 for bus master cycles, rather than the Short Wait (0101).
1001	<b>Ready More (Used only by peripheral for DMA cycle):</b> 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	<b>Error:</b> Sync achieved with error. This is generally used to replace the SERR# or IOCHK# signal on the PCI/ISA bus. It indicates that the data is to be transferred, but there is a serious error in this transfer. 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.

### 5.3.1.6 SYNC Time-out

There are several error cases that can occur on the LPC interface. [Table 5-9](#) indicates the failing case and the ICH2 response.

**Table 5-9. ICH2 Response to Sync Failures**

Possible Sync Failure	ICH2 Response
ICH2 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.	ICH2 aborts the cycle after the 4 <sup>th</sup> clock.
ICH2 drives a Memory, I/O, or DMA cycle, and a peripheral drives more than 8 consecutive valid SYNC patterns to insert wait states using the Short ('0101b') encoding for SYNC. This could occur if the peripheral is not operating properly.	Continues waiting
ICH2 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 interface.	ICH2 aborts the cycle when the invalid Sync is recognized.

**NOTE:** There may be other peripheral failure conditions; however, these are not handled by the ICH2.

### 5.3.1.7 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 ICH2 was reading data from a peripheral, data will still be transferred in the next two nibbles. This data may be invalid; however, it must be transferred by the peripheral. If the ICH2 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 ICH2 is transferring 4 bytes from a device and the device returns the error SYNC in the first byte, the other three bytes are not transferred.

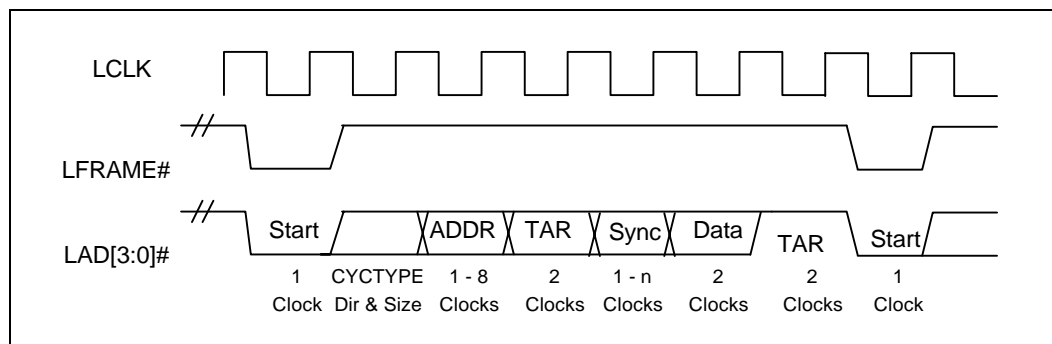
When recognizing the SYNC field indicating an error, the ICH2 treats this the same as IOCHK# going active on the ISA bus.

### 5.3.1.8 LFRAME# Usage

#### Start of Cycle

For Memory, I/O, and DMA cycles, the ICH2 asserts LFRAME# for 1 clock at the beginning of the cycle ([Figure 5-7](#)). During that clock, the ICH2 drives LAD[3:0] with the proper START field.

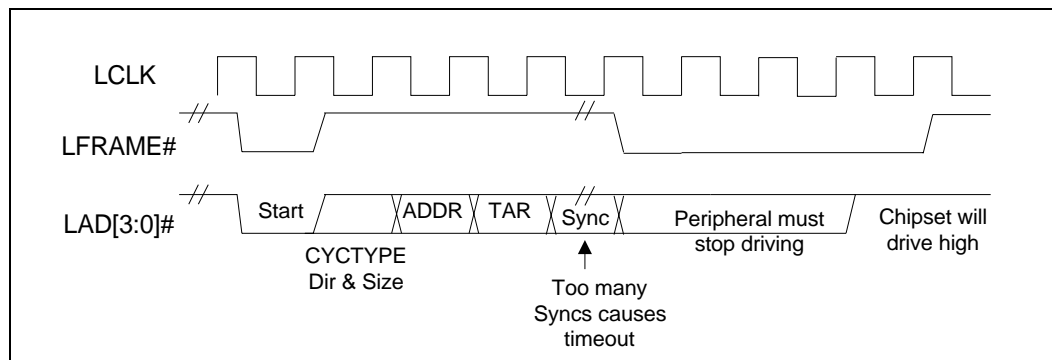
Figure 5-7. Typical Timing for LFRAME#



### Abort Mechanism

When performing an Abort, the ICH2 drives LFRAME# active for 4 consecutive clocks. On the 4<sup>th</sup> clock, the ICH2 drives LAD[3:0] to '1111b'.

Figure 5-8. Abort Mechanism



The ICH2 performs an abort for the following cases (possible failure cases):

- ICH2 starts a Memory, I/O, or DMA cycle and no device drives a valid SYNC after 4 consecutive clocks.
- ICH2 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.

### 5.3.1.9 I/O Cycles

For I/O cycles targeting registers specified in the ICH2's decode ranges, the ICH2 performs I/O cycles as defined in the LPC specification. These are 8-bit transfers. If the processor attempts a 16-bit or 32-bit transfer, the ICH2 will break 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 ICH2 returns 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.

### 5.3.1.10 Bus Master Cycles

The ICH2 supports Bus Master cycles and requests (using LDRQ#) as defined in the LPC specification. The ICH2 has two LDRQ# inputs; thus, ICH2 supports two separate bus master devices. It uses the associated START fields for Bus Master 0 ('0010b') or Bus Master 1 ('0011b').

**Note:** The ICH2 does not support LPC Bus Masters performing I/O cycles. LPC Bus Masters should only perform memory read or memory write cycles.

### 5.3.1.11 LPC Power Management

#### LPCPD# Protocol

Same timings as for SUS\_STAT#. Upon driving SUS\_STAT# low, LPC peripherals will drive LDRQ# low or tri-state it. ICH2 shuts off the LDRQ# input buffers. After driving SUS\_STAT# active, the ICH2 drives LFRAME# low and tri-states (or drive low) LAD[3:0].

### 5.3.1.12 Configuration and ICH2 Implications

#### LPC Interface Decoders

To allow the I/O cycles and memory mapped cycles to go to the LPC I/F, the ICH2 includes several decoders. During configuration, the ICH2 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 ICH2 can not 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; however, they 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 ICH2, which supports 2 LPC bus masters, it will drive 0010 for the START field for grants to bus master #0 (requested via LDRQ[0]#) and 0011 for grants to bus master #1 (requested via LDRQ[1]#). Thus, no registers are needed to configure the START fields for a particular bus master.

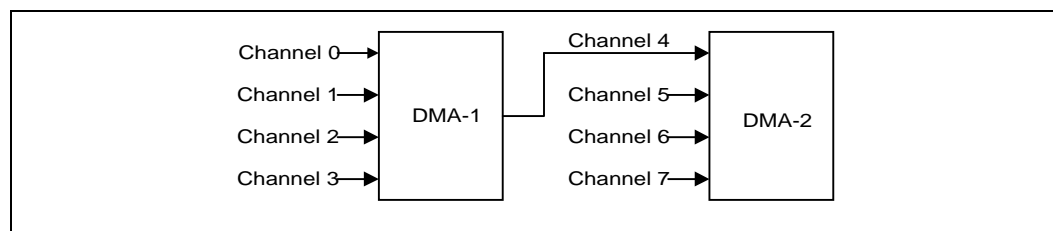
## 5.4 DMA Operation (D31:F0)

The ICH2 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 ICH2'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 5-9). 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 will default to cascade mode in the DMA Channel Mode (DCM) Register. Channel 4 is not available for any other purpose. In addition to accepting requests from DMA slaves, the DMA controller also responds to requests that software initiates. Software may initiate a DMA service request by setting any bit in the DMA Channel Request Register to a 1.

Figure 5-9. ICH2 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.

ICH2 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.

### 5.4.1 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 DMA Register description section.

### 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.

### 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) form the fourth position in the arbitration.

## 5.4.2 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 will be 010000h, not 020000h. Similarly, if a 24-bit address is 020000h and decrements, the next address will be 02FFFFh, not 01FFFFh. This is compatible with the 82C37 and Page Register implementation used in the PC-AT. This mode is set after CPURST is valid.

## 5.4.3 Summary of DMA Transfer Sizes

Table 5-10 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 is incremented or decremented.

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

**Table 5-10. DMA Transfer Size**

DMA Device Data 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 ICH2 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 5-11](#).

**Table 5-11. 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 A[16:1] A[23:17]	A0 A[16:1] A[23:17]	0 A[15:0] A[23:17]

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

### 5.4.4 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 processor 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.

## 5.4.5 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.

### 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 processor addresses 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).

### 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 acts on channels 0–3, and 0DAh acts on channels 4–7.

### 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.

## 5.5 PCI DMA

The ICH2 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, the ICH2 performs a two-cycle transfer. For example, if data is to be moved from the peripheral to main memory, the ICH2 first reads data from the peripheral and then writes the data to main memory. The location in main memory is the Current Address Registers in the 8237.

ICH2 supports up to 2 PC/PCI REQ/GNT pairs, REQ[A:B]# and GNT[A:B]#.

A 16-bit register is included in the ICH2 Function 0 PCI configuration space at offset 90h. It is divided into seven 2-bit fields that are used to configure the 7 DMA channels.

Each DMA channel can be configured to one of two options:

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

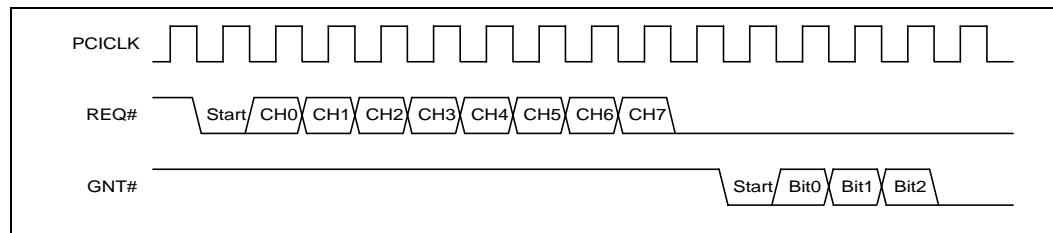
It is not possible for a particular DMA channel to be configured for more than one style of DMA; however, the seven channels can be programmed independently. For example, channel 3 can be set up for PC/PCI and channel 5 set up for LPC DMA.

The ICH2 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 ICH2 DMA controller. The REQ#/GNT# pair must follow the PC/PCI serial protocol described in the following section.

### 5.5.1 PCI DMA Expansion Protocol

The PCI expansion agent must support the PCI expansion Channel Passing Protocol defined in Figure 5-10 for both the REQ# and GNT# pins.

**Figure 5-10. 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.

The ICH2 encodes the granted channel on the GNT# line as shown above where the bits have the same meaning as shown in Figure 5-10. 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:

1. 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 ICH2. For example, if a PCI expansion agent had active requests for DMA Channel 1 and Channel 5, it would pass this information to the ICH2 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 the ICH2 that DMA channel 1 was still requesting the bus, even if that was the only request the expansion device had pending.
2. If a PCI DMA expansion agent has a request go inactive before ICH2 asserts GNT#, it must resend the expansion channel passing protocol to update the ICH2 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 ICH2 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 ICH2, the expansion agent must pull its REQ# line high for 1clock and resend the expansion channel passing information with only DMA channel 2 active. Note that the ICH2 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.
3. 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 detects 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).

## 5.5.2 PCI DMA Expansion Cycles

ICH2'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 5-12). Note that these cycles must be qualified by an active GNT# signal to the requesting device.

**Table 5-12. 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

## 5.5.3 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 5-12.

## 5.5.4 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 5-13 lists the PCI pins that the data appears 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 ICH2 performs an 8 bit memory write. The ICH2 does not assemble the I/O read into a DWord for writing to memory. Similarly, the ICH2 does not disassemble a DWord read from memory to the I/O device.

**Table 5-13. 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]

### 5.5.5 DMA Byte Enable Generation

The byte enables generated by the ICH2 on I/O reads and writes must correspond to the size of the I/O device. Table 5-14 defines the byte enables asserted for 8 and 16 bit DMA cycles.

**Table 5-14. 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) are a “don’t care”.

### 5.5.6 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 ICH2 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 ICH2 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#.

### 5.5.7 LPC DMA

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

### 5.5.8 Asserting DMA Requests

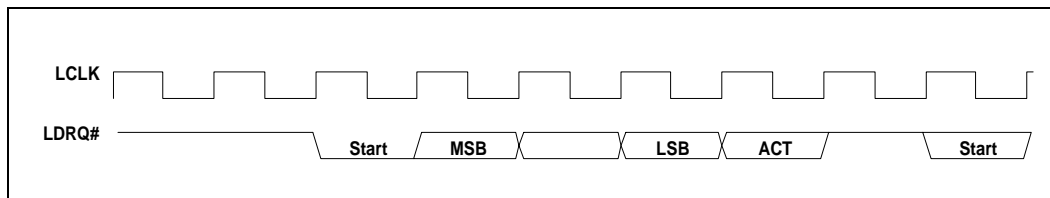
Peripherals that need DMA service encode their requested channel number on the LDRQ# signal. To simplify the protocol, each peripheral on the LPC I/F has its own dedicated LDRQ# signal (they may not be shared between two separate peripherals). The ICH2 has two LDRQ# inputs, allowing at least two devices to support DMA or bus mastering.

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

- Peripheral starts the sequence by asserting LDRQ# low (start bit). LDRQ# is high during idle conditions.
- The next 3 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 will be a 1 (high) to indicate if it is active and 0 (low) if it is inactive. The case where ACT is low will be rare, and is only used to indicate that a previous request for that channel is being abandoned.
- After the active/inactive indication, the LDRQ# signal must go high for at least 1 clock. After that one clock, LDRQ# signal can be brought low to the next encoding sequence.

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

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



### 5.5.9 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 that 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 ICH2, 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 aborts it or the host can choose to complete the cycle normally with any random data.

This method of DMA deassertion should be prevented when possible to limit boundary conditions both on the ICH2 and the peripheral.

### 5.5.10 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. ICH2 starts transfer by asserting 0000b on LAD[3:0] with LFRAME# asserted.
2. ICH2 asserts 'cycle type' of DMA, direction based on DMA transfer direction.
3. ICH2 asserts channel number and, if applicable, terminal count.
4. ICH2 indicates the size of the transfer: 8 or 16 bits.
5. If a DMA read,
  - The ICH2 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 ICH2 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.

### 5.5.11 Terminal Count

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

For example, on an 8-bit transfer size (SIZE field is 00b), if the TC bit is set, this is the last byte. On a 16-bit transfer (SIZE field is 01b), if the TC bit is set, the second byte is the last byte. Therefore, the peripheral 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.

### 5.5.12 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 does not transfer this data into main memory.

### 5.5.13 DMA Request Deassertion

An end of transfer is communicated to the ICH2 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 ICH2 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 ICH2 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 ICH2 that this is the last piece of data transferred on a DMA read (ICH2 to peripheral), or the byte which follows is the last piece of data transferred on a DMA write (peripheral to ICH2).

When the ICH2 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 ICH2 indicated a 16 bit transfer, the peripheral can end the transfer after one byte by indicating a SYNC value of 0000b or 1010b. The ICH2 will not attempt to transfer the second byte, and will deassert the DMA request internally.

If the peripheral indicates a 0000b or 1010b SYNC pattern on the last byte of the indicated size, then the ICH2 will only deassert 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, it uses a SYNC value of 1001b (ready plus more data). This indicates to the 8237 that more data bytes are requested after the current byte has been transferred; the ICH2 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 ICH2, the data will be transferred and the DMA request remains active to the 8237. At a later time, the ICH2 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 ICH2 is another grant to the peripheral if it had indicated a SYNC value of 1001b. On a single mode DMA device, the 8237 re-arbitrates after every transfer. Only demand mode DMA devices can be guaranteed that they will receive the next START indication from the ICH2.

**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, fill 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.

### 5.5.14 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 8 LCLKs after a deassertion is indicated through the SYNC field. This is needed to allow the 8237, which 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 will only perform 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 will be 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 which may appear on the LPC bus, which require higher bandwidth than 8 bit or 16 bit DMA allow, do so with a bus mastering interface and not rely on the 8237.

## 5.6 8254 Timers (D31:F0)

The ICH2 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 one 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 one counter period (838 ns) during each count cycle. The initial count value is loaded one 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).

### 5.6.1 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.

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

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

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

Table 5-15 lists the six operating modes for the interval counters.

**Table 5-15. 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 retriggeable 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.

## 5.6.2 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.

### 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 will not return a determinate value because the counting process is asynchronous to read operations. However, in the case of counter 2, the count can be stopped by writing to the GATE bit in port 61h.

## Counter Latch Command

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

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

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 will be the count at the time the first Counter Latch Command was issued.

## Read Back Command

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

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

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

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

If both count and status of a counter are latched, the first read operation from that counter will return 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, return the latched count. Subsequent reads return unlatched count.

## 5.7 8259 Interrupt Controllers (PIC) (D31:F0)

The ICH2 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 8 interrupts, numbered 0–7. [Table 5-16](#) shows how the cores are connected.

**Table 5-16. Interrupt Controller Core Connections**

8259	8259 Input	Typical Interrupt Source	Connected Pin / Function
Master	0	Internal	Internal Timer / Counter 0 output
	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
	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 or via SERIRQ
	7	Secondary IDE Cable	IRQ15 from input signal or via SERIRQ

The ICH2 cascades the slave controller onto the master controller through master controller interrupt input 2. This means there are only 15 possible interrupts for the ICH2 PIC. Interrupts can individually be programmed to be edge or level, except for IRQ[0, 2, 8#, 13].

Note that previous PIIXn devices internally latched IRQ[12 and 1] and required a port 60h read to clear the latch. The ICH2 can be programmed to latch IRQ[12 or 1] (see bit 11 and bit 12 in General Control Register, D31:F0, offset D0h).

## 5.7.1 Interrupt Handling

### 5.7.1.1 Generating Interrupts

The PIC interrupt sequence involves three bits, from the IRR, ISR, and IMR for each interrupt level. These bits are used to determine the interrupt vector returned, and status of any other pending interrupts. [Table 5-17](#) defines the IRR, ISR, and IMR.

**Table 5-17. Interrupt Status Registers**

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

### 5.7.1.2 Acknowledging Interrupts

The processor generates an interrupt acknowledge cycle that is translated by the host bridge into a PCI Interrupt Acknowledge Cycle to the ICH2. The PIC translates this command into two internal INTA# pulses expected by the 8259 cores. The PIC uses the first internal INTA# pulse to freeze the state of the interrupts for priority resolution. On the second INTA# pulse, the master or slave sends the interrupt vector to the processor with the acknowledged interrupt code. This code is based on bits [7:3] of the corresponding ICW2 register combined with three bits representing the interrupt within that controller.

**Table 5-18. Content of Interrupt Vector Byte**

Master,Slave Interrupt	Bits [7:3]	Bits [2:0]
IRQ[7,15]	ICW2[7:3]	111
IRQ[6,14]		110
IRQ[5,13]		101
IRQ[4,12]		100
IRQ[3,11]		011
IRQ[2,10]		010
IRQ[1,9]		001
IRQ[0,8]		000

### 5.7.1.3 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 ICH2.
4. Upon observing its own interrupt acknowledge cycle on PCI, the ICH2 converts it into the two cycles that the internal 8259 pair can respond. 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 will return 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.

## 5.7.2 Initialization Command Words (ICWx)

Before the operation can begin, each 8259 must be initialized. In the ICH2 this is a four-byte sequence. The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4.

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

### ICW1

An I/O write to the master or slave controller base address with data bit 4 equal to 1 is interpreted as a write to ICW1. Upon sensing this write, the ICH2 PIC expects three more byte writes to 21h for the master controller, or A1h for the slave controller, to complete the ICW sequence.

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

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

## ICW2

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

## ICW3

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

- For the master controller, ICW3 is used to indicate which IRQ input line is used to cascade the slave controller. Within the ICH2, IRQ2 is used. Therefore, bit 2 of ICW3 on the master controller is set to a 1 and the other bits are set to 0s.
- For the slave controller, ICW3 is the slave identification code used during an interrupt acknowledge cycle. On interrupt acknowledge cycles, the master controller broadcasts a code to the slave controller if the cascaded interrupt won arbitration on the master controller. The slave controller compares this identification code to the value stored in its ICW3, and if it matches, the slave controller assumes responsibility for broadcasting the interrupt vector.

## ICW4

The final write in the sequence (ICW4) must be programmed 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.

### 5.7.3 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.

## 5.7.4 Modes of Operation

### Fully Nested Mode

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

### Special Fully Nested Mode

This mode will be 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 will be 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 will be recognized by the master and will initiate interrupts to the processor. In the normal nested mode, a slave is masked out when its request is in service.
- When exiting the Interrupt Service routine, software has to check whether the interrupt serviced was the only one from that slave. This is done by sending a Non-Specific EOI command to the slave and then reading its ISR. If it is 0, a non-specific EOI can also be sent to the master.

### Automatic Rotation Mode (Equal Priority Devices)

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

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

### Specific Rotation Mode (Specific Priority)

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

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

### Poll Mode

Poll Mode can be used to conserve space in the interrupt vector table. Multiple interrupts that can be serviced by one interrupt service routine do not need separate vectors if the service routine uses the poll command. Polled 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 will contain a 1' in bit 7 if there is an interrupt, and the binary code of the highest priority level in bits 2:0.

### Cascade Mode

The PIC in the ICH2 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 3-bit internal bus. In the ICH2, 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.

### Edge-Triggered 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 ICH2, this bit is disabled and a new register for edge-triggered 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-triggered 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 will be returned.

### 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.

### 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 will clear 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 ICH2, as the interrupt being serviced currently is the interrupt entered with the interrupt acknowledge. When the PIC is operated in modes which preserve the fully nested structure, software can determine which ISR bit to clear by issuing a Specific EOI. An ISR bit that is masked will not be cleared by a Non-Specific EOI if the PIC is in the Special Mask Mode. An EOI command must be issued for both the master and slave controller.

### Automatic End of Interrupt Mode

In this mode, the PIC will automatically perform 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.

## 5.7.5 Masking Interrupts

### Masking on an Individual Interrupt Request

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

### Special Mask Mode

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

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

## 5.7.6 Steering PCI Interrupts

The ICH2 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 ICH2 will internally invert the PIRQx# line to send an active high level to the PIC. When a PCI interrupt is routed onto the PIC, the selected IRQ can no longer be used by an ISA device (through SERIRQ). However, active low non-ISA interrupts can share their interrupt with PCI interrupts.

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

In addition to the standard ISA compatible interrupt controller (PIC) described in the previous section, the ICH2 incorporates the Advanced Programmable Interrupt Controller (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.

### 5.8.1 Interrupt Handling

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

- **Method of Interrupt Transmission.** The I/O APIC transmits interrupts through a 3-wire bus 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 ICH2 supports a total of 24 interrupts.
- **Multiple Interrupt Controllers.** The I/O APIC interrupt transmission protocol has an arbitration phase that allows for multiple I/O APICs in the system with their own interrupt vectors. The ICH2 I/O APIC must arbitrate for the APIC bus before transmitting its interrupt message.

## 5.8.2 Interrupt Mapping

The I/O APIC within the ICH2 supports 24 APIC interrupts. Each interrupt has its own unique vector assigned by software. The interrupt vectors are mapped as follows and match “configuration 6” of the Multi-processor specification.

**Table 5-19. APIC Interrupt Mapping**

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
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
9	Yes	No	Yes	Option for SCI, TCO
10	Yes	No	Yes	Option for SCI, TCO
11	Yes	No	Yes	Option for SCI, TCO
12	Yes	No	Yes	
13	No	No	No	FERR# logic
14	Yes	Yes	Yes	
15	Yes	Yes	Yes	
16	PIRQA	PIRQA	No	
17	PIRQB	PIRQB	No	AC'97 Audio, Modem, option for SMBus
18	PIRQC	PIRQC	No	
19	PIRQD	PIRQD	No	USB #1
20	N/A	PIRQE	Yes	LAN, option for SCI, TCO
21	N/A	PIRQF	Yes	Option for SCI, TCO
22	N/A	PIRQG	Yes	Option for SCI, TCO
23	N/A	PIRQH	Yes	USB #2, option for SCI, TCO

## 5.8.3 APIC Bus Functional Description

### 5.8.3.1 Physical Characteristics of APIC

The APIC bus is a 3-wire synchronous bus connecting all I/O and local APICs. Two of these wires are used for data transmission and one wire is a clock. For bus arbitration, the APIC uses only one of the data wires. The bus is logically a wire-OR and electrically an open-drain connection providing for both bus use arbitration and arbitration for lowest priority. The APIC bus speed can run from 16.67 MHz to 33 MHz.

### 5.8.3.2 APIC Bus Arbitration

The I/O APIC uses one wire arbitration to win bus ownership. A rotating priority scheme is used for APIC bus arbitration. The winner of the arbitration becomes the lowest priority agent and assumes an arbitration ID of 0. All other agents, except the agent whose arbitration ID is 15, increment their Arbitration IDs by one. The agent whose ID was 15 will take the winner's arbitration ID and will increment it by one. Arbitration IDs are changed only for messages that are transmitted successfully (except for the Low Priority messages). A message is transmitted successfully if no CS error or acceptance error was reported for that message.

An APIC agent can use two different priority schemes: Normal or EOI. EOI has the highest priority. EOI priority is used to send EOI messages for level interrupts from a local APIC to an I/O APIC. When an agent requests the bus with EOI priority, all other agents requesting the bus with normal priorities will back off.

When ICH2 detects a bus idle condition on the APIC Bus and it has an interrupt to send over the APIC bus, it drives a start cycle to begin arbitration, by driving bit 0 to a '0' on an APICCLK rising edge. It then samples bit 1. If Bit 1 was a '0', then a local APIC started arbitration for an EOI message on the same clock edge that the ICH2 started arbitration. Thus, the ICH2 has lost arbitration and stops driving the APIC bus.

If the ICH2 did not detect an EOI message start, it will start transferring its arbitration ID, located in bits [27:24] of its Arbitration ID register (ARBID). Starting in Cycle 2 through Cycle 5, it will tri-state bit 0, and drive bit 1 to a '0' if ARBID[27] is a '1'. If ARBID[27] is a '0', it will also tri-state bit 1. At the end of each cycle, the ICH2 samples the state of Bit 1 on the APIC bus. If the ICH2 did not drive Bit 1 (ARBID[27] = '0') and it samples a '0', then another APIC agent started arbitration for the APIC bus at the same time as the ICH2, and it has higher priority. The ICH2 will stop driving the APIC bus. [Table 5-20](#) describes the arbitration cycles.

**Table 5-20. Arbitration Cycles**

Cycle	Bit 1	Bit 0	Comment
1	EOI	0	Bit 1 = 1: Normal, Bit 1 = 0: EOI
2	NOT (ARBID[27])	1	Arbitration ID. If ICH2 samples a different value than it sent, it lost arbitration.
3	NOT (ARBID[26])	1	
4	NOT (ARBID[25])	1	
5	NOT (ARBID[24])	1	

### 5.8.3.3 Bus Message Formats

After bus arbitration, the winner is granted exclusive use of the bus and will drive its message. APIC messages come in four formats determined by the delivery mode bits. These four messages are of different length and are known by all APICs on the bus through the transmission of the Delivery Mode bits.

**Table 5-21. APIC Message Formats**

Message	# of Cycles	Delivery Mode Bits	Comments
EOI	14	xxx	End of Interrupt transmission from Local APIC to I/O APIC on Level interrupts. EOI is known by the EOI bit at the start of arbitration.
Short	21	001, 010, 100, 101, 111	I/O APIC delivery on Fixed, NMI, SMI, Reset, ExtINT, and Lowest Priority with focus processor messages.
Lowest Priority	33	001	Transmission of Lowest Priority interrupts when the status field indicates that the processor does not have focus.
Remote Read	39	011	Message from one Local APIC to another to read registers.

#### EOI Message For Level-Triggered Interrupts

EOI messages are used by local APICs to send an EOI cycle occurring for a level-triggered interrupt to an I/O APIC. This message is needed so that the I/O APIC can differentiate between a new interrupt on the interrupt line versus the same interrupt on the interrupt line. The target of the EOI is given by the local APIC through the transmission of the priority vector (V7 through V0) of the interrupt. Upon receiving this message, the I/O APIC resets the Remote IRR bit for that interrupt. If the interrupt signal is still active after the IRR bit is reset, the I/O APIC will treat it as a new interrupt.

**Table 5-22. EOI Message**

Cycle	Bit 1	Bit 0	Comments
1	0	0	EOI message
2–5	ARBID	1	Arbitration ID
6	NOT(V7)	NOT(V6)	Interrupt vector bits V7 - V0 from redirection table register
7	NOT(V5)	NOT(V4)	
8	NOT(V3)	NOT(V2)	
9	NOT(V1)	NOT(V0)	
10	NOT(C1)	NOT(C0)	Check Sum from Cycles 6 - 9
11	1	1	Postamble
12	NOT(A)	NOT(A)	Status Cycle 0
13	NOT(A1)	NOT(A1)	Status Cycle 1
14	1	1	Idle

## Short Message

Short messages are used for the delivery of Fixed, NMI, SMI, Reset, ExtINT and Lowest Priority with Focus processor interrupts. The delivery mode bits (M2-M0) specify the message. All short messages take 21 cycles including the idle cycle.

**Table 5-23. Short Message**

Cycle	Bit 1	Bit 0	Comments
1	1	0	Normal Arbitration
2–5	ARBID	1	Arbitration ID
6	NOT(DM)	NOT(M2)	DM <sup>1</sup> = Destination Mode from bit 11 of the redirection table register
7	NOT(M1)	NOT(M0)	M2-M0 = Delivery Mode from bits 10:8 of the redirection table register
8	NOT(L)	NOT(TM)	L = Level, TM = Trigger Mode
9	NOT(V7)	NOT(V6)	Interrupt vector bits V7–V0 from redirection table register
10	NOT(V5)	NOT(V4)	
11	NOT(V3)	NOT(V2)	
12	NOT(V1)	NOT(V0)	
13	NOT(D7)	NOT(D6)	Destination field from bits 63:56 of redirection table register <sup>1</sup>
14	NOT(D5)	NOT(D4)	
15	NOT(D3)	NOT(D2)	
16	NOT(D1)	NOT(D0)	
17	NOT(C1)	NOT(C0)	Checksum for Cycles 6–16 <sup>2</sup>
18	1	1	Postamble <sup>3</sup>
19	NOT(A)	NOT(A)	Status Cycle 0. See <a href="#">Table 5-24</a> .
20	NOT(A1)	NOT(A1)	Status Cycle 1. See <a href="#">Table 5-24</a> .
21	1	1	Idle

### NOTES:

1. If DM is 0 (physical mode), then cycles 15 and 16 are the APIC ID and cycles 13 and 14 are sent as '1'. If DM is 1 (logical mode), then cycles 13 through 16 are the 8-bit Destination field. The interpretation of the logical mode 8-bit Destination field is performed by the local units using the Destination Format Register. Shorthands of "all-incl-self" and "all-excl-self" both use Physical Destination mode and a destination field containing APIC ID value of all ones. The sending APIC knows whether it should (incl) or should not (excl) respond to its own message.
2. The checksum field is the cumulative add (mod 4) of all data bits (DM, M0-3, L, TM, V0-7, D0-7). The APIC driving the message provides this checksum. This, in essence, is the lower two bits of an adder at the end of the message.
3. This cycle allows all APICs to perform various internal computations based on the information contained in the received message. One of the computations takes the checksum of the data received in cycles 6 through 16 and compares it with the value in cycle 18. If any APIC computes a different checksum than the one passed in cycle 17, then that APIC will signal an error on the APIC bus ("00") in cycle 19. If this happens, all APICs will assume the message was never sent and the sender must try sending the message again, which includes re-arbitrating for the APIC bus. In lowest priority delivery when the interrupt has a focus processor, the focus processor will signal this by driving a "01" during cycle 19. This tells all the other APICs that the interrupt has been accepted, the arbitration is preempted, and short message format is used. Cycle 19 and 20 indicates the status of the message (i.e., accepted, check sum error, retry or error). [Table 5-24](#) shows the status signal combinations and their meanings for all delivery modes.

Table 5-24. APIC Bus Status Cycle Definition

Delivery Mode	A	Comments	A1	Comments
Fixed, EOI	11	Checksum OK	1x	Error
			01	Accepted
			00	Retry
	10	Error	xx	
	01	Error	xx	
	00	Checksum Error	xx	
NMI, SMM, Reset, ExtINT	11	Checksum OK	1x	Error
			01	Accepted
			00	Error
	10	Error	xx	
	01	Error	xx	
	00	Checksum Error	xx	
Lowest Priority	11	Checksum OK: No Focus Processor	1x	Error
			01	End and Retry
			00	Go for Low Priority Arbitration
	10	Error	xx	
	01	Checksum OK: Focus Processor	xx	
	00	Checksum Error	xx	
Remote Read	11	Checksum OK	xx	
	10	Error	xx	
	01	Error	xx	
	00	Checksum Error	xx	

## Lowest Priority without Focus Processor (FP) Message

This message format is used to deliver an interrupt in the lowest priority mode in which it does not have a Focus Process. Cycles 1 through 21 for this message are same as for the short message discussed above. Status cycle 19 identifies if there is a Focus processor (10) and a status value of 11 in cycle 20 indicates the need for lowest priority arbitration.

**Table 5-25. Lowest Priority Message (Without Focus Processor)**

Cycle	Bit 1	Bit 0	Comments
1	1	0	Normal Arbitration
2–5	ARBID	1	Arbitration ID
6	NOT(DM)	NOT(M2)	DM = Destination Mode from bit 11 of the redirection table register
7	NOT(M1)	NOT(M0)	M2-M0 = Delivery Mode from bits 10:8 of the redirection table register
8	NOT(L)	NOT(TM)	L = Level, TM = Trigger Mode
9	NOT(V7)	NOT(V6)	Interrupt vector bits V7–V0 from redirection table register
10	NOT(V5)	NOT(V4)	
11	NOT(V3)	NOT(V2)	
12	NOT(V1)	NOT(V0)	
13	NOT(D7)	NOT(D6)	Destination field from bits 63:56 of redirection table register
14	NOT(D5)	NOT(D4)	
15	NOT(D3)	NOT(D2)	
16	NOT(D1)	NOT(D0)	
17	NOT(C1)	NOT(C0)	Checksum for Cycles 6–16
18	1	1	Postamble
19	NOT(A)	NOT(A)	Status Cycle 0.
20	NOT(A1)	NOT(A1)	Status Cycle 1.
21	P7	1	Inverted Processor Priority P7–P0
22	P6	1	
23	P5	1	
24	P4	1	
25	P3	1	
26	P2	1	
27	P1	1	
28	P0	1	
29	ArbID3	1	
30	ArbID2	1	
31	ArbID1	1	
32	ArbID0	1	
33	S	S	Status
34	1	1	Idle

### NOTES:

1. Cycle 21 through 28 are used to arbitrate for the lowest priority processor. The processor that takes part in the arbitration drives the processor priority on the bus. Only the local APICs that have "free interrupt slots" will participate in the lowest priority arbitration.

2. Cycles 29 through 32 are used to break tie in case two more processors have lowest priority. The bus arbitration IDs are used to break the tie.

### Remote Read Message

Remote read message is used when a local APIC wishes to read the register in another local APIC. The message format is same as short message for the first 21 cycles.

**Table 5-26. Remote Read Message**

Cycle	Bit 1	Bit 0	Comments
1	1	0	Normal Arbitration
2–5	ARBID	1	Arbitration ID
6	NOT(DM)	NOT(M2)	DM = Destination Mode from bit 11 of the redirection table register
7	NOT(M1)	NOT(M0)	M2-M0 = Delivery Mode from bits 10:8 of the redirection table register
8	NOT(L)	NOT(TM)	L = Level, TM = Trigger Mode
9	NOT(V7)	NOT(V6)	Interrupt vector bits V7 - V0 from redirection table register
10	NOT(V5)	NOT(V4)	
11	NOT(V3)	NOT(V2)	
12	NOT(V1)	NOT(V0)	
13	NOT(D7)	NOT(D6)	Destination field from bits 63:56 of redirection table register
14	NOT(D5)	NOT(D4)	
15	NOT(D3)	NOT(D2)	
16	NOT(D1)	NOT(D0)	
17	NOT(C1)	NOT(C0)	Checksum for Cycles 6 - 16
18	1	1	Postamble
19	NOT(A)	NOT(A)	Status Cycle 0.
20	NOT(A1)	NOT(A1)	Status Cycle 1.
21	d31	d30	Remote register data 31-0
22	d29	d28	
23	d27	d26	
24	d25	d24	
25	d23	d22	
26	d21	d20	
27	d19	d18	
28	d17	d16	
29	d15	d14	
30	d13	d12	
31	d11	d10	
32	d09	d08	
33	d07	d06	
34	d05	d04	
35	d03	d02	
36	d01	d00	
37	S	S	Data Status: 00 = valid, 11 = invalid
38	C	C	Check Sum for data d31-d00
39	1	1	Idle

**NOTE:** Cycle 21 through 36 contain the remote register data. The status information in cycle 37 specifies if the data is good or not. Remote read cycle is always successful (although the data may be valid or invalid)

in that it is never retried. The reason for this is that Remote Read is a debug feature, and a "hung" remote APIC that is unable to respond should not cause the debugger to hang.

## 5.8.4 PCI Message-Based Interrupts

### 5.8.4.1 Theory of Operation

The following scheme is only supported when the internal I/O(x) APIC is used (rather than just the 8259). The ICH2 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 ICH2's internal I/O APIC is enabled. Upon recognizing the write from the peripheral, the ICH2 sends the interrupt message to the processor using the I/O APIC's serial bus.

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 ICH2, the MESSAGE\_ADDRESS is the IRQ Pin Assertion Register, which is mapped to memory location: FEC0\_0020h (same as APIC).
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 will be 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 ICH2 positively decodes the cycles (as a slave) in medium time.
4. The ICH2 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 ICH2 supports interrupts 00h through 23h. Binary values outside this range will not cause any action.
5. After sending the interrupt message to the processor, the ICH2 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 (e.g., the standard PCI PIRQ[A:D], those received via SERIRQ#, or the internal level-triggered interrupts such as SCI or TCO).

The ICH2 ignores interrupt messages sent by PCI masters that attempt to use IRQ[0,2,8, or 13].

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

#### Capabilities Indication

The capability to support PCI interrupt delivery will be indicated via ACPI configuration techniques. This involves the BIOS creating a data structure that gets reported to the ACPI configuration software. The operating system reads the PRQ bit in the APIC Version Register to see if the ICH2 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.

## 5.8.5 Front-Side Interrupt Delivery

### 5.8.5.1 Theory of Operation

For processors that support Front-Side Bus interrupt delivery, the ICH2 has an option to let the integrated I/O APIC behave as an I/O (x) APIC. In this case, it delivers interrupt messages to the processor in a parallel manner, rather than using the I/O APIC serial scheme. The ICH2 is intended to be compatible with the I/O (x) APIC specification, Rev 1.1

This is done by the ICH2 writing (via the Hub Interface) directly to a memory location that is snooped by the processor(s). The processor(s) snoop the cycle to know which one goes active.

The processor enables the mode by setting the I/O APIC Enable (APIC\_EN) bit and by setting the DT bit in the I/O APIC ID register.

The following sequence is used:

1. When the ICH2 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 ICH2 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 ICH2 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 5.8.5.5](#).

**Note:** FSB Interrupt Delivery compatibility with CPU clock control depends on the CPU, not the ICH2.

### 5.8.5.2 Edge-Triggered Operation

In this case, the “Assert Message” is sent when there is an inactive-to-active edge on the interrupt. The “Deassert Message” is not used.

### 5.8.5.3 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.

If the interrupt was active but goes inactive before the EOI is received, the “Deassert Message” is sent.

### 5.8.5.4 Registers Associated with Front-Side Bus Interrupt Delivery

#### Capabilities Indication

The capability to support Front-Side bus interrupt delivery will be indicated via ACPI configuration techniques. This involves BIOS creating a data structure that gets reported to the ACPI configuration software.

## DT bit in the Boot Configuration Register

This enables the ICH2 to deliver interrupts as memory writes. This bit is ignored if the APIC mode is not enabled.

### 5.8.5.5 Interrupt Message Format

ICH2 writes the message to PCI (and to the Host Controller) as a 32-bit memory write cycle. It uses the formats shown in [Table 5-27](#) and [Table 5-28](#) for the address and data.

**Table 5-27. Interrupt Message Address Format**

Bit	Description
31:20	Will always be FEEh
19:12	<b>Destination ID:</b> This is the same as bits 63:56 of the I/O Redirection Table entry for the interrupt associated with this message.
11:4	Reserved (will always be 0)
3	<b>Redirection Hint:</b> This bit is used by the processor host bridge to allow the interrupt message to be redirected. 0 = The message will be delivered to the agent (processor) listed in bits 19:4. 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). The Redirection Hint bit = 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 = 0.
2	<b>Destination Mode:</b> This bit is used only the Redirection Hint bit = 1. If the Redirection Hint bit and the Destination Mode bit are both set to 1, the logical destination mode is used and the redirection is limited only to those processors that are part of the logical group as based on the logical ID.
1:0	Will always be 00.

**Table 5-28. Interrupt Message Data Format**

Bit	Description
31:16	Will always be 0000h.
15	<b>Trigger Mode:</b> 1 = Level, 0 = Edge. Same as the corresponding bit in the I/O Redirection Table for that interrupt.
14	<b>Delivery Status:</b> 1 = Assert, 0 = Deassert. If using edge-triggered interrupts, then this bit will always be 1, since only the assertion is sent. If using level-triggered interrupts, then this bit indicates the state of the interrupt input.
13:12	Will always be 00
11	<b>Destination Mode:</b> 1 = Logical. 0 = Physical. Same as the corresponding bit in the I/O Redirection Table for that interrupt.
10:8	<b>Delivery Mode:</b> This is the same as the corresponding bits in the I/O Redirection Table for that interrupt. 000 = Fixed 100 = NMI 001 = Lowest Priority 101 = INIT 010 = SMI/PMI 110 = Reserved 011 = Reserved 111 = ExtINT
7:0	<b>Vector:</b> This is the same as the corresponding bits in the I/O Redirection Table for that interrupt.

## 5.9 Serial Interrupt (D31:F0)

ICH2 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 ICH2, 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 ICH2 supports a message for 21 serial interrupts. These represent the 15 ISA interrupts (IRQ[0,1, 2:15]), the four PCI interrupts, and the SMI# and IOCHK# control signals. The serial IRQ protocol does not support the additional APIC interrupts (20–23).

### 5.9.1 Start Frame

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

- Continuous, where the ICH2 is solely responsible for generating the start frame
- Quiet, where a serial IRQ peripheral is responsible for beginning the start frame.

The mode that must first be entered when enabling the serial IRQ protocol is continuous mode. In this mode, the ICH2 will assert 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 ICH2 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 ICH2 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.

### 5.9.2 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, the SERIRQ device tri-states the SERIRQ signal. The SERIRQ line remains high due to pull-up resistors. 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 will tri-state the SERIRQ line.

### 5.9.3 Stop Frame

After all data frames, a Stop Frame is driven by ICH2. The SERIRQ signal is driven low by ICH2 for 2 or 3 PCI clocks. The number of clocks is determined by the SERIRQ configuration register. The number of clocks determines the next mode.

**Table 5-29. Stop Frame Explanation**

Stop Frame Width	Next Mode
2 PCI clocks	<b>Quiet Mode.</b> Any SERIRQ device may initiate a Start Frame
3 PCI clocks	<b>Continuous Mode.</b> Only the host (ICH2) may initiate a Start Frame

### 5.9.4 Specific Interrupts not Supported via SERIRQ

There are three interrupts seen through the serial stream that are not supported by the ICH2. 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#.

ICH2 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 ICH2.

## 5.9.5 Data Frame Format

Table 5-30 shows the format of the data frames. For the PCI interrupts (A-D), the output from the ICH2 is ANDed with the PCI input signal. Thus, the interrupt can be signaled via both the PCI interrupt input signal and via the SERIRQ signal (they are shared).

**Table 5-30. 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. Sets 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#

## 5.10 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 (128 bytes each); 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 us 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 that configure and report RTC functions.

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

**Note:** The ICH2 supports the ability to generate an SMI# based on year 2000 rollover. See [Section 5.10.4](#) for more information on the century rollover.

The ICH2 does not implement month/year alarms.

### 5.10.1 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 is incremented, overflow checked, a matching alarm condition checked, and the time and date are rewritten to the RAM locations. The update cycle starts at least 488 us after the UIP bit of register A is asserted and the entire cycle does not take more than 1984 us to complete. The time and date RAM locations (0–9) are disconnected from the external bus during this time.

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

## 5.10.2 Interrupts

The real-time clock interrupt is internally routed within the ICH2 both to the I/O APIC and the 8259. It is mapped to interrupt vector 8. This interrupt does not leave the ICH2, nor is it shared with any other interrupt. IRQ8# from the SERIRQ stream is ignored.

## 5.10.3 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 are not readable or writable. A write cycle to those locations has no effect. A read cycle to those locations does 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 invokes BIOS and allows it to relock the RAM range.

## 5.10.4 Century Rollover

ICH2 detects a rollover when the Year byte (RTC I/O space, index offset 09h) transitions from 99 to 00. Upon detecting the rollover, the ICH2 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 ICH2 also sets the NEWCENTURY\_STS bit; 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.

## 5.10.5 Clearing Battery-Backed RTC RAM

Clearing CMOS RAM in an ICH2-based platform can be done by using a jumper on RTCRST# or GPI or using the 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 the 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) is set and the configuration bits in the RTC power well are set to their default state. BIOS can monitor the state of this bit and manually clear the RTC CMOS array once the system is booted. The normal position would cause RTCRST# to be pulled up through a weak pull-up resistor. Table 5-31 shows which bits are set to their default state when RTCRST# is asserted.

**Table 5-31. Configuration Bits Reset By RTCRST# Assertion**

Bit Name	Default State	Register	Location	Bit(s)
CPU_BIST_EN	0	General Status	D31:F0:D7-D4	12
FREQ_STRAP[3:0]	0Fh	General Status	D31:F0:D7-D4	11:8
	undefined	RTC Register A	I/O space	ALL
PIE, UIE, SQWE	0, 0, 0	RTC Register B	I/O space	6,4,3
IRQF, PF, AF, UF	0, 0, undef, 0	RTC Register C	I/O space	7,6,5,4
VRT, Date Alarm	1, undefined	RTC Register D	I/O space	7, 5:0
RTC_PWR_STS	0	General PM Configuration 3	D31:F0:A4	2
PWR_FLR	0	General PM Configuration 3	D31:F0:A4	1
AFTERG3_EN	0	General PM Configuration 3	D31:F0:A4	0
PRBTNOR_STS	0	Power Management 1 Status	D31:F0 PMBase+0h	11
PME_EN, RI_STS	0, 0	General Purpose Event 0 Enable	D31:F0 PMBase+2Ah	11,8
NEW_CENTURY_STS	0	TCO1 Status	D31:F0 TCOBase+04h	7
INTRD_DET	0	TCO2 Status	D31:F0 TCOBase+06h	0

### Using a GPI to clear CMOS

A jumper on a GPI can also be used to clear CMOS values. BIOS detects 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 detects 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 a 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.

## 5.11 Processor Interface (D31:F0)

The ICH2 interfaces to the processor with a variety of signals:

- Standard outputs to the processor: A20M#, SMI#, NMI, INIT#, INTR, STPCLK#, IGNNE#, CPUSLP#
- Standard input from the processor: FERR#

Most ICH2 outputs to the processor use standard buffers. The ICH2 has a separate Vcc signal that is pulled up at the system level to the processor voltage and thus, determines Voh 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 ICH2 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 ICH2 does not support the processor's FRC mode.

### 5.11.1 Processor Interface Signals

This section describes each of the signals that interface between the ICH2 and the processor(s). Note that the behavior of some signals may vary during processor reset, as the signals are used for frequency strapping.

#### 5.11.1.1 A20M#

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).

#### 5.11.1.2 INIT#

The INIT# signal is active (driven low) based on any one of several events described in [Table 5-32](#). When any of these events occur, INIT# is driven low for 16 PCI clocks, then driven high.

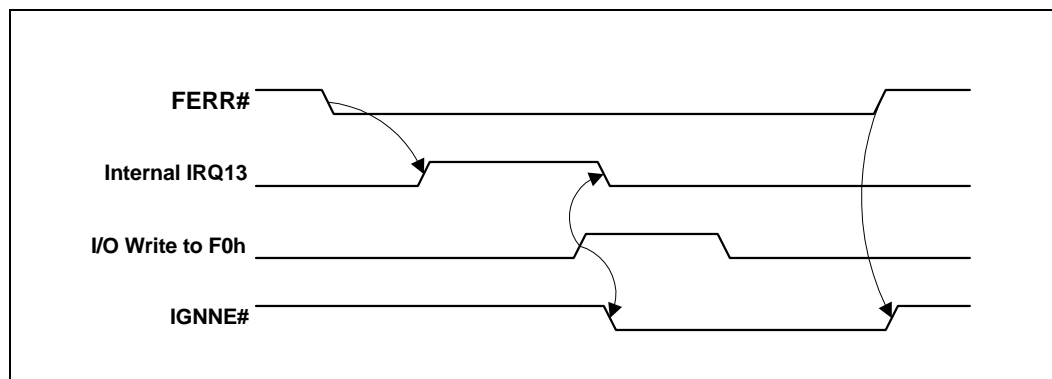
**Note:** The 16-clock counter for INIT# assertion halts while STPCLK# is active. Therefore, if INIT# is supposed to go active while STPCLK# is asserted, it actually goes active after STPCLK# goes inactive.

**Table 5-32. INIT# Going Active**

Cause of INIT# Going Active	Comment
Shutdown special cycle from the processor.	
PORT92 write, where INIT_NOW (bit 0) transitions from a 0 to a 1.	
PORTCF9 write, where RST_CPU (bit 2) was a 0 and SYS_RST(bit 1) 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 ICH2 arms INIT# to be generated again.
Processor BIST	To enter BIST, the software sets CPU_BIST_EN bit and then does a full processor reset using the CF9 register.

### 5.11.1.3 FERR#/IGNNE# (Coprocessor Error)

The ICH2 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 ICH2 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 5-12. Coprocessor Error Timing Diagram**


If COPROC\_ERR\_EN is not set, the assertion of FERR# will not generate an internal IRQ13; the write to F0h will not generate IGNNE#.

### 5.11.1.4 NMI

Non-Maskable Interrupts (NMIs) can be generated by several sources, as described in [Table 5-33](#).

**Table 5-33. 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).

### 5.11.1.5 STPCLK# and CPUSLP# Signals

The ICH2 power management logic controls these active-low signals. Refer to [Section 5.12](#) for more information on the functionality of these signals.

### 5.11.1.6 CPUPWRGOOD Signal

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 ICH2's PWROK and VRMPWRGD signals.

## 5.11.2 Dual Processor Issues

### 5.11.2.1 Signal Differences

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

**Table 5-34. DP Signal Differences**

Signal	Difference
A20M# / A20GATE	Generally not used, but still supported by ICH2.
STPCLK#	Used for S1 State as well as preparation for entry to S3–S5 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 ICH2.

### 5.11.2.2 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 ICH2 to know when it is safe to deassert STPCLK#.

Because the S1 state has the STPCLK# signal active, the STPCLK# signal can be connected to both processors. However, for ACPI implementations, the ICH2 does not support the C2 state for both processors, since there are not two processor control blocks. BIOS must indicate that the ICH2 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 ICH2 asserts STPCLK# to both processors. To meet the processor specifications, the CPUSLP# signal has to be delayed until the 2<sup>nd</sup> Stop-Grant cycle occurs. To ensure this, the ICH2 waits a minimum of 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 ICH2 asserts CPUSLP# 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 it cannot be guaranteed that both processors will be idle, the SLP\_EN bit must not be enabled. 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 and 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.

### 5.11.3 Speed Strapping for Processor

The ICH2 directly sets the speed straps for the processor, saving the external logic that has been needed with prior PCIsets. Refer to the processor specification for speed strapping definition. The ICH2 performs the following to set the speed straps for the processor:

1. While PCIRST# is active, the ICH2 drives A20M#, IGNNE#, NMI, and INTR high.
2. As soon as PWROK goes active, the ICH2 reads the FREQ\_STRAP field contents.
3. The next step depends on the power state being exited as described in [Table 5-35](#).

**Table 5-35. Frequency Strap Behavior Based on Exit State**

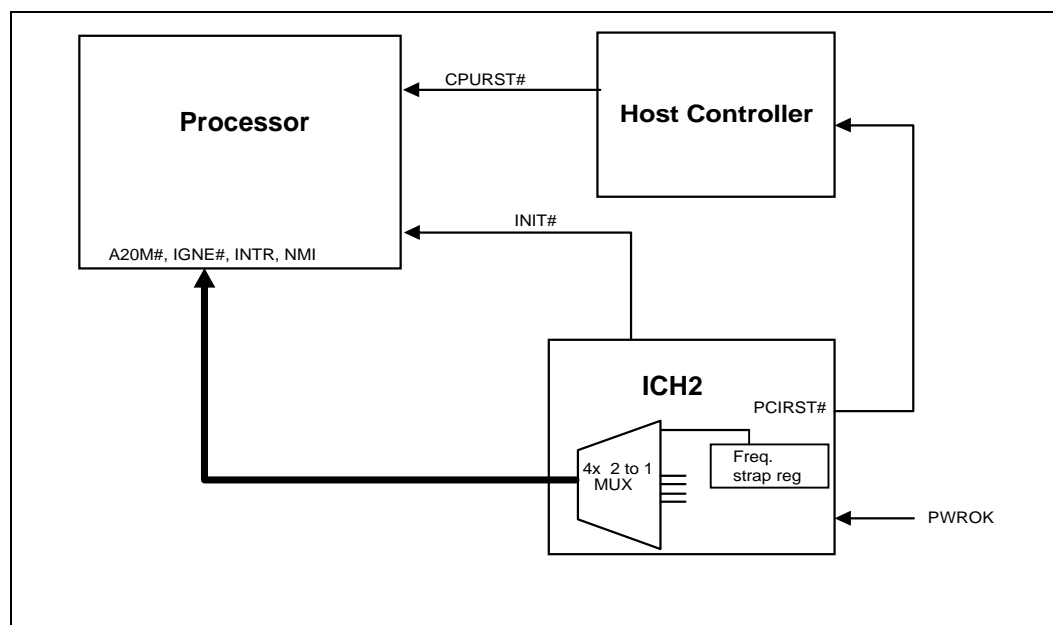
State Exiting	ICH2
S1	There is no processor reset, so no frequency strap logic is used.
S3, S4, S5, or G3	Based on PWROK going active, the ICH2 deasserts PCIRST#, and based on the value of the FREQ_STRAP field (D31:F0, Offset D4), the ICH2 drives the intended core frequency values on A20M#, IGNNE#, NMI, and INTR. The ICH2 holds these signals for 120 ns after CPURST# is deasserted by the Host controller.

**Table 5-36. 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 1111 via a pinstrap (AC\_SDOUT signal), or the ICH2 can automatically force the speed strapping to 1111 if the processor fails to boot.

**Figure 5-13. Signal Strapping**



## 5.12 Power Management (D31:F0)

### 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 Clock Control
  - ACPI C2 state: Stop-Grant state (using STPCLK# signal) halts processor's instruction stream
- System Sleeping State Control
  - ACPI S1 state: Like C2 state (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

### 5.12.1 ICH2 and System Power States

Table 5-37 shows the power states defined for ICH2-based platforms. The state names generally match the corresponding ACPI states.

**Table 5-37. General Power States for Systems using ICH2**

State/ Substates	Legacy Name / Description
G0/S0/C0	<b>Full On:</b> Processor operating. Individual devices may be shut down to save power. The different processor operating levels are defined by Cx states, as shown in Table 5-38. Within the C0 state, the ICH2 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	<b>Auto-Halt:</b> Processor has executed a AutoHalt instruction and is not executing code. The processor snoops the bus and maintains cache coherency.
G0/S0/C2	<b>Stop-Grant:</b> The STPCLK# signal goes active to the processor. The processor performs a Stop-Grant cycle, halts its instruction stream, and remains in that state until the STPCLK# signal goes inactive. In the Stop-Grant state, the processor snoops the bus and maintains cache coherency.
G1/S1	<b>Stop-Grant:</b> Similar to G0/S0/C2 state. ICH2 also has the option to assert the CPUSLP# signal to further reduce processor power consumption. Note: The behavior for this state is slightly different when supporting iA64 processors.
G1/S3	<b>Suspend-To-RAM (STR):</b> The system context is maintained in system DRAM, but power is shut off to non-critical circuits. Memory is retained and refreshes continue. All clocks stop except RTC clock.

**Table 5-37. General Power States for Systems using ICH2**

State/ Substates	Legacy Name / Description
G1/S4	<b>Suspend-To-Disk (STD):</b> The context of the system is maintained on the disk. All power is then shut off to the system except for the logic required to resume. Externally appears same as S5, but may have different wake events.
G2/S5	<b>Soft Off (SOFF):</b> System context is not maintained. All power is shut off except for the logic required to restart. A full boot is required when waking.
G3	<b>Mechanical OFF (MOFF):</b> 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 depends on the state just prior to the entry to G3 and the AFTERG3 bit in the GEN_PMCN3 register (D31:F0, offset A4). Refer to <a href="#">Table 5-46</a> for more details.

[Table 5-38](#) shows the transitions rules among the various states. Note that transitions among the various states may appear to temporarily transition through intermediate states. For example, in going from S0 to S1, it may appear to pass through the G0/S0/C2 states. These intermediate transitions and states are not listed in the table.

**Table 5-38. State Transition Rules for ICH2**

Present State	Transition Trigger	Next State
G0/S0/C0	<ul style="list-style-type: none"> <li>Processor halt instruction</li> <li>Level 2 Read</li> <li>Level 3 Read</li> <li>SLP_EN bit set</li> <li>Power Button Override</li> <li>Mechanical Off/Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C1</li> <li>G0/S0/C2</li> <li>G0/S0/C3</li> <li>G1/Sx or G2/S5state</li> <li>G2/S5</li> <li>G3</li> </ul>
G0/S0/C1	<ul style="list-style-type: none"> <li>Any Enabled Break Event</li> <li>STPCLK# goes active</li> <li>Power Button Override</li> <li>Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C0</li> <li>G0/S0/C2</li> <li>G2/S5</li> <li>G3</li> </ul>
G0/S0/C2	<ul style="list-style-type: none"> <li>Any Enabled Break Event</li> <li>STPCLK# goes inactive and previously in C1</li> <li>Power Button Override</li> <li>Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C0</li> <li>G0/S0/C1</li> <li>G2/S5</li> <li>G3</li> </ul>
G1/S1, G1/S3, or G1/S4	<ul style="list-style-type: none"> <li>Any Enabled Wake Event</li> <li>Power Button Override</li> <li>Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C0</li> <li>G2/S5</li> <li>G3</li> </ul>
G2/S5	<ul style="list-style-type: none"> <li>Any Enabled Wake Event</li> <li>Power Failure</li> </ul>	<ul style="list-style-type: none"> <li>G0/S0/C0</li> <li>G3</li> </ul>
G3	<ul style="list-style-type: none"> <li>Power Returns</li> </ul>	<ul style="list-style-type: none"> <li>Optional to go to S0/C0 (reboot) or G2/S5 (stay off until power button pressed or other wake event). (See Note 1)</li> </ul>

**NOTES:**

- Some wake events can be preserved through power failure.

## 5.12.2 System Power Planes

The system has several independent power planes, as described in [Table 5-39](#). Note that when a particular power plane is shut off, it should go to a 0V level.

**Table 5-39. System Power Plane**

Plane	Controlled By	Description
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. The processor, devices on the PCI bus, LPC interface, 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_S5# signal	When the SLP_S5# goes active, power can be shut off to any circuit not required to wake the system from the S4 or S5 state. Since the memory context does not need to be preserved in the S5 state, the power to the memory can also be shut down.
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.

## 5.12.3 ICH2 Power Planes

The ICH2 power planes were previously defined in [Section 3.1](#).

Although not specific power planes within the ICH2, there are many interface signals that go to devices that may be powered down. These include:

- IDE: ICH2 can tri-state or drive low all IDE output signals and shut off input buffers.
- USB: ICH2 can tri-state USB output signals and shut off input buffers if USB wakeup is not desired
- AC'97: ICH2 can drive low the outputs and shut off inputs

## 5.12.4 SMI#/SCI Generation

Upon any SMI# event taking place, ICH2 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 PCICLKs. 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 (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 still be routed to IRQ[9, 10, or 11] or it can be instead routed to one of the APIC interrupts 20:23. In the case where the SCI is routed to IRQ[20, 21, 22, or 23], the interrupt generated internally is an active low level. The interrupt remains low until all SCI sources are removed. In the case where the SCI is routed to IRQ[9, 10, or 11], the interrupt generated internally is active high. The interrupt remains high until all SCI sources are removed.

Table 5-40 shows which events can cause an SMI# and SCI. Note that some events can be programmed to cause either an SMI# or SCI. The usage of the event for SCI (instead of SMI#) is typically associated with an ACPI-based system. Each SMI# or SCI source has a corresponding enable and status bit.

**Table 5-40. Causes of SMI# and SCI**

Event	SMI	SCI	Comment
PCI PME# assertion	Yes	Yes	Can also cause Wake Event
PWRBTN# assertion	Yes	Yes	Can also cause Wake Event
RTC Alarm	Yes	Yes	
RI# (Ring Indicate) assertion	Yes	Yes	
AC '97 wake event	Yes	Yes	
USB wake events	Yes	Yes	
THRM# pin active (based on THRM#_POL)	Yes	Yes	The THRM# can cause an SMI# or SCI on either the rising or falling edge. Causes SCI if SCI_EN is set, causes SMI# if SCI_EN not set.
ACPI Timer overflow (2.34s)	Yes	Yes	
Any GPI	Yes	Yes	
TCO event (includes SERR#)	Yes	Yes	Can also cause IRQ (other than SCI).
BIOS_STS bit set	Yes	No	ACPI code in OS sets GBL_RLS bit to cause BIOS_STS bit active, which causes SMI#.
GBL_STS bit set	No	Yes	This bit is set when the BIOS sets the BIOS_RLS bit. The ACPI handler will clear the bit.
Write access to APM control register	Yes	No	OS or BIOS writes to the APMC register. SMM handler clears.
Periodic Timer expires	Yes	No	
SW SMI timer expires	Yes	No	Allows SMM handler to exit temporarily. Another SMI# occurs about 64 ms later.
LEGACY_USB_STS bit set	Yes	No	Bit set based on address decode or incoming USB IRQ.
Serial IRQ SMI	Yes	No	
Device Trap	Yes	No	Indicates that subsystems may need to be powered back on.
SMBus Host Controller	Yes	No	
SMBus Slave SMI	Yes	No	
Global Standby Timer expires in S1 state	Yes	No	
Access to Microcontroller range	Yes	No	Access to Microcontroller range (62h/66h) with MCSMI_EN set.
SLP_EN bit set	Yes	No	

Notes on causes of SCI and SMI:

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 one

## 5.12.5 Dynamic Processor Clock Control

ICH2 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 ICH2 supports the ACPI C0, C1 and C2 states.

The dynamic processor clock control is handled using the following signals:

- STPCLK#: Used to halt processor instruction stream.
- CPUSLP#: Must be asserted prior to STP\_CPU# (in Stop Grant mode)

The C1 state is entered based on the processor performing an autohalt instruction. The C2 state is entered based on the processor reading the Level 2 register in the ICH2.

A C1 or C2 state ends due to a break event. Based on the break event, the ICH2 returns the system to C0 state. [Table 5-41](#) lists the possible break events from C2. The break events from C1 are indicated in the processor's datasheet.

**Table 5-41. Break Events**

Event	Breaks from	Comment
Any unmasked interrupt goes active	C2	IRQ[0:15] when using the 8259s, IRQ[0:23] for I/O APIC. Since SCI is an interrupt, any SCI will also be a break event.
Any internal event that will cause an NMI or SMI#	C2	Many possible sources
Any internal event that will cause INIT# to go active	C2	Could be indicated by the keyboard controller via the RCIN input signal.

### 5.12.5.1 Throttling Using STPCLK#

Throttling is used to lower power consumption or reduce heat. The ICH2 asserts STPCLK# to throttle the processor clock and the processor appears to temporarily enter a C2 state. After a programmable time, the ICH2 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:

- 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.
- 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. Throttling does not occur when the system is in a C2 state, even if Thermal override occurs.

### 5.12.5.2 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).
- If the THTL\_EN bit is set, and a Level 2 read then occurs, the system should immediately go and stay in a C2 state until a break event occurs. A Level 2 read has higher priority than the software initiated throttling or thermal throttling.
- If Thermal Override is causing throttling and a Level 2 read then occurs, the system will stay in a C2 state until a break event occurs. A Level 2 read has higher priority than the Thermal Override.
- After an exit from a C2 state (due to a Break event), and if the THTL\_EN bit is still set, or if a Thermal Override is still occurring, the system will continue to throttle STPCLK#. Depending on the time of the break event, the first transition on STPCLK# active can be delayed by up to one period.
- 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 ICH2 observing the Stop-Grant cycle. This ensures that the STPCLK# signal stays active for a sufficient period after the processor observes the response phase.
- If in the C1 state and the STPCLK# signal goes active, the processor will generate a Stop-Grant cycle, and the system should go to the C2 state. When STPCLK# goes inactive, it should return to the C1 state.

### 5.12.6 Sleep States

The ICH2 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 will disable 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.

### 5.12.6.1 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 will then attempt to gracefully put the system into the corresponding Sleep state by first going to a C2 state. See [Section 5.12.5](#) for details on going to the C2 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 will be less graceful, since there will be no dependencies on observing Stop-Grant cycles from the processor or on clocks other than the RTC clock.

**Table 5-42. Sleep Types**

Sleep Type	Comment
S1	ICH2 asserts the CPUSLP# signal. This lowers the processor's power consumption. No snooping is possible in this state.
S3	ICH2 asserts SLP_S3#. The SLP_S3# signal controls the power to non-critical circuits. Power is only be retained to devices needed to wake from this sleeping state, as well as to the memory.
S4	ICH2 asserts SLP_S3# and SLP_S5#. The SLP_S5# signal shuts off the power to the memory subsystem. Only devices needed to wake from this state should be powered.
S5	Same as S4. ICH2 asserts SLP_S3# and SLP_S5#. The SLP_S5# signal shuts off the power to the memory subsystem. Only devices needed to wake from this state should be powered.

### 5.12.6.2 Exiting Sleep States

Sleep states (S10–S5) are exited based on Wake events. The Wake events will force the system to a full on state (S0), although some non-critical subsystems might still be shut off and have to be brought back manually. For example, the hard disk may be shut off during a sleep state, and have to be enabled via a GPIO pin before it can be used.

Upon exit from the ICH2-controlled Sleep states, the WAK\_STS bit will be set. The possible causes of Wake Events (and their restrictions) are shown in [Table 5-43](#).

**Note:** If in the S5 state due to a powerbutton override, the only wake event is power button.

**Table 5-43. 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
GPIO[0:n]	S1–S5 (Note 1)	GPE1_EN register
USB	S1–S4	Set USB1_EN and USB2_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

**Table 5-43. Causes of Wake Events (Continued)**

Cause	States Can Wake From	How Enabled
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–S4	SMB_WAK_EN in the GPE0 Register
SMBus Slave Message	S1–S5	Always enabled as a Wake Event

**NOTES:**

1. This will be a wake event from S5 only if the sleep state was entered by setting the SLP\_EN and SLP\_TYP bits via software.

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 5-44](#) summarizes the use of GPIs as wake events.

**Table 5-44. GPI Wake Events**

GPI	Power Well	Wake From	Notes
GPI[7:0], GPI[23:16]	Core	S1	
GPI[15:8]	Resume	S1–S5	ACPI Compliant

The latency to exit the various Sleep states varies greatly and is heavily dependent on power supply design. Approximations are shown in [Table 5-45](#). The time indicates from when the Wake event occurs (signal transition) to when the processor is allowed to start its first cycle (CPURST# goes inactive). There will be very large additional delays for the processor to execute sufficient amounts of BIOS to invoke the OS (such as coming out of S1–S3) or spinning up the hard drive (e.g., coming out of S4 or S5).

**Table 5-45. Sleep State Exit Latencies**

State	Latency
S1	<1 ms. Based on wake event to STPCLK# high + re-enumeration of PCI bus, USB, CardBus, etc. Must also add PLL spin-up times.
S3	Power Supply ramp + 20 ms
S4	Power Supply ramp + 20 ms
S5	Power Supply ramp + 20 ms

### 5.12.6.3 Sx–G3–Sx, Handling Power Failures

In desktop systems, 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 can 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.

- **PWRBTN#:** PWRBTN# is always enabled as a wake event. When RSMRST# is low (G3 state), the PWRBTN\_STS bit is reset. When the ICH2 exits G3 after power returns (RSMRST# goes high), the PWRBTN# signal is already high (because Vcc-standby goes high before RSMRST# goes high) and the PWRBTN\_STS bit is 0.
- **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 this as a wake event.
- **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 ICH2 monitors both PWROK and RSMRST# to detect 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 and PME\_STS bits are cleared by RSMRST#

**Table 5-46. Transitions Due To Power Failure**

State at Power Failure	AFTERG3_EN bit	Transition When Power Returns
S0, S1, S3	1 0	S5 S0
S4	1 0	S4 S0
S5	1 0	S5 S0

## 5.12.7 Thermal Management

The ICH2 has mechanisms to assist with managing thermal problems in the system.

### 5.12.7.1 THRM# Signal

The THRM# signal is used as a status input for a thermal sensor. Based on the THRM# signal going active, the ICH2 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 is 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.

### 5.12.7.2 THRM# Initiated Passive Cooling

If the THRM# signal remains active for some time greater than 2 seconds and the ICH2 is in the S0/G0/C0 state, then the ICH2 enters an auto-throttling mode, in which it provides a duty cycle on the STPCLK# signal. This will reduce 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 will result 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 will depend on the instruction stream, because the processor is allowed to finish the current instruction. Furthermore, the ICH2 waits for the STOP-GRANT cycle before starting the count of the time the STPCLK# signal is active.

When THRM# goes inactive, throttling stops. In case that the ICH2 is already attempting throttling because the THTL\_EN bit is set, the duty cycle associated with the THRM# signal will have higher priority. If the ICH2 is in the C2 or S1–S5 states, then no throttling will be caused by the THRM# signal being active.

### 5.12.7.3 THRM# Override Software Bit

The FORCE\_THTL bit allows BIOS to force passive cooling, just as if the THRM# signal had been active for 2 seconds. If this bit is set, the ICH2 starts throttling using the ratio in the THRM\_DTY field.

When this bit is cleared, the ICH2 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).

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

Using the THTL\_EN and THTL\_DTY bits, the ICH2 can force a programmed duty cycle on the STPCLK# signal. This reduces the effective instruction rate of the processor and cut its power consumption and heat generation.

#### 5.12.7.5 Active Cooling

Active cooling involves fans. The GPIO signals from the ICH2 can be used to turn on/off a fan.

### 5.12.8 Event Input Signals and Their Usage

The ICH2 has various input signals that trigger specific events. This section describes those signals and how they should be used.

#### 5.12.8.1 PWRBTN# — Power Button

The ICH2 PWRBTN# signal operates as a “Fixed Power Button” as described in the ACPI specification. PWRBTN# signal has a 16 ms de-bounce on the input. The state transition descriptions are included in the following table. 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.

**Table 5-47. 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 will typically initiate 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. Not latched nor detected.
S0–S4	PWRBTN# held low for at least 4 consecutive seconds	Unconditional transition to S5 state.	No dependence on processor (such as Stop-Grant cycles) or any other subsystem.

#### Power Button Override Function

If PWRBTN# is observed active for at least 4 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.

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 ICH2 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.

## Sleep Button

The ACPI specification 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 ICH2 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 ACPI specification for implementation details.

### 5.12.8.2 RI# — Ring Indicate

The Ring Indicator can cause a wake event (if enabled) from the S1–S5 states. [Table 5-48](#) shows when the wake event is generated or ignored in different states. If in the G0/S0/Cx states, the ICH2 generates an interrupt based on RI# active and the interrupt is set up as a break event.

**Table 5-48. Transitions Due to RI# signal**

Present State	Event	RI_EN	Event
S0	RI# Active	X	Ignored
S1–S5	RI# Active	0	Ignored
		1	Wake Event

**Note:** Filtering/Debounce on RI# will not be done in ICH2. Can be in modem or external.

### 5.12.8.3 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.

## 5.12.9 Alt Access Mode

Before entering a low power state, several registers from powered down parts may need to be saved. In the majority of cases, this is not an issue, as registers have read and write paths. However, several of the ISA compatible registers are either read only or write only. To get data out of write-only registers and to restore data into read-only registers, the ICH2 implements an alternate access mode.

### 5.12.9.1 Write Only Registers with Read Paths in Alternate Access Mode

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

**Note:** If the processor issues a locked cycle to a resource that is too slow (e.g., PCI), the ICH2 will not allow upstream requests to be performed until the cycle completion. 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 ICH2 is not able to issue a request for the next data. Snooper 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 controller. 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.

**Table 5-49. Write Only Registers with Read Paths in Alternate Access Mode**

Restore Data				Restore Data			
I/O Addr	# of Rds	Access	Data	I/O Addr	# of Rds	Access	Data
00h	2	1	DMA Chan 0 base address low byte	40h	7	1	Timer Counter 0 status, bits [5:0]
		2	DMA Chan 0 base address high byte			2	Timer Counter 0 base count low byte
01h	2	1	DMA Chan 0 base count low byte			3	Timer Counter 0 base count high byte
		2	DMA Chan 0 base count high byte			4	Timer Counter 1 base count low byte
02h	2	1	DMA Chan 1 base address low byte			5	Timer Counter 1 base count high byte
		2	DMA Chan 1 base address high byte			6	Timer Counter 2 base count low byte
03h	2	1	DMA Chan 1 base count low byte			7	Timer Counter 2 base count high byte
		2	DMA Chan 1 base count high byte	41h	1		Timer Counter 1 status, bits [5:0]
04h	2	1	DMA Chan 2 base address low byte	42h	1		Timer Counter 2 status, bits [5:0]
		2	DMA Chan 2 base address high byte	70h	1		Bit 7 = NMI Enable, Bits [6:0] = RTC Address

Table 5-49. Write Only Registers with Read Paths in Alternate Access Mode (Continued)

Restore Data				Restore Data			
I/O Addr	# of Rds	Access	Data	I/O Addr	# of Rds	Access	Data
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 <sup>2</sup>	CAh	2	1	DMA Chan 6 base count low byte
		2	DMA Chan 0–3 Request			2	DMA Chan 6 base count high byte
		3	DMA Chan 0 Mode: Bits(1:0) = “00”	CCh	2	1	DMA Chan 7 base address low byte
		4	DMA Chan 1 Mode: Bits(1:0) = “01”			2	DMA Chan 7 base address high byte
		5	DMA Chan 2 Mode: Bits(1:0) = “10”	CEh	2	1	DMA Chan 7 base count low byte
		6	DMA Chan 3 Mode: Bits(1:0) = “11”.			2	DMA Chan 7 base count high byte
20h	12	1	PIC ICW2 of Master controller	D0h	6	1	DMA Chan 4–7 Command <sup>2</sup>
		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 <sup>1</sup>			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 <sup>1</sup>				
		11	PIC OCW2 of Slave controller				
		12	PIC OCW3 of Slave controller				

**NOTE:**

1. The OCW1 register must be read before entering Alternate Access Mode.
2. Bits 5, 3, 1, and 0 return 0.

### 5.12.9.2 PIC Reserved Bits

Many bits within the PIC are reserved, and must have certain values written for the PIC to operate properly. Therefore, there is no need to return these values in alternate access mode. When reading PIC registers from 20h and A0h, the reserved bits shall return the values listed in [Table 5-50](#).

**Table 5-50. 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

### 5.12.9.3 Read Only Registers with Write Paths in Alternate Access Mode

The registers described in [Table 5-51](#) have write paths alternate access mode. Software restores these values after returning from a powered down state. These registers must be handled specially by software. When in normal mode, writing to the Base Address and Count Register also writes to the Current Address and Count Register. Therefore, the Base Address and Count must be written first, then the part is put into alternate access mode and the Current Address and Count Register is written.

**Table 5-51. Register Write Accesses in Alternate Access Mode**

I/O Address	Register Write Value
08h	DMA Status Register for channels 0–3.
D0h	DMA Status Register for channels 4–7.

## 5.12.10 System Power Supplies, Planes, and Signals

### Power Plane Control with SLP\_S3# and SLP\_S5#

The SLP\_S3# output signal can be used to cut power to the system core supply, since it will only go active for the STR state (typically mapped to ACPI S3). Power must be maintained to the ICH2 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\_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.

### PWROK Signal

The PWROK input should go active based on the core supply voltages becoming valid. PWROK should go active no sooner than 10 ms after Vcc3\_3 and VCC1\_8 have reached their nominal values.

**Note:** Traditional designs have a reset button logically AND'd with the PWROK signal from the power supply and the processor's voltage regulator module. If this is done with the ICH2, the PWROK\_FLR bit will be set. The ICH2 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), the ICH2 reboots (regardless of the state of the AFTERG3 bit). If the RSMRST# signal also goes low before PWROK goes high, this is a full power failure and the reboot policy is controlled by the AFTERG3 bit.

### 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.

### Controlling Leakage and Power Consumption During Low-Power States

To control leakage in the system, various signals will 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 C2 or S3 state, the processor signals that have pull-ups will be tri-stated or driven low.
- During S3 (STR), all signals attached to powered down planes will be tri-stated or driven low.

## 5.12.11 Clock Generators

The clock generator is expected to provide the frequencies shown in [Table 5-52](#).

**Table 5-52. ICH2 Clock Inputs**

Clock Domain	Frequency	Source	Usage
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 ICH2. 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'97 Link. Control policy is determined by the clock source.
APICCLK	16.67 MHz or 33 MHz	Main Clock Generator	Used for ICH2-processor interrupt messages. Should be running in C0, C1 and C2. Stopped in S3 ~ S5 based on SLP_S3# assertion.
LAN_CLK	0.8 to 50 MHz	LAN Connect	LAN Connect link. Control policy is determined by the clock source.

## 5.12.12 Legacy Power Management Theory of Operation

Instead of relying on ACPI software, legacy power management uses BIOS and various hardware mechanisms. ICH2 has a greatly simplified method for legacy power management compared with previous generations (e.g., PIIX4).

The scheme relies on the concept of detecting when individual subsystems are idle, detecting when the whole system is idle, and detecting when accesses are attempted to idle subsystems.

However, the operating system is assumed to be at least APM enabled. Without APM calls, there is no quick way to know when the system is idle between keystrokes. The ICH2 does not support the burst modes found in previous components (e.g., PIIX4).

### Desktop APM Power Management

The ICH2 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 are 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.

## 5.13 System Management (D31:F0)

The ICH2 provides various functions to make a system easier to manage and to lower the Total Cost of Ownership (TCO) of the system. Features and functions can be augmented via external A/D converters and GPIO, as well as an external microcontroller. The following features and functions are supported by the ICH2:

- Processor present detection.
  - Detects if processor fails to fetch the first instruction after reset.
- Various Error detection (e.g., 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 FWH programming
  - Detects if data on first read is FFh (indicates unprogrammed FWH)

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

### 5.13.1 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.

#### 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 ICH2 asserts PCIRST#.

#### Handling an Intruder

The ICH2 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 2 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 ICH2 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.

**Note:** The INTRD\_DET bit resides in the ICH2'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 2 RTC clocks (about 65  $\mu$ s) 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 will remain set and the SMI will be 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.

## Detecting Improper FWH Programming

The ICH2 can detect the case where the FWH is not programmed. This results in the first instruction fetched to have a value of FFh. If this occurs, the ICH2 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 5.13.2](#)).

## 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.

### 5.13.2 Alert on LAN\*

The ICH2 integrated LAN controller supports Alert on LAN\* functionality when used with the 82562EM 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.

The ICH2 also features an independent, dedicated SMBus interface, referred to as the SMLINK interface that can be used with an external Alert on LAN\* (or Alert on LAN 2\*) enabled LAN Controller. This separate interface is required, since devices on the system SMBus will be powered down during some low power states.

The basic scheme is for the ICH2 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 ICH2.

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 ICH2 sends a new message and increments the SEQ[3:0] field. For heartbeat messages, the sequence number does not increment.

If the policy is for the ICH2 to reboot the system after a hardware lockup, the ICH2 does not immediately send an Alert on LAN\* message. It first attempts to reboot the processor and let the BIOS perform the appropriate recovery (and potentially send the message). However, if the boot fails, the ICH2 sends the message.

If the policy is for the ICH2 not to reboot after a hardware lockup, the ICH2 sends an Alert on LAN\* message with the Watchdog (WD) Event Status bit set. This message is sent as soon as the lockup is detected. The message is sent with the next incremented sequence number. If a system is locked, the ICH2 continues sending the Alert on LAN\* messages every heartbeat period (30–32 seconds) unless one of the following occurs:

- The system is suspended via a PowerButton Override.
- The NO\_REBOOT bit (D31:F0, offset D4h, bit 1) is set and the system is reset using PWROK, or the system is reset remotely by SMLINK SMBus Slave write and BIOS clears the SECOND\_TO\_STS bit before a TCO timeout can occur.
- The NO\_REBOOT bit (D31:F0, offset D4h, bit 1) is not set causing the system to reboot automatically.

If another event occurs prior to a power button override, the ICH2 will send another Alert on LAN\* message with the next incremented sequence number and appropriate status bit set.

If a boot is unsuccessful (processor does not fetch the first instruction), then the ICH2 will send an Alert on LAN\* message with the processor event status bit set and the next incremented sequence number. This message will be sent as soon as the lockup is detected (2 TCO timer time-outs).

If the system is in a G1 (S1–S4) state the ICH2 will send a heartbeat message every 30–32 seconds. If an event occurs prior to the system being shutdown, the ICH2 immediately sends an event message with the next incremented sequence number. After the event message, the ICH2 resumes sending heartbeat messages.

**Note:** Normally, the ICH2 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).

When used with an external Alert on LAN\* enabled LAN controller, the ICH2 sends these messages via the SMLINK signals. When sending messages via these signals, the ICH2 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 ICH2 waits until the bus is idle and tries again. Table 5-53 shows the data included in the Alert on LAN\* messages.

**Table 5-53. Alert on LAN\* Message Data**

Field	Comment
Cover Tamper Status	1 = This bit will be set if the intruder detect bit is set (INTRD_DET).
Temp Event Status	1 = This bit will be set if the ICH2THERM# input signal is asserted.
Processor Missing Event Status	1 = This bit will be set if the processor failed to fetch the 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 FWH Event Status	1 = First BIOS fetch returned a value of FFh, indicating that the FWH has not yet been programmed (still erased).
GPIO Status	1 = This bit is set when GPIO[11] signal is high. 0 = This bit is cleared when GPIO[11] signal is low.
SEQ[3:0]	This is a sequence number. It will initially be 0, and will increment each time the ICH2 sends a new message. Upon reaching 1111, then the sequence number will roll over to 0000. MSB (SEQ3) sent first.
System Power State	00 = G0, 01 = G1, 10 = G2, 11 = Pre-Boot. MSB sent first
MESSAGE1	Same as the MESSAGE1 Register. MSB sent first.
MESSAGE2	Same as the MESSAGE2 Register. MSB sent first.
WDSTATUS	Same as the WDSTATUS Register. MSB sent first.

## 5.14 General Purpose I/O

### 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 ICH2 GPIOs may be connected to pins on devices that exist in the core well. If these GPIOs are outputs, there is a danger that a loss of core power (PWROK low) or a Power Button Override event will result in the ICH2 driving a pin to a logic 1 to another device that is powered down.

### SMI# and SCI Routing

The routing bits for GPIO[13:11,8:6,4:3,1:0] 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.

### Power Wells

GPIO[13:11,8:6,4:3,1] have "sticky" bits on the input. Refer to the GPE1\_STS register. As long as the signal goes active for at least 2 clocks, the ICH2 will keep 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 IDE Controller (D31:F1)

The ICH2 IDE controller features two sets of interface signals (Primary and Secondary) that can be independently enabled, tri-stated or driven low.

The IDE interfaces of the ICH2 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 ICH2. 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.

### 5.15.1 PIO Transfers

The ICH2 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 2 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.

#### 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 IDETIM Register and may be set to 2, 3, 4, or 5 PCI clocks. The recovery time is selected by the IDETIM 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 2 PCI clocks in duration.

The IDE timings for various transaction types are shown in [Table 5-54](#). 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 5-54. 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

### 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.

### 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 will be deasserted for at least 2 PCI clocks between the 2 cycles.

### PIO IDE Data Port Prefetching and Posting

The ICH2 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 must 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 ICH2. The ICH2 then runs the IDE cycle to transfer the data to the drive. If the ICH2 write buffer is non-empty and an unrelated (non-data or opposite channel) IDE transaction occurs, that transaction is stalled until all current data in the write buffer is transferred to the drive.

## 5.15.2 Bus Master Function

The ICH2 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 ICH2 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.

### Physical Region Descriptor Format

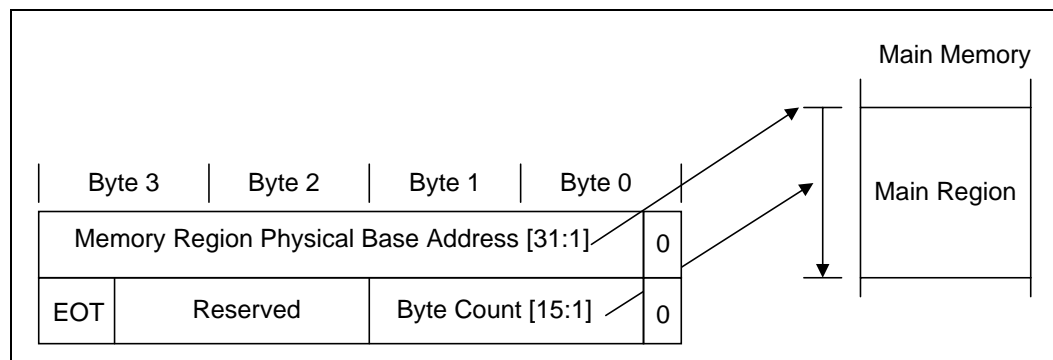
The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The PRDs are stored sequentially in a Descriptor Table in memory. The data transfer proceeds until all regions described by the PRDs in the table have been transferred. Note that the ICH2 bus master IDE function does not support memory regions or Descriptor tables located on ISA.

Descriptor Tables must be aligned on 64 KB boundaries. 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 KB boundary. The next two bytes specify the size or transfer count of the region in bytes (64 KB limit per region). A value of zero in these two bytes indicates 64 KB (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, causes 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 5-14. Physical Region Descriptor Table Entry



## Line Buffer

A single line buffer exists for the ICH2 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 ICH2 are retried until all data in the line buffers has been transferred to memory.

## 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 than 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.

## Interrupts

The ICH2 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 is operating under an interrupt based driver. Therefore, it does not operate under environments where the IDE device drives an interrupt but the interrupt is masked in the system.
- Bus Master IDE devices are connected directly off of ICH2. IDE interrupts cannot be communicated through PCI devices or the serial stream.

## 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 masked.

The buffer is flushed (when in the write state) or invalidated (when in the read state) when a terminal count condition exists (i.e., 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 5-55](#) 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 ICH2 IDE interface arbitrates between the primary and secondary IDE cables when a PRD expires.

**Table 5-55. 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.

## 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 zero 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 stops the transfer (i.e., reset the Active bit in the Command register) and sets 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.

When 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.

## 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 ICH2 does not use the 8237 for this mode.

### 5.15.3 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 such as the ICH2 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.

#### 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 5-56](#). Read cycles are defined as transferring data from the IDE device to the ICH2. Write cycles are defined as transferring data from ICH2 to IDE device.

**Table 5-56. UltraATA/33 Control Signal Redefinitions**

Standard IDE Signal Definition	Ultra ATA/33 Read Cycle Definition	Ultra ATA/33 Write Cycle Definition	ICH2 Primary Channel Signal	ICH2 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 ICH2 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 ICH2 (read). It is used by the ICH2 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 ICH2 to the IDE device (write). It is the data strobe signal driven by the ICH2 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 ICH2 (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 ICH2 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.

## Operation

Initial setup programming consists of enabling and performing the proper configuration of ICH2 and the IDE device for Ultra ATA/33 operation. For ICH2, 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 ICH2 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 ICH2 asserts the DMACK# signal. When DMACK# signal is asserted, the host controller drives CS0# and CS1# inactive, DA0–DA2 low. For write cycles, the ICH2 deasserts STOP, waits for the IDE device to assert DMARDY#, and then drives the first data word and STROBE signal. For read cycles, the ICH2 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 (ICH2 - 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 ICH2 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 ICH2 can stop a burst by asserting STOP; the IDE device acknowledges by deasserting DMARQ. The IDE device stops a burst by deasserting DMARQ and the ICH2 acknowledges by asserting STOP. The transmitter then drives the STROBE signal to a high level. The ICH2 then drives the CRC value on the DD lines and deasserts DMACK#. The IDE device latches the CRC value on the rising edge of DMACK#. The ICH2 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.

## 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 ICH2 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 ICH2 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 ICH2 CRC value to its own and reports an error if there is a mismatch.

#### 5.15.4 Ultra ATA/66 Protocol

In addition to Ultra ATA/33, the ICH2 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 will still plug into the standard IDE connector. The Ultra ATA/66 protocol also supports a 44 MB/s mode.

#### 5.15.5 Ultra ATA/100 Protocol

When the ATA\_FAST bit is set for any of the 4 IDE devices, the timings for the transfers to and from the corresponding device run at a higher rate. The ICH2 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.

#### 5.15.6 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 ICH2 will wait 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 3 Base Clocks. The ICH2, thus, toggles the write strobe signal every 22.5 ns, transferring two bytes of data on each strobe edge. This means that the ICH2 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; the ICH2 supports reads at the maximum rate of 100 MB/s.

## 5.16 USB Controller (Device 31:Functions 2 and 4)

The ICH2 contains two USB Host Controllers. Each Host Controller includes a root hub with two separate USB ports each, for a total of 4 USB ports. The ICH2 Host Controllers support the standard Universal Host Controller Interface (UHCI) Rev 1.1.

Overcurrent detection on all 4 USB ports is supported. The overcurrent inputs are 5V-tolerant, and can be used as GPIs if not needed.

The ICH2's USB controllers are arbitrated as differently than standard PCI devices to improve arbitration latency.

### 5.16.1 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 ICH2: 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.

#### 5.16.1.1 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 5-57](#).

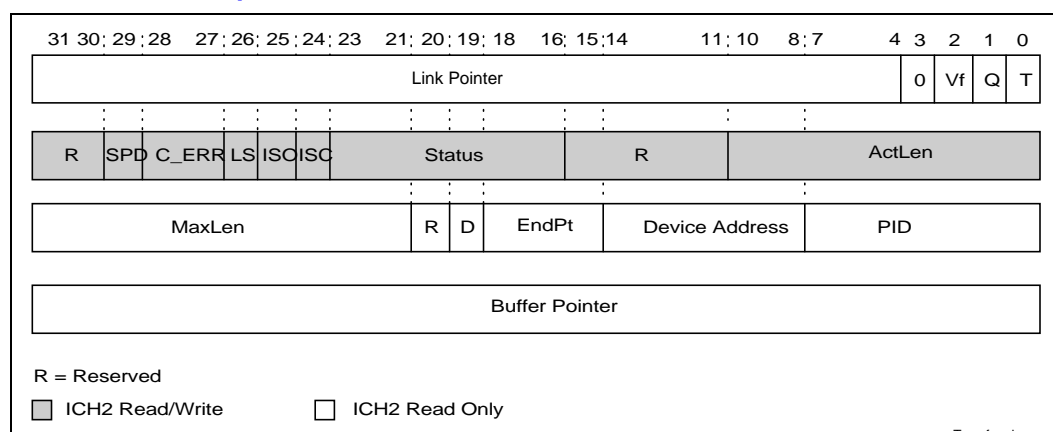
**Table 5-57. Frame List Pointer Bit Description**

Bit	Description
31:4	<b>Frame List Pointer (FLP).</b> 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	<b>QH/TD Select (Q).</b> 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 ICH2 to perform the proper type of processing on the item after it is fetched. 1 = QH 0 = TD
0	<b>Terminate (T).</b> This bit indicates to the ICH2 whether the schedule for this frame has valid entries in it. 1 = Empty Frame (pointer is invalid). 0 = Pointer is valid (points to a QH or TD).

### 5.16.1.2 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 5-15](#). The 4 different USB transfer types are supported by a small number of control bits in the descriptor that the ICH2 interprets during operation. All Transfer Descriptors have the same basic, 32-byte structure. During operation, the ICH2 hardware performs consistency checks on some fields of the TD. If a consistency check fails, the ICH2 halts immediately and issues an interrupt to the system. This interrupt cannot be masked within the ICH2.

**Figure 5-15. Transfer Descriptor**



**Table 5-58. TD Link Pointer**

Bit	Description
31:4	<b>Link Pointer (LP).</b> 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	<b>Depth/Breadth Select (VF).</b> 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 ICH2 to process the next transaction in the queue rather than starting a new queue. 1 = Depth first. 0 = Breadth first.
1	<b>QH/TD Select (Q).</b> This bit informs the ICH2 whether the item referenced by the link pointer is another TD or a QH. This allows the ICH2 to perform the proper type of processing on the item after it is fetched 1 = QH. 0 = TD.
0	<b>Terminate (T).</b> This bit informs the ICH2 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 ICH2 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 ICH2 that this is the last TD in the frame. 1 = Link Pointer field not valid. 0 = Link Pointer field is valid.

Table 5-59. TD Control and Status

Bit	Description																														
31:30	Reserved.																														
29	<p><b>Short Packet Detect (SPD).</b> 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 will be sent at the end of the frame.</p> <p>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.</p> <p>1 = Enable. 0 = Disable.</p>																														
28:27	<p><b>Error Counter (C_ERR).</b> 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 ICH2 decrements the count and writes it back to the TD if the transaction fails. If the counter counts from one to zero, the ICH2 marks the TD inactive, sets the “STALLED” and error status bit for the error that caused the transition to zero in the TD. An interrupt will be generated to Host Controller Driver (HCD) if the decrement to zero was caused by Data Buffer error, Bit stuff error, or if enabled, a CRC or Timeout error. If HCD programs this field to zero during setup, the ICH2 will not count errors for this TD and there will be no limit on the retries of this TD.</p> <table><tr><th>Bits[28:27]</th><th>Interrupt After</th></tr><tr><td>00</td><td>No Error Limit</td></tr><tr><td>01</td><td>1 Error</td></tr><tr><td>10</td><td>2 Errors</td></tr><tr><td>11</td><td>3 Errors</td></tr></table> <table><tr><th>Error</th><th>Decrement Counter</th><th>Error</th><th>Decrement Counter</th></tr><tr><td>CRC Error</td><td>Yes</td><td>Data Buffer Error</td><td>Yes</td></tr><tr><td>Timeout Error</td><td>Yes</td><td>Stalled</td><td>No*</td></tr><tr><td>NAK Received</td><td>No</td><td>Bit stuff Error</td><td>Yes</td></tr><tr><td>Babble Detected</td><td>No*</td><td></td><td></td></tr></table> <p>*Detection of Babble or Stall automatically deactivates the TD. Thus, count is not decremented. * Detection of Babble or Stall automatically deactivates the TD. Thus, count is not decremented.</p>	Bits[28:27]	Interrupt After	00	No Error Limit	01	1 Error	10	2 Errors	11	3 Errors	Error	Decrement Counter	Error	Decrement Counter	CRC Error	Yes	Data Buffer Error	Yes	Timeout Error	Yes	Stalled	No*	NAK Received	No	Bit stuff Error	Yes	Babble Detected	No*		
Bits[28:27]	Interrupt After																														
00	No Error Limit																														
01	1 Error																														
10	2 Errors																														
11	3 Errors																														
Error	Decrement Counter	Error	Decrement Counter																												
CRC Error	Yes	Data Buffer Error	Yes																												
Timeout Error	Yes	Stalled	No*																												
NAK Received	No	Bit stuff Error	Yes																												
Babble Detected	No*																														
26	<p><b>Low Speed Device (LS).</b> 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 ICH2 root hub port is connected to a full speed device and this bit is set to a 1 for a low speed transaction, the ICH2 sends out a low speed preamble on that port before sending the PID. No preamble is sent if a ICH2 root hub port is connected to a low speed device.</p> <p>1 = Low Speed Device 0 = Full Speed Device</p>																														
25	<p><b>Isochronous Select (IOS).</b> 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.</p> <p>1 = Isochronous Transfer Descriptor 0 = Non-isochronous Transfer Descriptor</p>																														
24	<p><b>Interrupt on Complete (IOC).</b> This specifies that the ICH2 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.</p> <p>1 = Issue IOC</p>																														
23	<p><b>Active.</b> For ICH2 schedule execution operations, see the Data Transfers To/From Main Memory section.</p> <p>1 = Set to 1 by software to enable the execution of a message transaction by the ICH2. 0 = When the transaction associated with this descriptor is completed, the ICH2 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.</p>																														

Table 5-59. TD Control and Status (Continued)

Bit	Description
22	<b>Stalled.</b> 1 = Set to a 1 by the ICH2 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 zero, 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	<b>Data Buffer Error (DBE).</b> 1 = Set to a 1 by the ICH2 during status update to indicate that the ICH2 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 will not match. In the case of an underrun, the ICH2 transmits an incorrect CRC (thus invalidating the data at the endpoint) and leaves the TD active (unless error count reached zero). If a overrun condition occurs, the ICH2 forces a timeout condition on the USB, invalidating the transaction at the source.
20	<b>Babble Detected (BABD).</b> 1 = Set to a 1 by the ICH2 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 ICH2 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	<b>Negative Acknowledgment (NAK) Received (NAKR).</b> 1 = Set to a 1 by the ICH2 during status update when the ICH2 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 is also be reported.
18	<b>CRC/Time Out Error (CRC_TOUT).</b> 1 = Set to a 1 by the ICH2 as follows: 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 USB specification. During a status update when a Cyclical Redundancy Check (CRC) error is detected during the transaction associated with this transfer descriptor. In the transmit case (OUT or SETUP Command), this is in response to the ICH2 detecting a timeout from the target device/endpoint. In the receive case (IN Command), this is in response to the ICH2'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	<b>Bit stuff Error (BSE).</b> 1 = This bit is set to a 1 by the ICH2 during status update to indicate that the receive data stream contained a sequence of more than 6 ones in a row.
16	<b>Bus Turn Around Time-out (BTTO).</b> 1 = This bit is set to a 1 by the ICH2 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	<b>Actual Length (ACTLEN).</b> The Actual Length field is written by the ICH2 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 5-60. TD Token

Bit	Description
31:21	<p><b>Maximum Length (MAXLEN).</b> 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 USB specification is 1023 bytes. The valid encodings for this field are:</p> <p>0x000 = 1 byte  0x001 = 2 bytes  ....  0x3FE = 1023 bytes  0x3FF = 1024 bytes  ....  0x4FF = 1280 bytes  0x7FF = 0 bytes (null data packet)</p> <p>Note that values from 500h to 7FEh are illegal and cause a consistency check failure.</p> <p>In the transmit case, the ICH2 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 ICH2 may be unable to access memory (e.g., due to excessive latency) in time to avoid underrunning the transmitter. In this instance the ICH2 would transmit fewer bytes than specified in the Maximum Length field.</p>
20	Reserved.
19	<p><b>Data Toggle (D).</b> 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.</p>
18:15	<p><b>Endpoint (ENDPT).</b> 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.</p>
14:8	<p><b>Device Address.</b> This field identifies the specific device serving as the data source or sink.</p>
7:0	<p><b>Packet Identification (PID).</b> 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 ICH2. Bits [3:0] are complements of bits [7:4].</p>

Table 5-61. TD Buffer Pointer

Bit	Description
31:0	<p><b>Buffer Pointer (BUFF_PNT).</b> 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 in the TD Token. The data buffer may be byte-aligned.</p>

### 5.16.1.3 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; the elements are shown in [Table 5-62](#).

**Table 5-62. Queue Head Block**

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

**Table 5-63. Queue Head Link Pointer**

Bit	Description
31:4	<b>Queue Head Link Pointer (QHLP).</b> 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	<b>QH/TD Select (Q).</b> This bit indicates to the hardware whether the item referenced by the link pointer is another TD or a QH. 1=QH 0=TD
0	<b>Terminate (T).</b> This bit indicates to the ICH2 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. 1 = Last QH (pointer is invalid). 0 = Pointer is valid (points to a QH or TD).

**Table 5-64. Queue Element Link Pointer**

Bit	Description
31:4	<b>Queue Element Link Pointer (QELP).</b> 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	<b>QH/TD Select (Q).</b> 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. 1 = QH 0 = TD
0	<b>Terminate (T).</b> This bit indicates to the ICH2 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. 1 = Terminate (No valid queue entries). 0 = Pointer is valid.

## 5.16.2 Data Transfers To/From Main Memory

The following sections describe the details on how HCD and the ICH2 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.

### 5.16.2.1 Executing the Schedule

Software programs the ICH2 with the starting address of the Frame List and the Frame List index, then causes the ICH2 to execute the schedule by setting the Run/Stop bit in the Control register to Run. The ICH2 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 ICH2 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 ICH2 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 ICH2 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 ICH2 fetches the next entry from the Frame List. If the ICH2 is not able to process all of the transfer descriptors during a given frame, those descriptors are retired by software without having been executed.

### 5.16.2.2 Processing Transfer Descriptors

The ICH2 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. ICH2 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
  - [PCI 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 [*PCI 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.

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

**Table 5-65. Command Register, Status Register and TD Status Bit Interaction**

Condition	ICH2 USB Status Register Actions	TD Status Register Actions
CRC/Time Out Error	Set USB Error Int bit <sup>1</sup> , Clear HC Halted bit	Clear Active bit <sup>1</sup> and set Stall bit <sup>1</sup>
Illegal PID, PID Error, Max Length (illegal)	Clear Run/Stop bit in command register Set HC Process Error and HC Halted bits	
PCI Master/Target Abort	Clear Run/Stop bit in command register Set Host System Error and HC Halted bits	
Suspend Mode	Clear Run/Stop bit in command register <sup>2</sup> Set HC Halted bit	
Resume Received and Suspend Mode = 1	Set Resume received bit	
Run/Stop = 0	Clear Run/Stop bit in command register Set HC Halted bit	
configuration Flag Set	Set configuration Flag in command register	
HC Reset/Global Reset	Clear Run/Stop and configuration Flag in command register 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 <sup>1</sup> and set Stall bit
Bit Stuff/Data Buffer Error	Set USB Error Int bit <sup>1</sup>	Clear Active bit <sup>1</sup> and set Stall bit <sup>1</sup>
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 will be 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 is reset to 0 and Stalled bit to 1 as normal.

### 5.16.2.4 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).

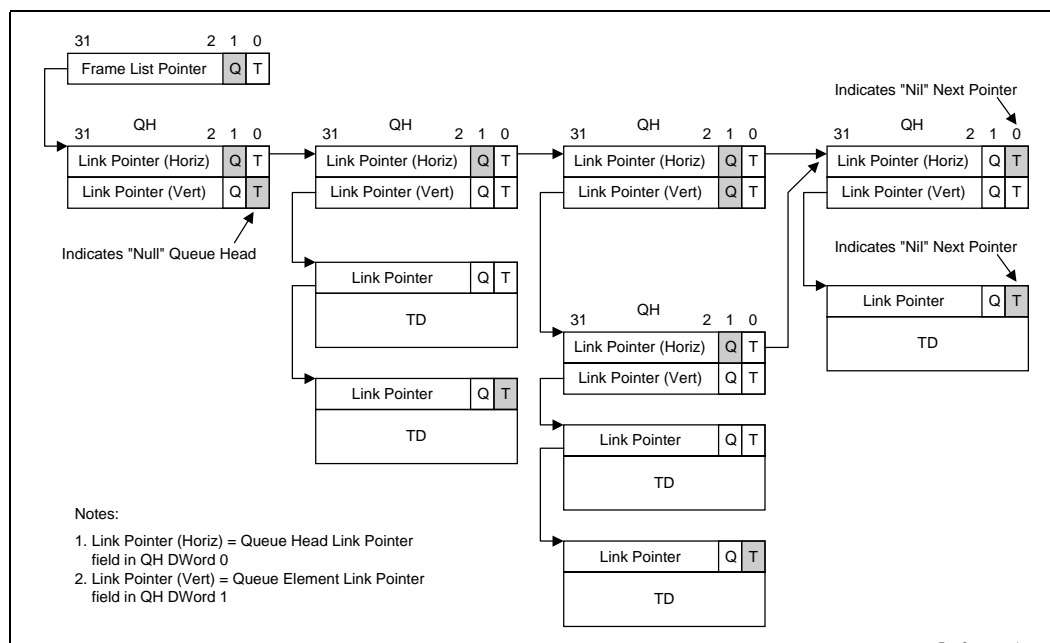
The QH contains two link pointers and is organized as two contiguous DWords. The first DWord is a horizontal pointer (Queue Head Link Pointer), used to link a single transfer queue with either another transfer queue, or a TD (target data structure depends on Q bit). If the T bit is set, this QH represents the last data structure in the current Frame. The T bit informs the ICH2 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 5-16 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 5-16 (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 (ICH2 does 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 ICH2 assumes there is no more work to complete for the current Frame.

**Figure 5-16. 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 ICH2 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 5-66](#), the ICH2 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 ICH2 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 ICH2 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 ICH2 is currently processing a queue, and the advance criteria are met, and the Vf bit is set, the ICH2 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 ICH2 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.

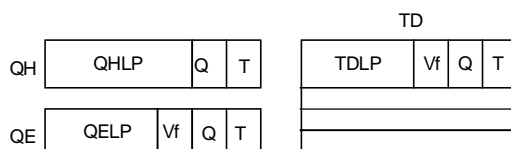
A queue is **never** advanced on an error completion status (even in the event the error count was exhausted).

[Table 5-66](#) 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 5-66. 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 5-67 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 5-67. 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	<ul style="list-style-type: none"> <li>Not in Queue - execute TD.</li> <li>Use TD.LP to get next TD</li> </ul>
0	-	-	-	-	x	x	1	<ul style="list-style-type: none"> <li>Not in Queue - execute TD. End of Frame</li> </ul>
0	-	-	-	-	x	1	0	<ul style="list-style-type: none"> <li>Not in Queue - execute TD.</li> <li>Use TD.LP to get next (QH+QE).</li> <li>Set Q Context to 1.</li> </ul>
1	0	0	0	0	0	x	x	<ul style="list-style-type: none"> <li>In Queue. Use QE.LP to get TD.</li> <li>Execute TD. Update QE.LP with TD.LP.</li> <li>Use QH.LP to get next TD.</li> </ul>
1	x	x	0	0	1	0	0	<ul style="list-style-type: none"> <li>In Queue. Use QE.LP to get TD.</li> <li>Execute TD. Update QE.LP with TD.LP.</li> <li>Use TD.LP to get next TD.</li> </ul>
1	x	x	0	0	1	1	0	<ul style="list-style-type: none"> <li>In Queue. Use QE.LP to get TD.</li> <li>execute TD. Update QE.LP with TD.LP.</li> <li>Use TD.LP to get next (QH+QE).</li> </ul>
1	0	0	x	1	x	x	x	<ul style="list-style-type: none"> <li>In Queue. Empty queue.</li> <li>Use QH.LP to get next TD</li> </ul>
1	x	x	1	0	-	-	-	<ul style="list-style-type: none"> <li>In Queue. Use QE.LP to get (QH+QE)</li> </ul>
1	x	1	0	0	0	x	x	<ul style="list-style-type: none"> <li>In Queue. Use QE.LP to get TD.</li> <li>execute TD. Update QE.LP with TD.LP.</li> <li>End of Frame</li> </ul>
1	x	1	x	1	x	x	x	<ul style="list-style-type: none"> <li>In Queue. Empty queue. End of Frame</li> </ul>

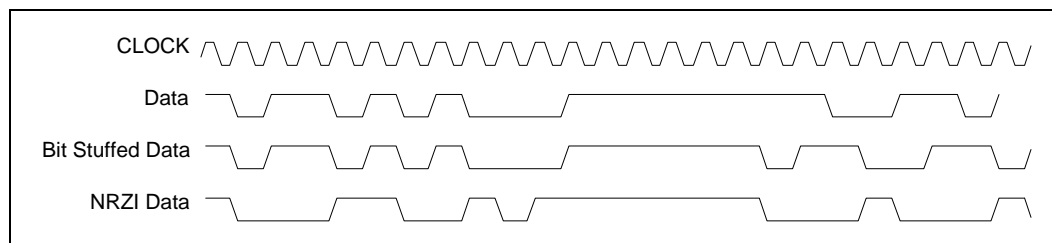
**Table 5-67. USB Schedule List Traversal Decision Table (Continued)**

Q Context	QH.Q	QH.T	QE.Q	QE.T	TD.Vf	TD.Q	TD.T	Description
1	1	0	0	0	0	x	x	<ul style="list-style-type: none"> <li>• In Queue. Use QE.LP to get TD.</li> <li>• execute TD. Update QE.LP with TD.LP.</li> <li>• Use QH.LP to get next (QH+QE).</li> </ul>
1	1	0	x	1	x	x	x	<ul style="list-style-type: none"> <li>• In Queue. Empty queue.</li> <li>• Use QH.LP to get next (QH+QE)</li> </ul>

### 5.16.3 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 zeros causes the NRZI data to toggle each bit time. A string of ones causes long periods with no transitions in the data. 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 5-17](#).

**Figure 5-17. USB Data Encoding**



Bit stuffing is enabled beginning with the Sync Pattern and throughout the entire transmission. The data “one” 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 zero bit will be inserted even if it is the last bit before the end-of-packet (EOP) signal.

## 5.16.4 Bus Protocol

### 5.16.4.1 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.

### 5.16.4.2 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 eight 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.

### 5.16.4.3 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 5-68. 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 5-68](#). 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 ones 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 5-69](#).

**Table 5-69. 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.

#### 5.16.4.4 Address Fields

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

**Table 5-70. Address Field**

Bit	Data Sent	Bit	Data Sent
0	ADDR 0	4	ADDR 4
1	ADDR 1	5	ADDR 5
2	ADDR 2	6	ADDR 6
3	ADDR 3		

#### 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 5-70](#), 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 5-71](#), 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 5-71. Endpoint Field**

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

#### 5.16.4.5 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 x7FFh and is sent only for SOF tokens at the start of each frame.

#### 5.16.4.6 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.

#### 5.16.4.7 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.

## 5.16.5 Packet Formats

### 5.16.5.1 Token Packets

Table 5-72 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 ICH2 can issue token packets. IN PIDs define a data transaction from a function to the ICH2. OUT and SETUP PIDs define data transactions from the ICH2 to a function.

Token packets have a five-bit CRC that 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 5-72. Token Format**

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

### 5.16.5.2 Start of Frame Packets

Table 5-73 shows a start of frame (SOF) packet. SOF packets are issued by the host at a nominal rate of once every 1.00 ms. 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, that 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 5-73. SOF Packet**

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

### 5.16.5.3 Data Packets

A data packet consists of a PID, a data field, and a CRC as shown in [Table 5-74](#). 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 5-74. Data Packet Format**

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

### 5.16.5.4 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.

### 5.16.5.5 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 ICH2, 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 will issue 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.

## 5.16.6 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 ICH2 operation error. All transaction-based sources can be masked by software through the ICH2's Interrupt Enable register. Additionally, individual transfer descriptors can be marked to generate an interrupt on completion.

When the ICH2 drives an interrupt for USB, it drives the PIRQD# pin active for interrupts occurring due to ports 0 and 1 until all sources of the interrupt are cleared.

### 5.16.6.1 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 ICH2 to a USB device or a packet transmitted from a USB device to the ICH2 generates a CRC error. The ICH2 is informed of this event by a time-out from the USB device or by the ICH2'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 will cause the C\_ERR field of the TD to decrement. When the C\_ERR field decrements to zero, 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 is 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 zero 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 1 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 ICH2 (due to incorrect schedule for instance), the ICH2 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 ICH2 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 zero, 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 than 6 ones in a row within the incoming data stream. This will cause the C\_ERR field of the TD to be decremented. When the C\_ERR field decrements to zero, 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.

### 5.16.6.2 Non-Transaction Based Interrupts

If an ICH2 process error or system error occur, the ICH2 halts and immediately issues a hardware interrupt to the system.

## Resume Received

This event indicates that the ICH2 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 will be signaled to the system allowing the USB to be brought out of the suspend state and returned to normal operation.

## ICH2 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 ICH2 sets this bit to 1 when a PCI Parity error, PCI Master Abort, or PCI Target Abort occurs. When this error occurs, the ICH2 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.

## 5.16.7 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 ICH2 so that a device on a port may wake the system. Such a device may be a fax-modem, that wakes up the machine to receive a fax or takes a voice message. The settings of the following bits in I/O space is maintained when the ICH2 enters the S3, S4 or S5 states.

**Table 5-75. 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 ICH2 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 is generated.

### 5.16.8 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 DOS legacy software will not run; this is because the keyboard is not identified. The ICH2 implements a series of trapping operations which snoop accesses that go to the keyboard controller and put the expected data from the USB keyboard into the keyboard controller.

**Note:** The scheme described below assumes that the keyboard controller (8042 or equivalent) is on the LPC bus.

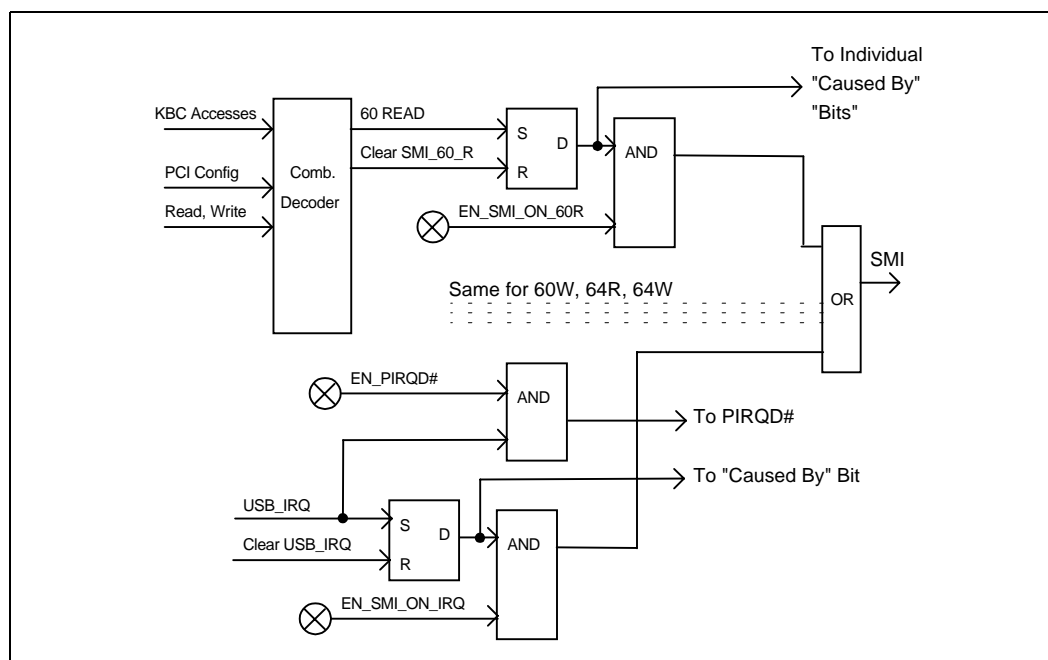
This legacy operation is performed through SMM space.

Figure 5-18 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 does not complete the cycle before the SMI is observed. This method is used on MPIIX and has been validated.

The logic will also need to block the accesses to the 8042. If there is an external 8042, this is accomplished by not activating the 8042 CS. This is done by logically ANDing the 4 enables (60R, 60W, 64R, 64W) with the 4 types of accesses to determine if the 8042CS should go active. An additional term is required for the "Pass-through" case. The state table for the diagram is shown in Table 5-76.

**Figure 5-18. USB Legacy Keyboard Flow Diagram**



**Table 5-76. 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 goes to 1.
IDLE	64h / Write	Not D1h	IDLE	Bit 3 in configuration Register determines if cycle passed through to 8042 and if SMI# generated.
IDLE	64h / Read	N/A	IDLE	Bit 2 in configuration Register determines if cycle passed through to 8042 and if SMI# generated.
IDLE	60h / Write	Don't Care	IDLE	Bit 1 in configuration Register determines if cycle passed through to 8042 and if SMI# generated.
IDLE	60h / Read	N/A	IDLE	Bit 0 in configuration 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 configuration 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 configuration Register. No SMI# generated. PSTATE remains 1. Stay in GateState1 because this is part of the double-trigger sequence.
GateState1	64h / Write	Not D1h	IDLE	Bit 3 in configuration space determines if cycle passed through to 8042 and if SMI# generated. PSTATE goes to 0. If Bit 7 in configuration Register is set, then SMI# should be generated.
GateState1	60h / Read	N/A	IDLE	This is an invalid sequence. Bit 0 in configuration Register determines if cycle passed through to 8042 and if SMI# generated. PSTATE goes to 0. If Bit 7 in configuration 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 configuration 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 configuration Space determines if SMI# should be generated.
GateState2	64h / Write	Not FFh	IDLE	Improper end of sequence. Bit 3 in configuration Register determines if cycle passed through to 8042 and if SMI# generated. PSTATE goes to 0. If Bit 7 in configuration 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 configuration Register. PSTATE remains 1.
GateState2	60h / Write	XXh	IDLE	Improper end of sequence. Bit 1 in configuration Register determines if cycle passed through to 8042 and if SMI# generated. PSTATE goes to 0. If Bit 7 in configuration Register is set, then SMI# should be generated.
GateState2	60h / Read	N/A	IDLE	Improper end of sequence. Bit 0 in configuration Register determines if cycle passed through to 8042 and if SMI# generated. PSTATE goes to 0. If Bit 7 in configuration Register is set, then SMI# should be generated.

## 5.17 SMBus Controller Functional Description (D31:F3)

The ICH2 provides an SMBus 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 ICH2 is also capable of operating in a mode in which it can communicate with I<sup>2</sup>C compatible devices.

The Slave Interface allows an external master to read from or write to the ICH2. 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 ICH2's internal Host Controller cannot access the ICH2's internal Slave Interface.

The ICH2 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 ICH2 SMBus controller logic is clocked by RTC clock.

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 (e.g., 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.

### 5.17.1 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 then tells the controller to start. When the controller has finished transmitting data on writes, or receiving data on reads, it will generate an SMI# or interrupt, if enabled.

The host controller supports 7 command protocols of the SMBus interface (see System Management Bus Specs, Rev 1.0): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Process Call, and Block Read/Write.

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 will update all registers while completing the new command.

Using the SMB Host Controller to send commands to the ICH2's SMB slave port is not supported.

### 5.17.1.1 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 is 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 format of the protocol is shown in [Table 5-77](#).

**Table 5-77. Quick Protocol**

Bit	Description
1	Start Condition
2:8	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.

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 5-78](#).

**Table 5-78. Send / Receive Byte Protocol**

Send Byte Protocol		Receive Byte Protocol	
Bit	Description	Bit	Description
1	Start	1	Start
2:8	Slave Address - 7 bits	2:8	Slave Address - 7 bits
9	Write	9	Read
10	Acknowledge from slave	10	Acknowledge from slave
11:18	Command code - 8 bits	11:18	Data byte from slave
19	Acknowledge from slave	19	NOT Acknowledge
20	Stop	20	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. The format of the protocol is shown in [Table 5-79](#).

Table 5-79. Write Byte/Word Protocol

Write Byte Protocol		Write Word Protocol	
Bit	Description	Bit	Description
1	Start	1	Start
2:8	Slave Address - 7 bits	2:8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11:18	Command code - 8 bits	11:18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave
20:27	Data Byte - 8 bits	20:27	Data Byte Low - 8 bits
28	Acknowledge from Slave	28	Acknowledge from Slave
29	Stop	29:36	Data Byte High - 8 bits
		37	Acknowledge from slave
		38	Stop

### Read Byte/Word

Reading data is slightly more complicated than writing data. First the ICH2 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.

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 5-80.

Table 5-80. Read Byte/Word Protocol

Read Byte Protocol		Read Word Protocol	
Bit	Description	Bit	Description
1	Start	1	Start
2:8	Slave Address - 7 bits	2:8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11:18	Command code - 8 bits	11:18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave
20	Repeated Start	20	Repeated Start
21:27	Slave Address - 7 bits	21:27	Slave Address - 7 bits
28	Read	28	Read
29	Acknowledge from slave	29	Acknowledge from slave
30:37	Data from slave - 8 bits	30:37	Data Byte Low from slave - 8 bits
38	NOT acknowledge	38	Acknowledge
39	Stop	39:46	Data Byte High from slave - 8 bits
		47	NOT acknowledge
		48	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 ICH2 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 format of the protocol is shown in [Table 5-81](#).

**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 5-81. Process Call Protocol**

Bit	Description
1	Start
2:8	Slave Address - 7 bits
9	Write
10	Acknowledge from Slave
11:18	Command code - 8 bits
19	Acknowledge from slave
20:27	Data byte Low - 8 bits
28	Acknowledge from slave
29:36	Data Byte High - 8 bits
37	Acknowledge from slave
38	Repeated Start
39:45	Slave Address - 7 bits
46	Read
47	Acknowledge from slave
48:55	Data Byte Low from slave - 8 bits
56	Acknowledge
57:64	Data Byte High from slave - 8 bits
65	NOT acknowledge
66	Stop

## Block Read/Write

The Block Write begins with a slave address and a write condition. After the command code, the ICH2 issues a byte count which describes 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 the 20 bytes of data. The byte count may not be 0.

Note that, unlike the PIIX4, which implements 32-byte buffer for Block Read/Write command, the ICH2 implements the Block Data Byte register (D31:F3, I/O offset 07h) for Block Read/Write command.

When programmed for a block write command, the Transmit Slave Address, Host Command, and Data0 (count) registers are sent. Data is then sent from the Block Data Byte register. After the byte has been sent, the ICH2 sets the BYTE\_DONE\_STS bit in the Host Status register. If there are more bytes to send, software writes the next byte to the Block Data Byte register and also clears the BYTE\_DONE\_STS bit. The ICH2 then sends the next byte. When doing a block write, first poll the BYTE\_DONE\_STS register until it is set, then write the next byte, then clear the BYTE\_DONE\_STS register.

On block read commands, after the byte count is stored in the DATA 0 register, the first data byte goes in the Block Data Byte Register; the ICH2 will then set the BYTE\_DONE\_STS bit and generate an SMI# or interrupt. The SMI# or interrupt handler reads the byte and then clears the BYTE\_DONE\_STS bit to allow the next byte to be read into the Block Data Byte register. Note that after receiving data byte N-1 of the block, the software needs to set the LAST\_BYTE bit in the Host Control Register; this allows the ICH2 to send a NOT ACK (instead of an ACK) after receiving the last data byte (byte N) of the block.

After each byte of a block message the ICH2 sets the BYTE\_DONE\_STS bit and generates an interrupt or SMI#. Software clears the BYTE\_DONE\_STS bit before the next transfer occurs. When the interrupt handler clears the BYTE\_DONE\_STS bit after the last byte has been transferred, the ICH2 sets the INTR bit and generates another interrupt to signal the end of the block transfer. Thus, for a block message of n bytes, the ICH2 generates n+1 interrupts. The interrupt handler needs to be implemented to handle all of these interrupts

The format of the Block Read/Write protocol is shown in [Table 5-82](#).

**Note:** For Block Write, if the I<sup>2</sup>C\_EN bit is set, the format of the command changes slightly. The ICH2 still sends the number of bytes indicated in the DATA0 register. However, it does not send the contents of the Data 0 register as part of the message.

**Table 5-82. Block Read/Write Protocol**

Block Write Protocol		Block Read Protocol	
Bit	Description	Bit	Description
1	Start	1	Start
2:8	Slave Address - 7 bits	2:8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11:18	Command code - 8 bits	11:18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave
20:27	Byte Count - 8 bits (Skip this step if I <sup>2</sup> C_En bit set)	20	Repeated Start
28	Acknowledge from Slave (Skip this step if I2C_EN bit set)	21:27	Slave Address - 7 bits
29:36	Data Byte 1 - 8 bits	28	Read
37	Acknowledge from Slave	29	Acknowledge from slave
38:45	Data Byte 2–8 bits	30:37	Byte Count from slave - 8 bits
46	Acknowledge from slave	38	Acknowledge
...	Data Bytes / Slave Acknowledges...	39:46	Data Byte 1 from slave - 8 bits
...	Data Byte N - 8 bits	47	Acknowledge
...	Acknowledge from Slave	48:55	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

## I<sup>2</sup>C Read

This command allows the ICH2 to perform block reads to certain I<sup>2</sup>C devices (e.g., serial E<sup>2</sup>PROMs). The SMBus Block Read sends both the 7-bit address, as well as the Command field. This command field could be used as the extended 10-bit address for accessing I<sup>2</sup>C devices that use 10-bit addressing.

However, this does not allow access to devices using the I<sup>2</sup>C “Combined Format” that has data bytes after the address. Typically, these data bytes correspond to an offset (address) within the serial memory chips.

**Note:** This new command is supported independent of the setting of the I<sup>2</sup>C\_EN bit.

For I<sup>2</sup>C Read command, the value written into bit 0 of the Transmit Slave Address Register (SMB I/O register, offset 04h) needs to be 0. The format that is used for the new command is shown in [Table 5-83](#):

Table 5-83. I<sup>2</sup>C Block Read

Bit	Description
1	Start
2:8	Slave Address - 7 bits
9	Write
10	Acknowledge from slave
11:18	Command code - 8 bits
19	Acknowledge from slave
20:27	Send DATA0 register
28	Acknowledge from slave
29:36	Send DATA1 register
37	Acknowledge from slave
38	Repeated start
39:45	Slave Address - 7 bits
46	Read
47	Acknowledge from slave
48:55	Data byte from slave
56	Acknowledge
57:64	Data byte 2 from slave - 8 bits
65	Acknowledge
-	Data bytes from slave / Acknowledge
-	Data byte N from slave - 8 bits
-	NOT Acknowledge
-	Stop

The ICH2 continues reading data from the peripheral until the NAK is received.

### 5.17.1.2 I<sup>2</sup>C Behavior

When the I<sup>2</sup>C\_EN bit is set, the ICH2 SMBus logic is instead set to communicate with I<sup>2</sup>C devices. This forces the following changes:

1. The Process Call command will skip the Command code (and its associated acknowledge)
2. The Block Write command will skip sending the Byte Count (DATA0)

In addition, the ICH2 supports the new I<sup>2</sup>C Read command. This is independent of the I<sup>2</sup>C\_EN bit.

### 5.17.1.3 Heartbeat for Use With the External LAN Controller

This method allows the ICH2 to send messages to an *external* LAN Controller when the processor is otherwise unable to do so. It uses the SMLINK I/F between the ICH2 and the external LAN Controller. The actual Heartbeat message is a Block Write. Only 8 bytes are sent.

## 5.17.2 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 ICH2 continuously monitors the SMBDATA line. When the ICH2 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 ICH2 stops transferring data.

If the ICH2 sees that it has lost arbitration, the condition is called a collision. The ICH2 sets the BUS\_ERR bit in the Host Status Register, and, if enabled, generates an interrupt or SMI#. The processor is responsible for restarting the transaction.

When the ICH2 is a SMBus master, it drives the clock. When the ICH2 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 does not start toggling the clock until the start or stop condition meets proper setup and hold time. The ICH2 also guarantees minimum time between SMBus transactions as a master.

The ICH2 supports the same arbitration protocol for both the SMBus and the System Management (SMLINK) interfaces.

### Clock Stretching

Some devices may not be able to handle their clock toggling at the rate that the ICH2, as an SMBus master, would like. They have the capability of stretching the low time of the clock. When the ICH2 attempts to release the clock (allowing the clock to go high), the clock will remain low for an extended period of time.

The ICH2 monitors the SMBus clock line after it releases the bus to determine whether to enable the counter for the high time of the clock. While the bus is still low, the high time counter must not be enabled. Similarly, the low period of the clock can be stretched by an SMBus master if it is not ready to send or receive data.

The ICH2 SMBus Host Controller will never stretch the low period of the clock (SMBCLK). It always has the data to transfer on writes and it always has a spot for the data on reads.

The SMLINK interface, however, always stretches the low period of the clock, effectively forcing transfers down to 16 KHz.

### Bus Time Out (ICH2 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 times out. The ICH2 discards the cycle and sets the DEV\_ERR bit. The time-out minimum is 25 ms. The time-out counter inside the ICH2 starts after the last bit of data is transferred by the ICH2 and it is waiting for a response. The 25 ms is a count of 800 RTC clocks.

## 5.17.3 Interrupts / SMI#

The ICH2 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.

## 5.17.4 SMBALERT#

SMBALERT# is multiplexed with GPIO[11]. When enabled and the signal is asserted, the ICH2 can generate an interrupt, an SMI#, or a wake event from S1-S4. To resume using SMBALERT#, the SMB\_SMI\_EN bit must be enabled to generate an SMI (see [Section 12.1.12, “HOSTC—Host Configuration Register \(SMBUS—D31:F3\)”](#) on page 12-4).

## 5.17.5 SMBus Slave Interface

The ICH2's SMBus Slave interface is accessed via the SMLINK[1:0] signals. The slave interface allows the ICH2 to decode cycles and allows an external microcontroller to perform specific actions. Key features and capabilities include:

- Supports decode of two messages type: Write and Read
- Receive Slave Address register: This is the address that the ICH2 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 ICH2. See [Table 5-88](#)
- Status bit to indicate that the SMBus logic caused an SMI# due to the reception of a message that matched the slave address. See [Section 9.8.3.12](#).

### Format of Slave Write Cycle

The external master performs Byte Write commands to the ICH2 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 5-84](#). [Table 5-85](#) lists the values associated with the registers.

**Table 5-84. Slave Write Cycle Format**

Bits	Description	Driven by	Comment
1	Start Condition	External Microcontroller	
2:8	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
9	Write	External Microcontroller	Always 0
10	ACK	ICH2	
11:18	Command	External Microcontroller	This field indicates which register will be accessed. See <a href="#">Table 5-85</a> below for the register definitions
19	ACK	ICH2	
20:27	Register Data	External Microcontroller	See <a href="#">Table 5-85</a> below for the register definitions
28	ACK	ICH2	
29	Stop	External Microcontroller	

**Table 5-85. Slave Write Registers**

Register	Function
0	Command Register. See Table 65 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 ICH2 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. ICH2 will not attempt to cover this race condition (i.e., unpredictable results in this case).

**Table 5-86. Command Types**

Command Type	Description
0	Reserved
1	<b>WAKE/SMI#:</b> Wake system if it is not already awake. If the system is already awake, an SMI# is generated.
2	<b>Unconditional Powerdown:</b> This command sets the PWRBTNOR_STS bit and has the same effect as the Powerbutton Override occurring. This functionality depends upon the BIOS having cleared the PWRBTN_STS bit.
3	<b>Hard Reset without Cycling:</b> This 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	<b>Hard Reset System:</b> This causes a hard reset of the system (including cycling of the power supply). This is equivalent to a write to the CF9h register with bits 3:1 set to 1.
5	<b>Disable the TCO Messages.</b> This command disables the ICH2 from sending Heartbeat and Event messages (as described in <a href="#">Section 5.13.2</a> ). 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	<b>WD RELOAD:</b> Reload watchdog timer.
7–FFh	Reserved

## Format of Read Command

The external master performs Byte Read commands to the ICH2 SMBus Slave interface. The “Command” field (bits 11:18) indicate which register is being accessed. The Data field (bits 30:37) contain the value that should be read from that register. [Table 5-87](#) shows the Read Cycle format. [Table 5-88](#) shows the register mapping for the data byte.

**Table 5-87. Read Cycle Format**

Bit	Description	Driven by	Comment
1	Start	External Microcontroller	
2:8	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
9	Write	External Microcontroller	Always 0
10	ACK	ICH2	
11:18	Command code – 8 bits	External Microcontroller	Indicates which register is being accessed. See <a href="#">Table 5-88</a> .
19	ACK	ICH2	
20	Repeated Start	External Microcontroller	
21:27	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
28	Read	External Microcontroller	Always 1
29	ACK	ICH2	
30:37	Data Byte	ICH2	Value depends on register being accessed. See <a href="#">Table 5-88</a> .
38	NOT ACK	External Microcontroller	
39	Stop	ICH2	

**Table 5-88. Data Values for Slave Read Registers**

Register	Bits	Description
0	7:0	Reserved.
1	2:0	System Power State 000 = S0 001 = S1 010 = Reserved 011 = S3 100 = S4 101 = S5 110 = Reserved 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 is set if the ICH2's THRM# input signal is active. Need to take after polarity control.
4	2	DOA processor status. This bit is 1 to indicate that the processor is dead.
4	3	1 = Watchdog timer expired. This bit is set if the ICH2's TCO timers have timed out.
4	6:4	Reserved.
4	7	Will reflect the state of the ICH2's GPIO[11].
5	0	Unprogrammed FWH bit. This bit will be 1 to indicate that the first BIOS fetch returned FFh, which indicates that the FWH is probably blank.
5	7:1	Reserved
6	7:0	Contents of the Message 1 register. See <a href="#">Section 9.9.10</a> .
7	7:0	Contents of the Message 2 register. See <a href="#">Section 9.9.10</a> .
8	7:0	Contents of the WDSTATUS register. See <a href="#">Section 9.9.11</a> .
9–FFh	7:0	Reserved

### Behavioral Notes

According to SMBus protocol, Read and Write messages always begin with a Start bit - Address - Write bit sequence. When the ICH2 detects that the address matches the value in the Receive Slave Address register, it assumes that the protocol is always followed and ignores the Write bit (bit 9) and signal an Acknowledge during bit 10 (See [Table 5-84](#) and [Table 5-87](#)). In other words, if a Start - Address - Read occurs (which is illegal for SMBus Read or Write protocol), and the address matches the ICH2's Slave Address, the ICH2 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 5-87](#)). Once again, if the Address matches the ICH2'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 ICH2's SMBus Slave logic until at least 1 second after both RTCRST# and RSMRST# are deasserted (high).

## 5.18 AC'97 Controller Functional Description (Audio D31:F5, Modem D31:F6)

**Note:** All references to AC'97 in this document refer to the AC'97 2.1 specification. For further information on the operation of the AC-link protocol, see the AC'97 specification.

The ICH2 AC '97 Controller features include:

- Independent PCI functions for audio and modem.
- Independent bus master logic for Mic input, PCM Audio input (2-channel stereo), PCM audio output (2, 4 or 6-channel stereo), Modem input and Modem output.
- 16 bit sample resolution
- Multiple sample rates up to 48 KHz
- 16 GPIOs
- Single modem line
- Dual codec configuration with two SDIN pins

Table 5-89 shows a detailed list of features supported by the ICH2 AC'97 digital controller.

**Table 5-89. Featured Supported by ICH2**

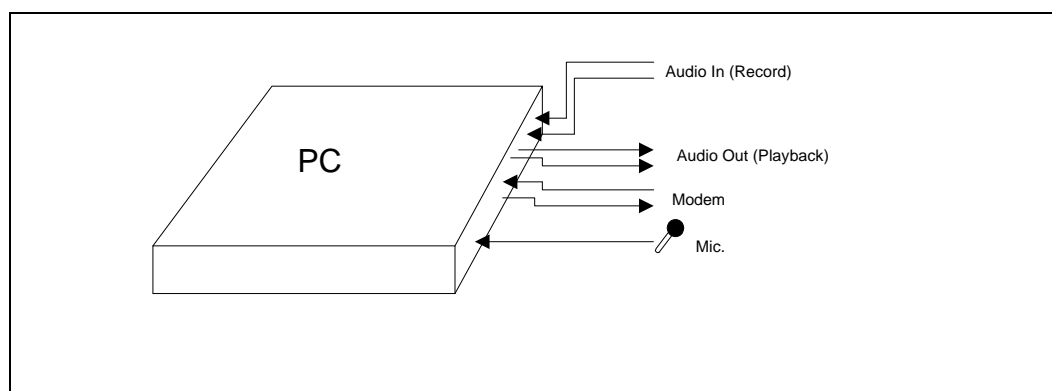
Feature	Description
System Interface	<ul style="list-style-type: none"> <li>• Isochronous low latency bus master memory interface</li> <li>• Scatter/gather support for word-aligned buffers in memory (all mono or stereo 16-bit data types are supported, no 8-bit data types are supported)</li> <li>• Data buffer size in system memory from 3 to 65535 samples per input</li> <li>• Data buffer size in system memory from 0 to 65535 samples per output</li> <li>• Independent PCI audio and modem functions with configuration and IO spaces</li> <li>• AC'97 codec registers are shadowed in system memory via driver (not PCI IO space)</li> <li>• AC'97 codec register accesses are serialized via semaphore bit in PCI IO space (new accesses are not allowed while a prior access is still in progress)</li> </ul>
Power Management	<ul style="list-style-type: none"> <li>• Power management via ACPI control methods</li> <li>• Support for audio states: <i>D0</i>, <i>D2</i>, <i>D3hot</i>, <i>D3cold</i></li> <li>• Support for modem states: <i>D0</i>, <i>D3hot</i>, <i>D3cold</i></li> <li>• SCI event generation for PCI modem function with wake-up from <i>D3cold</i></li> <li>• Independent codec D3 w/ Link down event, synchronized via two bit semaphore (in PCI IO Space)</li> </ul>
PCI Audio Function	<ul style="list-style-type: none"> <li>• Read/write access to audio codec registers 00h-3Ah and vendor registers 5Ah-7Eh</li> <li>• 16-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)</li> <li>• 16-bit stereo PCM input, up to 48 kHz (L,R channels on slots 3,4)</li> <li>• 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)</li> <li>• 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)</li> <li>• 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)</li> </ul>

**Table 5-89. Featured Supported by ICH2 (Continued)**

Feature	Description
PCI Modem function	<ul style="list-style-type: none"> <li>• Read/write access to modem codec registers 3Ch-58h and vendor registers 5Ah-7Eh</li> <li>• 16-bit mono modem line1 output and input, up to 48 kHz (slot 5)</li> <li>• Low latency GPIO[13:11,8:6,4:3,1:0] via hardwired update between slot 12 and PCI IO register</li> <li>• Programmable PCI interrupt on modem GPIO input changes via slot 12 GPIO_INT</li> <li>• SCI event generation on primary or secondary SDIN wake-up signal</li> </ul>
AC-link	<ul style="list-style-type: none"> <li>• AC'97 2.1 compliant AC-link interface</li> <li>• Variable sample rate output support via AC'97 SLOTREQ protocol (slots 3,4,5,6,7,8,9)</li> <li>• Variable sample rate input support via monitoring of slot valid tag bits (slots 3,4,5,6)</li> <li>• 3.3 V digital operation meets AC'97 2.1 DC switching levels</li> <li>• AC-Link IO driver capability meets AC'97 2.1 dual codec specifications</li> <li>• Codec register status reads must be returned with data in the next AC-link frame, per AC'97 2.1 specification.</li> </ul>
Multiple Codec	<ul style="list-style-type: none"> <li>• Dual codec addressing: All AC'97 codec register accesses are addressable to codec ID 00 (primary) or codec ID 01 (secondary)</li> <li>• Dual codec receive capability via primary and secondary SDIN pins (primary, secondary SDIN frames are internally validated, synchronized, and OR'd)</li> </ul>

**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.

**Figure 5-19. ICH2 Based AC'97 2.1**



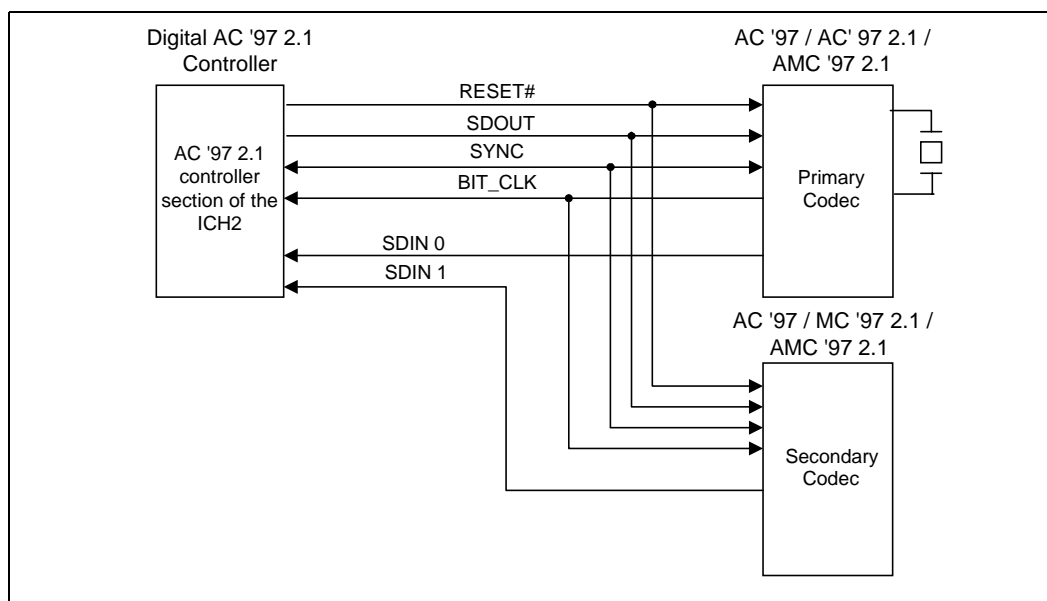
## 5.18.1 AC-link Overview

The ICH2 is an AC'97 2.1 compliant 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 ICH2 AC-link allows a maximum of two codecs to be connected.

Figure 5-20 shows a two codec topology of the AC-link for the ICH2.

Figure 5-20. AC'97 2.1 Controller-Codec Connection



The AC-link consists of a five signal interface between the controller and codec. Table 5-90 indicates the AC-link signal pins on the ICH2 and their associated power wells.

Table 5-90. AC'97 Signals

Signal Name	Type	Power Well*	Description
AC_RESET#	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_SDIN 0	Input	Resume	Serial input data
AC_SDIN 1	Input	Resume	Serial input data

**NOTE:** Power well voltage levels are 3.3V

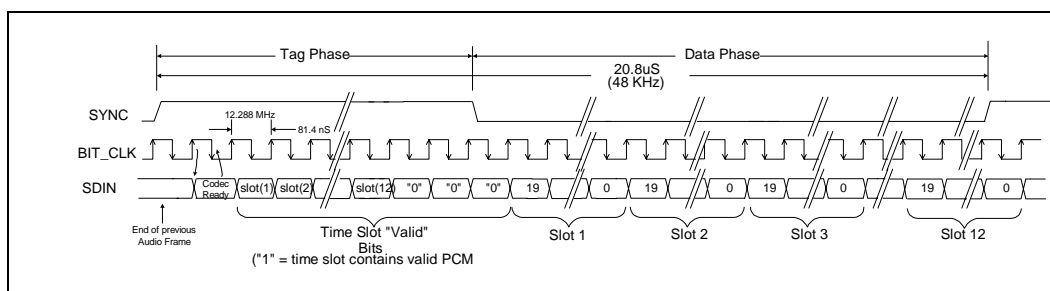
ICH2 core well outputs may be used as strapping options for the ICH2, sampled during system reset. These signals may have weak pull-ups/put-downs; however, this will not interfere with link operation. ICH2 inputs integrate weak put-downs to prevent floating traces when a secondary codec is not attached. 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 pull-ups/put-downs.

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 BIT\_CLK. The receiver of AC-link data samples each serial bit on the falling edge of BIT\_CLK.

Synchronization of all AC-link data transactions is signaled by the AC'97 controller via the AC\_SYNC signal, as shown in Figure 5-21. 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 BIT\_CLK. AC\_SYNC remains

high for a total duration of 16 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 BIT\_CLK.

**Figure 5-21. AC-link Protocol**



The ICH2 has two SDIN pins allowing a single or dual codec configuration. When two codecs are connected, the primary and secondary codecs can be connected to either SDIN line, however it is recommended that the primary codec be attached to SDIN [0]. The ICH2 does not distinguish between primary and secondary codecs on its SDIN[1:0] pins; however, the registers do distinguish between SDIN[0] and SDIN[1] for wake events, etc. The primary codec can be an AC (audio codec), MC (modem codec), or AMC (audio/modem codec) device. The secondary codec can be an AC, MC, or AMC device.

The MC can be either on the primary or the secondary codec, while the AC can be either on the primary or the secondary codec, or BOTH the primary or the secondary codec.

The ICH2 does not support optional test modes as outlined in the AC'97 specification.

## AC-link Output Frame (SDOUT)

A new audio output frame begins with a low to high transition of AC\_SYNC. AC\_SYNC is synchronous to the rising edge of BIT\_CLK. On the immediately following falling edge of 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 BIT\_CLK, the ICH2 transitions 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 BIT\_CLK, and subsequently sampled by the codec on the following falling edge of 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 ICH2 generates 16 bit samples and, in compliance with the AC'97 specification, pads the 4 least significant bits of valid slots with zeros.

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 ICH2, software must ensure both left and right sample stream time slots are filled with the same data.

## 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 one 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 zero in the corresponding bit position of slot 0, the ICH2 stuffs the corresponding slot with zeros 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 one, this indicates that the current frame contains at least one slot with valid data. When there is no transaction in progress, the ICH2 deasserts the frame valid bit. Note that after a write to slot 12, that slot always stays valid; 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 ICH2 using the SLOTRREQ bits as described in the AC'97 specification.

### 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 split codec implementation, accesses to the codecs are differentiated by the driver using address offsets 00h–7Fh for the primary codec and address offsets 80h–FEh for the secondary codec. The differentiation on the link, however, is done via the codec ID bits. See [Section](#) for further details.

### 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 then the entire slot time stuffed with 0s by the ICH2. Bits [19:4] contain the write data. Bits [3:0] are reserved and are stuffed with zeros.

### 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 ICH2 transmits sample streams of 16 bits and stuffs the remaining bits with zeros.

Data in output slots 3 and 4 from the ICH2 should be duplicated by software if there is only a single channel out.

### 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 ICH2 transmits sample streams of 16 bits and stuffs the remaining bits with zeros.

Data in output slots 3 and 4 from the ICH2 should be duplicated by software if there is only a single channel out.

**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 ICH2 stuffs trailing bit positions within this time slot with 0s.

**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. If not set up for 6 channel mode, this channel will always be stuffed with 0s by ICH2.

**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.

**Output Slot 9: Playback SubWoofer Channel**

When set for 6 channel mode, this slot is used for the SubWoofer. The format is the same as Slots 3. If not set up for 6 channel mode, this channel will always be stuffed with 0s by ICH2.

**Output Slots 10–11: Reserved**

Output frame slots 10–11 are reserved and are always stuffed with 0s by the ICH2 AC'97 controller.

**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 will remain invalid until a register write to 54h/D4h.
2. A write to offset 54h/D4h in codec I/O space will cause 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 will cause the new data to be sent out on the next frame.
4. Slot 12 will get invalidated after the following events: PCI reset, AC'97 cold reset, warm reset, and hence a wake from S3, S4, or S5. Slot 12 will remain invalid until the next write to offset 54h/D4h.

## AC-link Input Frame (SDIN)

There are two SDIN lines on the ICH2 for use with a primary and secondary codec. Each SDIN pin can have a codec attached. Depending upon which codec (AC, MC, or AMC) is attached, various slots will be valid or invalid. The data slots on the two inputs must be completely orthogonal (except for the tag slot 0), that is, no two data slots at the same location will be valid on both lines. This precludes the use of two similar codecs (e.g., two ACs or MCs) which use the same time slots.

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 BIT\_CLK. On the immediately following falling edge of 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 BIT\_CLK, the codec transitions 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 BIT\_CLK and subsequently sampled by the ICH2 on the following falling edge of BIT\_CLK. This sequence ensures that data transitions and subsequent sample points for both incoming and outgoing data streams are time aligned.

SDIN data stream must follow the AC'97 specification and be MSB justified with all non-valid bit positions (for assigned and/or unassigned time slots) stuffed with zeros. SDIN data is sampled by the ICH2 on the falling edge of BIT\_CLK.

## 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 operation. If the codec ready bit in slot 0 is a zero, the codec is not ready for normal operation. 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 ICH2. 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 ICH2 contain valid data, just as in the output frame. The remaining bits in this slot are stuffed with zeros.

## 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 multiple 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 multiple sample rate input, the tag bit for each input slot indicates whether valid data is present or not.

**Table 5-91. Input Slot 1 Bit Definitions**

Bit	Description
19	Reserved (Set to zero)
18:12	Control Register Index (Stuffed with zeros if tagged invalid)
11	Slot 3 Request: PCM Left Channel*
10	Slot 4 Request: PCM Right Channel*
9	Slot 5 Request: Modem Line 1
8	Slot 6 Request: PCM Center Channel*
7	Slot 7 Request: PCM Left Surround*
6	Slot 8 Request: PCM Right Surround*
5	Slot 9 Request: PCM LFE Channel*
4:2	Slot Request 10-12: Not Implemented
1:0	Reserved (Stuffed with zeros)

**NOTE:** \*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 5-91](#), 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.

### 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.

### Input Slot 3: PCM Record Left Channel

Input slot 3 is the left channel input of the codec. ICH2 supports 16 bit sample resolution. Samples transmitted to the ICH2 must be in left/right channel order.

### Input Slot 4: PCM Record Right Channel

Input slot 4 is the right channel input of the codec. The ICH2 supports 16 bit sample resolution. Samples transmitted to the ICH2 must be in left/right channel order.

### Input Slot 5: Modem Line

Input slot 5 contains MSB justified modem data. The ICH2 supports 16 bit sample resolution.

### 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 that enables more precise echo cancellation algorithm for speakerphone applications. The ICH2 supports 16 bit resolution for slot 6 input.

### Input Slots 7-11: Reserved

Input frame slots 7–11 are reserved for future use and should be stuffed with zeros by the codec, per the AC'97 specification.

### 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. Bit 0 of this slot indicates the GPI status. When a GPI changes state, this bit gets set for one frame by the codec. This bit can cause an interrupt to the processor if enabled via the Global Control register.

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.

### Register Access

In the ICH2 implementation of the AC-link, up to two codecs can be connected to the SDOUT pin. The following mechanism is used to address the primary and secondary 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 5-92](#). 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; for I/O writes, both slots 1 and 2 valid bits are set.

The secondary 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 bit 0 (bit 0 of slot 0) is set to 1. This allows the secondary codec to monitor the slot valid bits of slots 1 and 2, and bit 0 of slot 0 to determine if the access is directed to the secondary codec. If the register access is targeted to the secondary 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 5-92. 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)

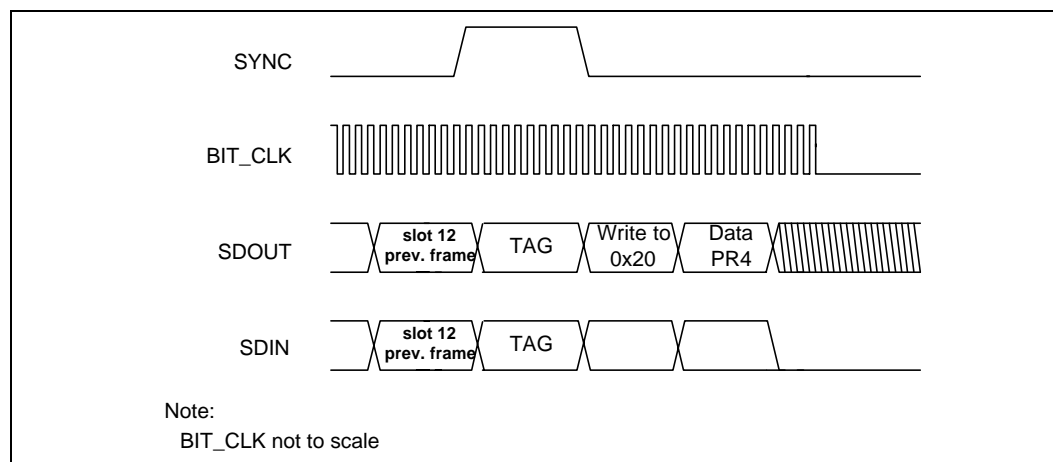
When accessing the codec registers, only one I/O cycle can be pending across the AC-link at any time. The ICH2 implements write posting on I/O writes across the AC-link (i.e., writes across the link are indicated as complete before they are actually sent across the link). To prevent a second I/O write 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 is reads to offset 54h/D4h (slot 12) which are returned immediately with the most recently received slot 12 data. Writes to offset 54h and D4h (primary and secondary 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 will 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.

## 5.18.2 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 BIT\_CLK and SDIN will be brought to and held at a logic low voltage level.

Figure 5-22. AC-link Powerdown Timing



BIT\_CLK and SDIN transition low immediately following a write to the Powerdown Register (26h) with PR4. 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.

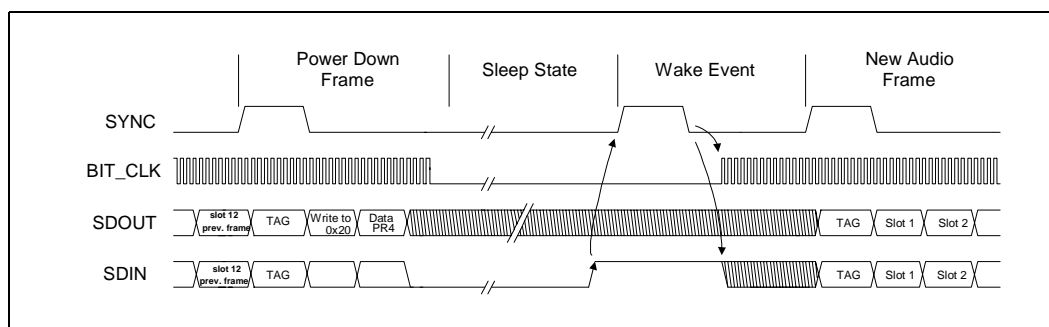
The AC'97 controller also drives AC\_SYNC, and SDOUT low after programming AC'97 to this low power, halted mode. Once the codec has been instructed to halt BIT\_CLK, a special wake up protocol must be used to bring the AC-link to the active mode since normal output and input frames can not be communicated in the absence of BIT\_CLK. Once in a low power mode, the ICH2 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.

## External Wake Event

Codecs can signal the controller to wake the AC-link and wake the system using SDIN. The minimum SDIN wake up pulse width is 1 us. The rising edge of SDIN[0] or SDIN[1] causes the ICH2 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 BIT\_CLK as diagrammed in Figure 5-23. The codec that signaled the wake event must keep its SDIN high until it has sampled AC\_SYNC having gone high, and then low.

Figure 5-23. SDIN Wake Signaling



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 4 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.

### 5.18.3 AC'97 Cold Reset

A cold reset is achieved by asserting AC\_RST# for 1 us. By driving AC\_RST# low, BIT\_CLK, and 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.

### 5.18.4 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 us in the absence of BIT\_CLK.

Within normal frames, AC\_SYNC is a synchronous AC'97 input to the codec. However, in the absence of 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 BIT\_CLK until AC\_SYNC has been sampled low again by the codec. This will prevent the false detection of a new frame.

**Note:** On receipt of wake up signalling from the codec, the digital controller will issue an interrupt if enabled. Software will then have to issue a warm or cold reset to the codec by setting the appropriate bit in the Global Control Register.

## 5.18.5 System Reset

Table 5-93 indicates the states of the link during various system reset and sleep conditions.

**Table 5-93. AC-link state during PCIRST#**

Signal	Power Plane	I/O	During PCIRST#	After PCIRST#	S1	S3	S4/S5
AC_RST#	Resume <sup>3</sup>	Output	Low	Low	Cold Reset bit (Hi)	Low	Low
AC_SDOUT	Core <sup>1</sup>	Output	Low	Running	Low	Low	Low
AC_SYNC	Core <sup>1</sup>	Output	Low	Running	Low	Low	Low
BIT_CLK	Core	Input	Driven by codec	Running	Low <sup>2,4</sup>	Low <sup>2,4</sup>	Low <sup>2,4</sup>
SDIN[1:0]	Resume	Input	Driven by codec	Running	Low <sup>2,4</sup>	Low <sup>2,4</sup>	Low <sup>2,4</sup>

**NOTE:**

1. ICH2 core well outputs are used as strapping options for the ICH2. They are sampled during system reset. These signals may have weak pull-ups/put-downs. The ICH2 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# will be held low during S3–S5. It cannot be programmed high during a suspend state.
4. BIT\_CLK and SDIN[1:0] 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 S1 sleep 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# will be asserted (low) by the ICH2 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 will never deassert 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.

## 5.19 Firmware Hub Interface

This section describes the memory cycle type to be used on the Firmware Hub (FWH) interface. Below are the various types of cycles that are supported by the product.

Cycle Type	Comment
FWH Memory Read	New chip select and addressing are used.
FWH Memory Write	New chip select and addressing are used.

### 5.19.1 Field Definitions

#### START

This one clock field indicates the start of a cycle. It is valid on the last clock that LFRAME# is sampled low. The two start fields that are used for the cycle are shown in the table below. If the start field that is sampled is not one of these values, then the cycle attempted is not a FWH Memory Cycle. It may be a valid memory cycle that the FWH component may wish to decode (i.e., it may be of the LPC memory cycle variety).

AD[3:0]	Indication
1101	FWH Memory Read
1110	FWH Memory Write

#### IDSEL (Device Select)

This one clock field is used to indicate which FWH component is being selected. The four bits transmitted over AD[3:0] during this clock are compared with values strapped onto pins on the FWH component. If there is a match, the FWH component will continue to decode the cycle to determine which bytes are requested on a read or which bytes to update on a write. If there is not a match, the FWH component may discard the rest of the cycle and go into a standby power state.

#### MSIZE (Memory Size)

The value '0000b' is sent in this field. A value of '0000b' corresponds to a single byte transfer. Other encodings of this field are reserved for future use.

#### MADDR (Memory Address)

This is a 7-clock field that provides a 28 bit memory address. This allows for up to 256 MB per memory device, for a total of a 4 GB addressable space. The address is transferred with the most significant nibble first.

#### SYNC

The SYNC protocol is the same as described in the LPC specification.

#### TAR

The TAR fields are the same as described in the LPC specification. Refer to this specification for further details.

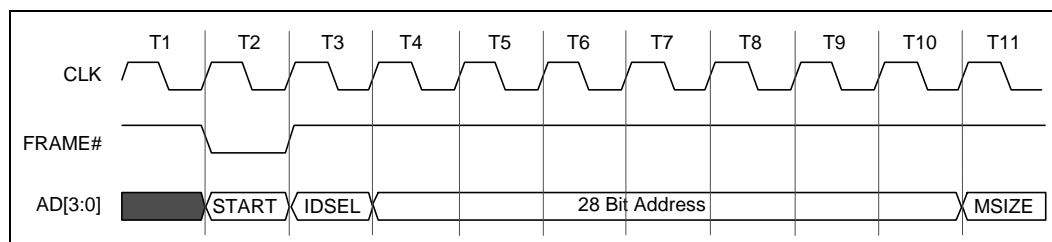
## 5.19.2 Protocol

The FWH Memory cycles use a sequence of events that start with a START field (LFRAME# active with appropriate AD[3:0] combination) and end with the data transfer. The following sections describe the cycles in detail.

### Preamble

The initiation of the FWH Memory cycles is shown in Figure 5-24. The FWH Memory transaction begins with LFRAME# going low and a START field driven on AD[3:0]. For FWH Memory Read cycles, the START field must be '1101b'; for FWH Memory Write cycles, the START field must be '1110b'. Following the START field is the IDSEL field. This field acts like a chip select in that it indicates which device should respond to the current transaction. The next seven clocks are the 28-bit address from where to begin reading in the selected device. Next, an MSIZE value of 0 indicates the master is requesting a single byte.

Figure 5-24. FWH Memory Cycle Preamble



### Read Cycle (Single Byte)

For read cycles, after the pre-amble (described above), the host drives a TAR field to give ownership of the bus to the FWH. After the second clock of the TAR phase, the target device assumes the bus and begins driving SYNC values. When it is ready, it drives the low nibble, then the high nibble of data, followed by a TAR to give control back to the host.

Figure 5-25. Single Byte Read

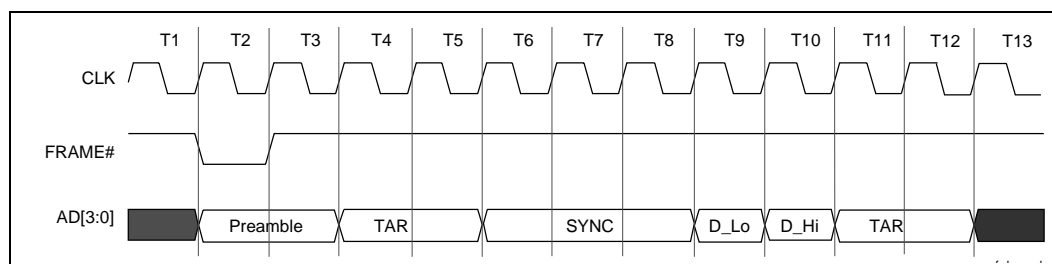


Figure 5-25 shows a device that requires 3 SYNC clocks to access data. Since the access time can begin once the address phase has been completed, the two clocks of the TAR phase can be considered as part of the access time of the part. For example, a device with a 120 ns access time could assert '0101b' for clocks 1 and 2 of the SYNC phase and '0000b' for the last clock of the SYNC phase. This would be equivalent to 5 clocks worth of access time if the device started that access at the conclusion of the Preamble phase. Once SYNC is achieved, the device returns the data in two clocks and gives ownership of the bus back to the host with a TAR phase.

### Write Cycles (Single Byte)

All devices that support FWH memory write cycles must support single byte writes. FWH memory write cycles use the same preamble as FWH memory read cycles that is described above.

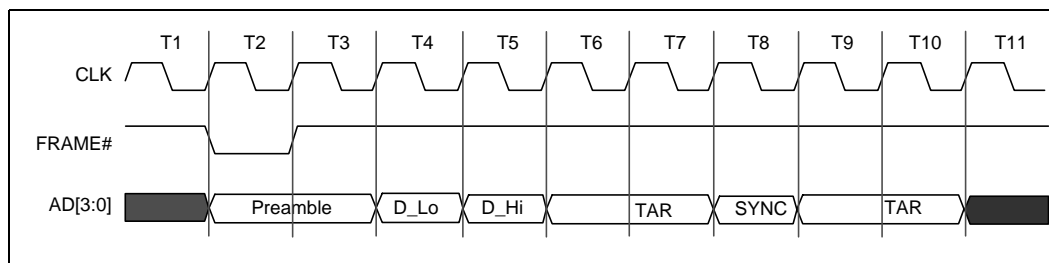
**Figure 5-26. Single Byte Write**

Figure 5-26 shows an FWH memory write cycle where a single byte is transferred. The master asserts an MSIZE value of 0. After the address has been transferred, the 2 clock data phase begins. Following the data phase, bus ownership is transferred to the FWH component with a TAR cycle. Following the TAR phase, the device must assert a SYNC value of '0000b' (ready) or '1010b' (error) indicating the data has been received. Bus ownership is then given back to the master with another TAR phase.

FWH Memory Writes only allow one clock for the SYNC phase. The TAR + SYNC + TAR phases at the end of FWH memory write cycles must be exactly 5 clocks.

## Error Reporting

There is no error reporting over the FWH interface for FWH memory cycles. If an error occurs (e.g., an address out of range or an unsupported memory size), the cycle will continue from the host unabated. This is because these errors are the result of illegal programming, and there is no efficient error reporting method that can be done to counter the programming error.

Therefore, the FWH component must not report the error conditions over the FWH interface. It must only report wait states and the 'ready' condition. It may choose to log the error internally to be debugged, but it must not signal an error through the FWH interface itself

# Register and Memory Mapping

## 6

The ICH2 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 ICH2 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.

<b>RO</b>	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.
<b>WO</b>	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.
<b>R/W</b>	Read/Write. A register with this attribute can be read and written.
<b>R/WC</b>	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.
<b>Default</b>	When ICH2 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 ICH2 registers accordingly.
<b>Bold</b>	Register bits that are highlighted in bold text indicate that the bit is implemented in the ICH2. Register bits that are not implemented or are rewired will remain in plain text.

## 6.1 PCI Devices and Functions

The ICH2 incorporates a variety of PCI functions as shown in [Table 6-1](#). These functions are divided into three logical devices (B0:D30, B0:D31 and B1:D8). D30 is the hub interface-to-PCI bridge, D31 contains the PCI-to-LPC Bridge, IDE Controller, USB Controllers, SMBus Controller and the AC'97 Audio and Model Controller functions. B1:D8 is the integrated LAN Controller.

**Note:** From a software perspective, the integrated LAN Controller resides on the ICH2'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 a particular system platform does not want to support any one of Device 31's Functions 1–6, they can individually be disabled. The integrated LAN Controller will be disabled if no Platform LAN Connect component is detected (See [Section 5.2, “LAN Controller \(B1:D8:F0\)” on page 5-5](#)). 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 6-1. 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 <sup>1</sup>
Bus 0:Device 31:Function 1	IDE Controller
Bus 0:Device 31:Function 2	USB Controller #1
Bus 0:Device 31:Function 3	SMBus Controller
Bus 0:Device 31:Function 4	USB Controller #2
Bus 0:Device 31:Function 5	AC'97 Audio Controller
Bus 0:Device 31:Function 6	AC'97 Modem Controller
Bus 1: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 ICH2 has a set of PCI configuration registers. The register address map tables for these register sets are included at the beginning of the chapter for the particular function.

Configuration Space registers are accessed through configuration cycles on the PCI bus by the Host bridge using configuration mechanism #1 detailed in the PCI 2.1 specification.

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. In some cases they can be disabled. Variable ranges can be moved and can also be disabled.

## 6.3.1 Fixed I/O Address Ranges

Table 6-2 shows the fixed I/O decode ranges from the processor perspective. Note that for each I/O range, there may be a separate behavior for reads and writes. The hub interface cycles that go to target ranges that are marked as “Reserved” are not decoded by the ICH2; they are passed to PCI. If a PCI master targets one of the fixed I/O target ranges, it will be positively decoded by the ICH2 in Medium speed.

Refer to Table A-1 for a complete list of all fixed I/O registers. Address ranges that are not listed or marked “Reserved” are NOT decoded by the ICH2 (unless assigned to one of the variable ranges).

**Table 6-2. Fixed I/O Ranges Decoded by ICH2**

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
2Eh–2Fh	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 <sup>5</sup>	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

Table 6-2. Fixed I/O Ranges Decoded by ICH2 (Continued)

I/O Address	Read Target	Write Target	Internal Unit
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
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 interface
170h–177h	IDE Controller <sup>1</sup>	IDE Controller <sup>1</sup>	Forwarded to IDE
1F0h–1F7h	IDE Controller <sup>2</sup>	IDE Controller <sup>2</sup>	Forwarded to IDE
376h	IDE Controller <sup>1</sup>	IDE Controller <sup>1</sup>	Forwarded to IDE
3F6h	IDE Controller <sup>2</sup>	IDE Controller <sup>2</sup>	Forwarded to IDE
4D0h–4D1h	Interrupt Controller	Interrupt Controller	Interrupt
CF9h	Reset Generator	Reset Generator	processor interface

**NOTES:**

1. Only if IDE Standard I/O space is enabled for Primary Drive. Otherwise, the target is PCI.
2. Only if IDE Standard I/O space is enabled for Secondary Drive. Otherwise, the target is PCI.
3. If POS\_DEC\_EN bit is enabled, reads from F0h will not be decoded by the ICH2. If POS\_DEC\_EN is not enabled, reads from F0h will forward to LPC.

## 6.3.2 Variable I/O Decode Ranges

Table 6-3 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 ICH2 positively decodes the cycle. If the response is on the behalf of an LPC device, ICH2 will forward the cycle to the LPC interface.

Refer to Table A-2 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 ICH2 does not perform any checks for conflicts.

**Table 6-3. Variable I/O Decode Ranges**

Range Name	Mappable	Size (Bytes)	Target
ACPI	Anywhere in 64 KB I/O Space	64	Power Management
IDE	Anywhere in 64 KB I/O Space	16	IDE Unit
USB #1	Anywhere in 64 KB I/O Space	32	USB Unit 1
SMBus	Anywhere in 64 KB I/O Space	16	SMB Unit
AC'97 Audio Mixer	Anywhere in 64 KB I/O Space	256	AC'97 Unit
AC'97 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
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 #2	Anywhere in 64 KB I/O Space	32	USB Unit 2
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

## 6.4 Memory Map

Table 6-4 shows (from the processor perspective) the memory ranges that the ICH2 decodes. Cycles that arrive from the MCH will first be driven out on PCI. The ICH2 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 will be positively decoded unless it falls in the PCI-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 will be forwarded up the hub interface to the Host Controller.

**Table 6-4. Memory Decode Ranges from Processor Perspective**

Memory Range	Target	Dependency/Comments
0000 0000h–000D FFFFh 0010 0000–TOM (Top of Memory)	Main Memory	TOM registers in Host Controller
000E 0000h–000F FFFFh	FWH	Bit 7 in FWH Decode Enable Register is set
FEC0 0000h–FEC0 0100h	I/O APIC inside ICH2	
FFC0 0000h–FFC7 FFFFh FF80 0000h–FF87 FFFFh	FWH	Bit 0 in FWH Decode Enable Register
FFC8 0000h–FFCF FFFFh FF88 0000h–FF8F FFFFh	FWH	Bit 1 in FWH Decode Enable Register
FFD0 0000h–FFD7 FFFFh FF90 0000h–FF97 FFFFh	FWH	Bit 2 in FWH Decode Enable Register is set
FFD8 0000h–FFDF FFFFh FF98 0000h–FF9F FFFFh	FWH	Bit 3 in FWH Decode Enable Register is set
FFE0 000h–FFE7 FFFFh FFA0 0000h–FFA7 FFFFh	FWH	Bit 4 in FWH Decode Enable Register is set
FFE8 0000h–FFE7 FFFFh FFA8 0000h–FFAF FFFFh	FWH	Bit 5 in FWH Decode Enable Register is set
FFF0 0000h–FFF7 FFFFh FFB0 0000h–FFB7 FFFFh	FWH	Bit 6 in FWH Decode Enable Register is set.
FFF8 0000h–FFFF FFFFh FFB8 0000h–FFBF FFFFh	FWH	Always enabled. The top two 64 KB blocks of this range can be swapped as described in <a href="#">Section 6.4.1</a> .
FF70 0000h–FF7F FFFFh FF30 0000h–FF3F FFFFh	FWH	Bit 3 in FWH Decode Enable 2 Register is set
FF60 0000h–FF6F FFFFh FF20 0000h–FF2F FFFFh	FWH	Bit 2 in FWH Decode Enable 2 Register is set
FF50 0000h–FF5F FFFFh FF10 0000h–FF1F FFFFh	FWH	Bit 1 in FWH Decode Enable 2 Register is set
FF40 0000h–FF4F FFFFh FF00 0000h–FF0F FFFFh	FWH	Bit 0 in FWH Decode Enable 2 Register is set
Anywhere in 4 GB range	D110 LAN Controller	Enable via BAR in Device 29:Function 0 (D110 LAN Controller)
All other	PCI	None

### 6.4.1 Boot-Block Update Scheme

The ICH2 supports a “Top-Block Swap” mode that has the ICH2 swap the top block in the FWH (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 ICH2 will invert A16 for cycles targeting FWH BIOS space. When this bit is 0, the ICH2 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-Block Swap” bit. This inverts A16 for cycles going to the FWH. Processor access to FFFF\_0000 through FFFF\_FFFF are directed to FFFF\_0000 through FFFE\_FFFF in the FWH. Processor accesses to FFFE\_0000 through FFFE\_FFFF are directed to FFFF\_0000 through FFFF\_FFFF.
4. Software erases the top block
5. Software writes the new top block
6. Software checks the new top block
7. Software clears the top-block swap bit

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 2.20.1](#)). 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 FWH BIOS space, not feature space.

**Note:** The Top Block Swap mode has no effect on accesses below FFFE\_0000.

This page is intentionally left blank.

# LAN Controller Registers (B1:D8:F0) 7

The ICH2 integrated LAN Controller appears to reside at PCI Device 8, Function 0 on the secondary side of the ICH2's virtual PCI-to-PCI Bridge (See [Table 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 (B1:D8:F0)

**Note:** Registers that are not shown should be treated as Reserved (See [Section 6.2](#) for details).

**Table 7-1. PCI Configuration Map (LAN Controller—B1:D8:F0)**

Offset	Mnemonic	Register Name/Function	Default	Type
00–01h	VID	Vendor ID	8086h	RO
02–03h	DID	Device ID	2449h	RO
04–05h	PCICMD	PCI Device Command Register	0000h	R/W
06–07h	PCISTS	PCI Device Status Register	0290h	R/W
08h	REVID	Revision ID	Note 1	RO
0Ah	SCC	Sub Class Code	00h	RO
0Bh	BCC	Base Class Code	02h	RO
0Dh	PMLT	PCI Master Latency Timer	00h	R/W
0Eh	HEADTYP	Header Type	00h	RO
10–13h	CSR_MEM_BASE	CSR Memory-mapped Base Address	0008h	R/W
14–17h	CSR_IO_BASE	CSR I/O-mapped Base Address	0001h	R/W
2C–2Dh	SVID	Subsystem Vendor ID	0000h	RO
2E–2Fh	SID	Subsystem ID	0000h	RO
34h	CAP_PTR	Capabilities Pointer	DCh	RO
3Ch	INT_LN	Interrupt Line	00h	R/W
3Dh	INT_PN	Interrupt Pin	01h	RO
3Eh	MIN_GNT	Minimum Grant	08h	RO
3Fh	MAX_LAT	Maximum Latency	38h	RO
DCh	CAP_ID	Capability ID	01h	RO
DDh	NXT_PTR	Next Item Pointer	00h	RO
DE–DFh	PM_CAP	Power Management Capabilities	FE21h	RO
E0–E1h	PMCSR	Power Management Control/Status	0000h	R/W
E3h	DATA	Data		RO

**NOTE:** Refer to the Specification Update for the value of the Revision ID Register.

### 7.1.1 VID—Vendor ID Register (LAN Controller—B1:D8:F0)

Offset Address: 00–01h      Attribute: RO  
Default Value: 8086h      Size: 16 bits

Bit	Description
15:0	<b>Vendor Identification Number.</b> This is a 16-bit value assigned to Intel.

### 7.1.2 DID—Device ID Register (LAN Controller—B1:D8:F0)

Offset Address: 02–03h      Attribute: RO  
Default Value: 2449h      Size: 16 bits

Bit	Description
15:0	<b>Device Identification Number.</b> This is a 16 bit value assigned to the ICH2 integrated LAN Controller.

### 7.1.3 PCICMD—PCI Command Register (LAN Controller—B1:D8:F0)

Offset Address: 04–05h      Attribute: RO, R/W  
Default Value: 0000h      Size: 16 bits

Bit	Description
15:10	Reserved.
9	<b>Fast Back to Back Enable (FBE)</b> —RO. Hardwired to 0. The integrated LAN Controller will not run fast back-to-back PCI cycles.
8	<b>SERR# Enable (SERR_EN)</b> —R/W. 1 = Enable. Allow SERR# to be generated. 0 = Disable.
7	<b>Wait Cycle Control (WCC)</b> —RO. Hardwired to 0. Not implemented.
6	<b>Parity Error Response (PER)</b> —R/W 1 = The integrated LAN Controller will take normal action when a PCI parity error is detected. The generation of parity is also enabled on the hub interface. 0 = The LAN Controller will ignore PCI parity errors.
5	<b>VGA Palette Snoop (VPS)</b> —RO. Hardwired to 0. Not Implemented.
4	<b>Memory Write and Invalidate Enable (MWIE)</b> —R/W. 0 = Disable. The LAN Controller will not use the Memory Write and Invalidate command. 1 = Enable.
3	<b>Special Cycle Enable (SCE)</b> —RO. Hardwired to 0. The LAN Controller ignores special cycles.
2	<b>Bus Master Enable (BME)</b> —R/W. 1 = Enable. The ICH2's integrated may function as a PCI bus master. 0 = Disable.
1	<b>Memory Space Enable (MSE)</b> —R/W. 1 = Enable. The ICH2's integrated LAN Controller will respond to the memory space accesses. 0 = Disable.
0	<b>I/O Space Enable (IOE)</b> —R/W. 1 = Enable. The ICH2's integrated LAN Controller will respond to the I/O space accesses. 0 = Disable.

## 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

Bit	Description
15	<b>Detected Parity Error (DPE)</b> —R/WC. 1 = The ICH2'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). 0 = This bit is cleared by writing a 1 to the bit location.
14	<b>Signaled System Error (SSE)</b> —R/WC. 1 = The ICH2's integrated LAN Controller has asserted SERR#. (SERR# can be routed to cause NMI, SMI# or interrupt). 0 = This bit is cleared by writing a 1 to the bit location.
13	<b>Master Abort Status (RMA)</b> —R/WC. 1 = The ICH2's integrated LAN Controller (as a PCI master) has generated a master abort. 0 = This bit is cleared by writing a 1 to the bit location.
12	<b>Received Target Abort (RTA)</b> —R/WC. 1 = The ICH2's integrated LAN Controller (as a PCI master) has received a target abort. 0 = This bit is cleared by writing a 1 to the bit location.
11	Signaled Target Abort (STA)—RO. Hardwired to 0. The device will never signal Target Abort.
10:9	DEVSEL# Timing Status (DEV_STS)—RO. 01h = Medium timing.
8	<b>Data Parity Error Detected (DPED)</b> —R/WC. 1 = All of the following three conditions have been met: 1.The LAN Controller is acting as bus master 2.The LAN Controller has asserted PERR# (for reads) or detected PERR# asserted (for writes) 3.The Parity Error Response bit in the LAN Controller's PCI Command Register is set. 0 = This bit is cleared by writing a 1 to the bit location.
7	Fast Back to Back (FB2B)—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 66MHz PCI.
4	<b>Capabilities List (CAP_LIST)</b> —RO. 1 = The EEPROM indicates that the integrated LAN controller supports PCI Power Management. 0 = The EEPROM indicates that the integrated LAN controller does not support PCI Power Management.
3:0	Reserved.

## 7.1.5 REVID—Revision ID Register (LAN Controller—B1:D8:F0)

Offset Address: 08h Attribute: RO  
Default Value: 00h Size: 8 bits

Bit	Description
7:0	<b>Revision Identification Number.</b> 8-bit value that indicates the revision number for the integrated LAN Controller. The three least significant bits in this register may be overridden by the ID and REV ID fields in the EEPROM.

### 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	<b>Sub-Class Code.</b> 8-bit value that 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	<b>Base Class Code.</b> 8-bit value that 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	<b>Cache Line Size (CLS)—RW.</b> 00 = Memory Write and Invalidate (MWI) command will not be used by the integrated LAN Controller. 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). 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). 11 = Invalid. MWI command will not be used.
2:0	Reserved.

### 7.1.9 PMLT—PCI Master Latency Timer Register (LAN Controller—B1:D8:F0)

Offset Address: 0Dh                      Attribute: RW  
Default Value: 00h                      Size: 8 bits

Bit	Description
7:3	<b>Master Latency Timer Count (MLTC)—RW.</b> 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	<b>Multi-function Device</b> —RO. Hardwired to 0 to indicate a single function device.
6:0	<b>Header Type</b> —RO. 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: 0000 0008h Size: 32 bits

**Note:** The ICH2'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	<b>Base Address</b> —R/W. Upper 20 bits of the base address provides 4 KB of memory-mapped space for the LAN Controller's Control/Status Registers.
11:4	Reserved.
3	<b>Pre-fetchable</b> —RO. Hardwired to 0 to indicate that this is not a pre-fetchable memory-mapped address range.
2:1	<b>Type</b> —RO. Hardwired to 00b to indicate the memory-mapped address range may be located anywhere in 32-bit address space.
0	<b>Memory-Space Indicator</b> —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  
Default Value: 0000 0001h Size: 32 bits

**Note:** The ICH2'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	<b>Base Address</b> —R/W. Provides 64 bytes of I/O-mapped address space for the LAN Controller's Control/Status Registers.
5:1	Reserved.
0	<b>I/O Space Indicator</b> —RO. Hardwired to 1 to indicate that this base address maps to I/O space.

### 7.1.13 SVID—Subsystem Vendor ID(LAN Controller—B1:D8:F0)

Offset Address: 2C–2D      Attribute: RO  
 Default Value: 0000h      Size: 16 bits

Bit	Description
15:0	Subsystem Vendor ID—RO.

### 7.1.14 SID—Subsystem ID(LAN Controller—B1:D8:F0)

Offset Address: 2E–2Fh      Attribute: RO  
 Default Value: 0000h      Size: 16 bits

Bit	Description
15:0	Subsystem ID—RO.

**Note:** The ICH2'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 of the EEPROM. The LAN Controller checks bits 15:13 in the EEPROM word Ah, and functions according to [Table 7-2](#).

**Table 7-2. 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	2449h	8086h	00h	0000h	0000h
01b	0b	2449h	8086h	00h	Word Bh	Word Ch
01b	1b	Word Bh	Word Ch	Word Ah, bits 10:8	Word Bh	Word Ch

### 7.1.15 CAP\_PTR—Capabilities Pointer (LAN Controller—B1:D8:F0)

Offset Address: 34h      Attribute: RO  
 Default Value: DCh      Size: 8 bits

Bit	Description
7:0	<b>Capabilities Pointer (CAP_PTR)</b> —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 Attribute: R/W  
Default Value: 00h Size: 8 bits

Bit	Description
7:0	<b>Interrupt Line (INT_LN)</b> —R/W. 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 Attribute: RO  
Default Value: 01h Size: 8 bits

Bit	Description
7:0	<b>Interrupt Pin (INT_PN)</b> —RO. Hardwired to 01h to indicate that the LAN Controller's interrupt request is connected to PIRQA#. However, in the ICH2 implementation, when the LAN Controller interrupt is generated PIRQ[E]# will go active, not PIRQ[A]#.

### 7.1.18 MIN\_GNT—Minimum Grant Register (LAN Controller—B1:D8:F0)

Offset Address: 3Eh Attribute: RO  
Default Value: 08h Size: 8 bits

Bit	Description
7:0	<b>Minimum Grant (MIN_GNT)</b> —RO. Indicates the amount of time (in increments of 0.25 $\mu$ s) 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 Attribute: RO  
Default Value: 38h Size: 8 bits

Bit	Description
7:0	<b>Maximum Latency (MAX_LAT)</b> —RO. Defines how often (in increments of 0.25 $\mu$ s) the LAN Controller needs to access the PCI bus.

### 7.1.20 CAP\_ID—Capability ID Register (LAN Controller—B1:D8:F0)

Offset Address: DCh      Attribute: RO  
 Default Value: 01h      Size: 8 bits

Bit	Description
7:0	<b>Capability ID (CAP_ID)</b> —RO. Hardwired to 01h to indicate that the ICH2's integrated LAN Controller supports PCI Power Management.

### 7.1.21 NXT\_PTR—Next Item Pointer (LAN Controller—B1:D8:F0)

Offset Address: DDh      Attribute: RO  
 Default Value: 00h      Size: 8 bits

Bit	Description
7:0	<b>Next Item Pointer (NXT_PTR)</b> —RW. Hardwired to 00b to indicate that power management is the last item in the Capabilities list.

### 7.1.22 PM\_CAP—Power Management Capabilities (LAN Controller—B1:D8:F0)

Offset Address: DE–DFh      Attribute: RO  
 Default Value: FE22h      Size: 16 bits

Bit	Description
15:11	<b>PME Support.</b> 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	<b>D2 Support.</b> Hardwired to 1 to indicate that the LAN Controller supports the D2 power state.
9	<b>D1 Support.</b> Hardwired to 1 to indicate that the LAN Controller supports the D1 power state.
8:6	<b>Auxiliary Current.</b> 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	<b>Device Specific Initialization (DSI).</b> 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	<b>PME Clock.</b> Hardwired to 0 to indicate that the LAN Controller does not require a clock to generate a power management event.
2:0	<b>Version.</b> 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 Attribute: RO, R/W, R/WC  
Default Value: 0000h Size: 16 bits

Bit	Description
15	<b>PME Status</b> —R/WC. 1 = Set upon occurrence of a wake-up event, independent of the state of the PME Enable bit. 0 = Software clears this bit by writing a 1 to the bit location. 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.
14:13	<b>Data Scale</b> —RO. This field indicates the data register scaling factor. It equals 10b for registers zero through eight and 00b for registers nine through fifteen, as selected by the "Data Select" field.
12:9	<b>Data Select</b> —R/W. This field is used to select which data is reported through the Data register and Data Scale field.
8	<b>PME Enable</b> —R/W. This bit enables the ICH2's integrated LAN controller to assert PME#. 1 = Enable PME# assertion when PME Status is set. 0 = The device will not assert PME#.
7:5	Reserved.
4	<b>Dynamic Data</b> —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	<b>Power State</b> —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: 00 = D0 01 = D1 10 = D2 11 = D3

### 7.1.24 DATA—Data Register (LAN Controller—B1:D8:F0)

Offset Address: E3h Attribute: RO  
Default Value: 00h Size: 8 bits

Bit	Description
7:0	<b>Data Value</b> . State dependent power consumption and heat dissipation data.

**Note:** The data register is an 8-bit read only register that provides a mechanism for the ICH2'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 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 7-3](#).

Table 7-3. 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)

Table 7-4. ICH2 Integrated LAN Controller CSR Space

Offset	Register Name/Function	Default	Type
01h–00h	SCB Status Word	0000h	R/WC
03h–02h	SCB Command Word	0000h	R/W
07h–04h	SCB General Pointer	0000 0000h	R/W
0Bh–08h	PORT	0000 0000h	R/W (special)
0Dh–0Ch	Reserved	—	—
0Eh	EEPROM Control Register	00	R/W
0Fh	Reserved	—	—
13h–10h	MDI Control Register	0000 0000h	R/W (special)
17h–14h	Receive DMA Byte Count	0000 0000h	RO
18h	Early Receive Interrupt	00h	R/W
1A–19h	Flow Control Register	0000h	R/W
1Bh	PMDR	00h	R/WC
1Ch	General Control	00	R/W
1Dh	General Status	N/A	RO
1Eh–3Ch	Reserved	—	—

## 7.2.1 System Control Block Status Word Register

Offset Address: 00–01h Attribute: R/WC, RO  
Default Value: 0000h Size: 16 bits

The ICH2's integrated LAN Controller places the status of its Command and Receive units and interrupt indications in this register for the processor to read.

Bit	Description																
15	<b>Command Unit (CU) Executed (CX)</b> —R/WC. 1 = Interrupt signaled because the CU has completed executing a command with its interrupt bit set. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.																
14	<b>Frame Received (FR)</b> —R/WC. 1 = Interrupt signaled because the Receive Unit (RU) has finished receiving a frame 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.																
13	<b>CU Not Active (CNA)</b> —R/WC. 1 = The Command Unit left the Active state or entered the Idle state. There are 2 distinct states of the CU. When configured to generate CNA interrupt, the interrupt will be activated when the CU leaves the Active state and enters either the Idle or the Suspended state. When configured to generate CI interrupt, an interrupt will be generated only when the CU enters the Idle state. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.																
12	<b>Receive Not Ready (RNR)</b> —R/WC. 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. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.																
11	<b>Management Data Interrupt (MDI)</b> —R/WC. 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). 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.																
10	<b>Software Interrupt (SWI)</b> —R/WC. 1 = Set when software generates an interrupt. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.																
9	<b>Early Receive (ER)</b> —R/WC. 1 = Indicates the occurrence of an Early Receive Interrupt. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.																
8	<b>Flow control Pause (FCP)</b> —R/WC. 1 = Indicates Flow Control Pause interrupt. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.																
7:6	<b>Command Unit Status (CUS)</b> —RO. 00 = Idle 01 = Suspended 10 = LPQ (Low Priority Queue) active 11 = HPQ (High Priority Queue) active																
5:2	<b>Receive Unit Status (RUS)</b> —RO. <table> <tr> <td>0000 = Idle</td><td>1000 = Reserved</td></tr> <tr> <td>0001 = Suspended</td><td>1001 = Suspended with no more RBDs</td></tr> <tr> <td>0010 = No Resources</td><td>1010 = No resources due to no more RBDs</td></tr> <tr> <td>0011 = Reserved</td><td>1011 = Reserved</td></tr> <tr> <td>0100 = Ready</td><td>1100 = Ready with no RBDs present</td></tr> <tr> <td>0101 = Reserved</td><td>1101 = Reserved</td></tr> <tr> <td>0110 = Reserved</td><td>1110 = Reserved</td></tr> <tr> <td>0111 = Reserved</td><td>1111 = Reserved</td></tr> </table>	0000 = Idle	1000 = Reserved	0001 = Suspended	1001 = Suspended with no more RBDs	0010 = No Resources	1010 = No resources due to no more RBDs	0011 = Reserved	1011 = Reserved	0100 = Ready	1100 = Ready with no RBDs present	0101 = Reserved	1101 = Reserved	0110 = Reserved	1110 = Reserved	0111 = Reserved	1111 = Reserved
0000 = Idle	1000 = Reserved																
0001 = Suspended	1001 = Suspended with no more RBDs																
0010 = No Resources	1010 = No resources due to no more RBDs																
0011 = Reserved	1011 = Reserved																
0100 = Ready	1100 = Ready with no RBDs present																
0101 = Reserved	1101 = Reserved																
0110 = Reserved	1110 = Reserved																
0111 = Reserved	1111 = Reserved																
1:0	Reserved.																

## 7.2.2 System Control Block Command Word Register

Offset Address: 02–03h      Attribute: R/W  
 Default Value: 0000h      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	<b>CX Mask</b> —R/W. 0 = Interrupt not masked. 1 = Disable the generation of a CX interrupt.
14	<b>FR Mask</b> —R/W. 0 = Interrupt not masked. 1 = Disable the generation of an FR interrupt.
13	<b>CNA Mask</b> —R/W. 0 = Interrupt not masked. 1 = Disable the generation of a CNA interrupt.
12	<b>RNR Mask</b> —R/W. 0 = Interrupt not masked. 1 = Disable the generation of an RNR interrupt.
11	<b>ER Mask</b> —R/W. 0 = Interrupt not masked. 1 = Disable the generation of an ER interrupt.
10	<b>FCP Mask</b> —R/W. 0 = Interrupt not masked. 1 = Disable the generation of an FCP interrupt.
9	<b>Software Generated Interrupt (SI)</b> —WO. 0 = No Effect. 1 = Setting this bit causes the LAN Controller to generate an interrupt.
8	<b>Interrupt Mask (IM)</b> —R/W. This bit enables or disables the LAN Controller's assertion of the INTA# signal. This bit has higher precedence than the Specific Interrupt Mask bits and the SI bit. 0 = Enable the assertion of INTA#. 1 = Disable the assertion of INTA#.

Bit	Description
7:4	<p><b>Command Unit Command (CUC).</b> Valid values are listed below. All other values are Reserved.</p> <p>0000 = <b>NOP:</b> Does not affect the current state of the unit.</p> <p>0001 = <b>CU Start:</b> 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.</p> <p>0010 = <b>CU Resume:</b> Resume operation of the Command unit by executing the next command. This command will be ignored if the CU is idle.</p> <p>0011 = <b>CU HPQ Start:</b> 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.</p> <p>0100 = <b>Load Dump Counters Address:</b> Tells 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.</p> <p>0101 = <b>Dump Statistical Counters:</b> Tells the device to dump its statistical counters to the area designated by the Load Dump Counters Address command.</p> <p>0110 = <b>Load CU Base:</b> The device's internal CU Base Register is loaded with the value in the CSB General Pointer.</p> <p>0111 = <b>Dump and Reset Statistical Counters:</b> Tells the device to dump its statistical counters to the area designated by the Load Dump Counters Address command, and then to clear these counters.</p> <p>1010 = <b>CU Static Resume:</b> 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.</p> <p>1011 = <b>CU HPQ Resume:</b> Resume execution of the first command on the HPQ CBL. this command will be ignored if the HPQ was never started.</p>
3	Reserved.
2:0	<p><b>Receive Unit Command (RUC).</b> Valid values are:</p> <p>000 = <b>NOP:</b> Does not affect the current state of the unit.</p> <p>001 = <b>RU Start:</b> 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.</p> <p>010 = <b>RU Resume:</b> Resume frame reception (only when in suspended state).</p> <p>011 = <b>RCV DMA Redirect:</b> 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).</p> <p>100 = <b>RU Abort:</b> Abort RU receive operation immediately.</p> <p>101 = <b>Load Header Data Size (HDS):</b> 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 zeros 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.</p> <p>110 = <b>Load RU Base:</b> The device's internal RU Base Register is loaded with the value in the SCB General Pointer.</p> <p>111 = <b>RBD Resume:</b> 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.</p>

### 7.2.3 System Control Block General Pointer Register

Offset Address: 04–07h      Attribute: R/W  
 Default Value: 0000 0000h      Size: 32 bits

Bit	Description
15:0	<b>SCB General Pointer.</b> 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 Register

Offset Address: 08–0Bh      Attribute: R/W (special)  
 Default Value: 0000 0000h      Size: 32 bits

The PORT interface allows the processor to reset the ICH2'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	<b>Pointer Field.</b> 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	<p><b>PORT Function Selection.</b> Valid values are listed below. All other values are Reserved.</p> <p>0000 = <b>PORT Software Reset:</b> 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 <math>\mu</math>s after issuing this command before attempting to access the LAN Controller's registers again.</p> <p>0001 = <b>Self Test:</b> 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 <a href="#">Table 7-5</a>. 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.</p> <p>0010 = <b>Selective Reset:</b> 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 <math>\mu</math>s after issuing this command before attempting to access the LAN Controller's registers again.</p>

**Table 7-5. Self-Test Results Format**

Bit	Description
31:13	Reserved
12	<b>General Self-Test Result.</b> 0 = Pass 1 = Fail
11:6	Reserved
5	<b>Diagnose Result.</b> This bit provides the result of an internal diagnostic test of the Serial Subsystem. 0 = Pass 1 = Fail
4	Reserved
3	<b>Register Result.</b> This bit provides the result of a test of the internal Parallel Subsystem registers. 0 = Pass 1 = Fail
2	<b>ROM Content Result.</b> This bit provides the result of a test of the internal microcode ROM. 0 = Pass 1 = Fail
1:0	Reserved

## 7.2.5 EEPROM Control Register

Offset Address: 0Eh      Attribute: RO, R/W  
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	<b>EEPROM Serial Clock (EESK)</b> —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. 0 = Drives the ICH2's EE_SHCLK signal low. 1 = Drives the ICH2's EE_SHCLK signal high.
2	<b>EEPROM Chip Select (EECS)</b> —R/W. 0 = Drives the ICH2's EE_CS signal low, to disable the EEPROM. this bit must be set to 0 for a minimum of 1μs between consecutive instruction cycles. 1 = Drives the ICH2's EE_CS signal high, to enable the EEPROM.
1	<b>EEPROM Serial Data In (EEDI)</b> —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.
0	<b>EEPROM Serial Data Out (EEDO)</b> —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.

## 7.2.6 Management Data Interface (MDI) Control Register

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	<b>Interrupt Enable.</b> 1 = Enables the LAN Controller to assert an interrupt to indicate the end of an MDI cycle. 0 = Disable.
28	<b>Ready.</b> 1 = Set by the LAN Controller at the end of an MDI transaction. 0 = Expected to be reset by software at the same time the command is written.
27:26	<b>Opcode.</b> These bits define the opcode: 00 = Reserved 01 = MDI write 10 = MDI read 11 = Reserved
25:21	<b>LAN Connect Address.</b> This field of bits contains the LAN Connect address.
20:16	<b>LAN Connect Register Address.</b> This field of bits contains the LAN Connect Register Address.
15:0	<b>Data.</b> 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 Receive DMA Byte Count Register

Offset Address: 14–17h      Attribute: RO  
 Default Value: 0000 0000h      Size: 32 bits

Bit	Description
31:0	<b>Receive DMA Byte Count</b> —RO. Keeps track of how many bytes of receive data have been passed into host memory via DMA.

## 7.2.8 Early Receive Interrupt Register

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 utilize 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	<b>Early Receive Count</b> —R/W. When some non-zero value <i>x</i> is programmed into this register, the LAN controller sets the ER bit in the SCB Status Word Register and assert INTA# when the byte count indicates that there are <i>x</i> 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 Control Register

Offset Address: 19–1Ah  
Default Value: 0000h

Attribute: RO, R/W (special)  
Size: 16 bits

Bit	Description																								
15:13	Reserved																								
12	<b>FC Paused Low</b> —RO. 1 = Set when the LAN Controller receives a Pause Low command with a value greater than zero. 0 = Cleared when the FC timer reaches zero, or a Pause frame is received.																								
11	<b>FC Paused</b> —RO. 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). 0 = Cleared when the FC timer reaches zero.																								
10	<b>FC Full</b> —RO. 1 = Set when the LAN Controller sends a Pause command with a value greater than zero. 0 = Cleared when the FC timer reaches zero.																								
9	<b>Xoff</b> —R/W (special). This bit should only be used if the LAN Controller is configured to operate with IEEE frame-based flow control. 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. 0 = This bit can only be cleared by writing a 1 to the Xon bit (bit 8 in this register).																								
8	<b>Xon</b> —WO. This bit should only be used if the LAN Controller is configured to operate with IEEE frame-based flow control. 1 = Writing a 1 to this bit resets the Xoff request to the LAN Controller, clearing bit 9 in this register. 0 = This bit always returns 0 on reads.																								
7:3	Reserved																								
2:0	<b>Flow Control Threshold</b> —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.  <table><tr><th colspan="3">Free Bytes</th></tr><tr><th>Bits 2:0</th><th>in Receive FIFO</th><th>Comment</th></tr><tr><td>000</td><td>0.50 KB</td><td rowspan="4">Fast system (recommended default)</td></tr><tr><td>001</td><td>1.00 KB</td></tr><tr><td>010</td><td>1.25 KB</td></tr><tr><td>011</td><td>1.50 KB</td></tr><tr><td>100</td><td>1.75 KB</td><td rowspan="4">Slow system</td></tr><tr><td>101</td><td>2.00 KB</td></tr><tr><td>110</td><td>2.25 KB</td></tr><tr><td>111</td><td>2.50 KB</td></tr></table>	Free Bytes			Bits 2:0	in Receive FIFO	Comment	000	0.50 KB	Fast system (recommended default)	001	1.00 KB	010	1.25 KB	011	1.50 KB	100	1.75 KB	Slow system	101	2.00 KB	110	2.25 KB	111	2.50 KB
Free Bytes																									
Bits 2:0	in Receive FIFO	Comment																							
000	0.50 KB	Fast system (recommended default)																							
001	1.00 KB																								
010	1.25 KB																								
011	1.50 KB																								
100	1.75 KB	Slow system																							
101	2.00 KB																								
110	2.25 KB																								
111	2.50 KB																								



## 7.2.10 Power Management Driver (PMDR) Register

Offset Address: 1Bh      Attribute: R/WC  
Default Value: 00h      Size: 8 bits

The ICH2's internal LAN Controller provides an indication in the PMDR that a wake-up event has occurred.

Bit	Description
7	<b>Link Status Change Indication</b> —R/WC. 1 = The link status change bit is set following a change in link status. 0 = Software clears this bit by writing a 1 to the bit location.
6	<b>Magic Packet</b> —R/WC. 1 = This bit is set when a Magic Packet is received regardless of the Magic Packet wake-up disable bit in the configuration command and the PME Enable bit in the Power Management Control/Status Register. 0 = Software clears this bit by writing a 1 to the bit location.
5	<b>Interesting Packet</b> —R/WC. 1 = This bit is set when an "interesting" packet is received. Interesting packets are defined by the LAN Controller packet filters. 0 = Software clears this bit by writing a 1 to the bit location.
4:1	Reserved.
0	<b>PME Status</b> —R/WC. This bit is a reflection of the PME Status bit in the Power Management Control/Status Register (PMCSR). 1 = Set upon a wake-up event, independent of the PME Enable bit. 0 = Software clears this bit by writing a 1 to the bit location. This also clears the PME Status bit in the PMCSR and deasserts the PME signal.

## 7.2.11 General Control Register

Offset Address: 1Ch      Attribute: R/W  
Default Value: 00h      Size: 8 bits

Bit	Description
7:4	Reserved. These bits should be set to 0000b.
3	<b>LAN Connect Software Reset</b> —R/W. 1 = Software can set this bit to force a reset condition on the LAN Connect interface. 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.
2	Reserved. This bit should be set to 0.
1	<b>Deep Power-Down on Link Down Enable.</b> 1 = Enable. The ICH2'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 = Disable
0	Reserved.

## 7.2.12 General Status Register

Offset Address: 1Dh      Attribute: RO  
 Default Value: N/A      Size: 8 bits

Bit	Description
7:3	Reserved.
2	<b>Duplex Mode.</b> This bit indicates the wire duplex mode. 1 = Full duplex 0 = Half duplex
1	<b>Speed.</b> This bit indicates the wire speed: 1 = 100 Mbps 0 = 10 Mbps
0	<b>Link Status Indication.</b> This bit indicates the status of the link: 1 = Valid 0 = Invalid

## 7.2.13 Statistical Counters

The ICH2'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 (i.e., 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 7-6. Statistical Counters**

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 deassertion 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.

**Table 7-6. Statistical Counters**

ID	Counter	Description
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 deasserts 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.
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 zero by the ICH2's integrated LAN Controller after reset. They cannot be preset to anything other than zero. 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 ICH2 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:** Registers that are not shown should be treated as Reserved (See [Section 6.2](#) for details).

**Table 8-1. PCI Configuration Map (HUB-PCI—D30:F0)**

Offset	Mnemonic	Register Name/Function	Default	Type
00–01h	VID	Vendor ID	8086h	RO
02–03h	DID	Device ID	244Eh	RO
04–05h	CMD	PCI Device Command Register	0001h	R/W
06–07h	PD_STS	PCI Device Status Register	0080h	R/W
08h	REVID	Revision ID	See Note	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	IO Base Register	F0h	R/W
1Dh	IOLIM	IO Limit Register	00h	R/W
1E–1Fh	SECSTS	Secondary Status Register	0280h	R/W
20–21h	MEMBASE	Memory Base	FFF0h	R/W
22–23h	MEMLIM	Memory Limit	0000h	R/W
24–25h	PREF_MEM_BASE	Prefetchable Memory Base	0000h	RO
26–27h	PREF_MEM_MLT	Prefetchable Memory Limit	0000h	RO
30–31h	IOBASE_HI	I/O Base Upper 16 Bits	0000h	RO
32–33h	IOLIMIT_HI	I/O Limit Upper 16 Bits	0000h	RO
3Ch	INT_LINE	Interrupt Line	00h	RO
3E–3Fh	BRIDGE_CNT	Bridge Control	0000h	R/W

Table 8-1. PCI Configuration Map (HUB-PCI—D30:F0) (Continued)

Offset	Mnemonic	Register Name/Function	Default	Type
40h	BRIDGE_CNT2	Bridge Control 2	00	R/W
50–51h	CNF	ICH2 Configuration Register	0000h	R/W
70h	MTT	Multi-Transaction Timer	20h	R/W
82h	PCI_MAST_STS	PCI Master Status	00h	R/W
90h	ERR_CMD	Error Command Register	00h	R/W
92h	ERR_STS	Error Status Register	00h	R/W

**NOTE:** Refer to the Specification Update for the value of the Revision ID Register

### 8.1.1 VID—Vendor ID Register (HUB-PCI—D30:F0)

Offset Address: 00–01h      Attribute: RO  
 Default Value: 8086h      Size: 16 bits

Bit	Description
15:0	<b>Vendor Identification Number</b> —RO. This is a 16-bit value assigned to Intel. Intel VID = 8086h.

### 8.1.2 DID—Device ID Register (HUB-PCI—D30:F0)

Offset Address: 02–03h      Attribute: RO  
 Default Value: 244Eh      Size: 16 bits

Bit	Description
15:0	<b>Device Identification Number</b> —RO. This is a 16 bit value assigned to the ICH2 hub interface to PCI bridge (i.e., Device #2). DID = 244Eh.

### 8.1.3 CMD—Command Register (HUB-PCI—D30:F0)

Offset Address: 04–05h Attribute: R/W  
Default Value: 0001h Size: 16 bits

Bit	Description
15:10	Reserved.
9	Fast Back to Back Enable (FBE)—RO. Hardwired to 0. The ICH2 does not support this capability.
8	<b>SERR# Enable (SERR_EN)</b> —R/W. 1 = Enable the ICH2 to generate an NMI (or SMI# if NMI routed to SMI#) when the D30:F0 SSE bit (offset 06h, bit 14) is set. 0 = Disable.
7	Wait Cycle Control—RO. Hardwired to 0
6	<b>Parity Error Response</b> —R/W. 1 = The ICH2 is allowed to report parity errors detected on the hub interface. 0 = The ICH2 will ignore parity errors on the hub interface.
5	VGA Palette Snoop—RO. Hardwired to 0.
4	Postable Memory Write Enable (PMWE)—RO. Hardwired to 0.
3	Special Cycle Enable (SCE)—RO. Hardwired to 0 by P2P Bridge specification.
2	<b>Bus Master Enable (BME)</b> —R/W. 1 = Allows the Hub interface-to-PCI bridge to accept cycles from PCI to run on the hub interface. Note: This bit does not affect the CF8h and CFCh I/O accesses. 0 = Disable
1	<b>Memory Space Enable (MSE)</b> —R/W. The ICH2 provides this bit as read/writable for software only. However, the ICH2 ignores the programming of this bit, and runs hub interface memory cycles to PCI.
0	<b>I/O Space Enable (IOE)</b> —R/W. The ICH2 provides this bit as read/writable for software only. However, the ICH2 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 PD\_STS—Primary Device Status Register (HUB-PCI—D30:F0)

Offset Address: 06–07h Attribute: R/WC  
 Default Value: 0080h Size: 16 bits

For the writable bits in this register, writing a 1 will clear the bit. Writing a 0 to the bit will have no effect.

Bit	Description
15	<b>Detected Parity Error (DPE)</b> —R/WC. 1 = Indicates that the ICH2 detected a parity error on the hub interface. This bit gets set even if the Parity Error Response bit (offset 04, bit 6) is not set. 0 = Software clears this bit by writing a 1 to the bit location.
14	<b>Received System Error (SSE)</b> —R/WC. 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 ICH2 will generate an NMI (or SMI# if NMI routed to SMI#) 0 = Software clears this bit by writing a 1 to the bit location.
13	<b>Received Master Abort (RMA)</b> —R/WC. 1 = ICH2 received a master abort from the hub interface device. 0 = Software clears this bit by writing a 1 to the bit location.
12	<b>Received Target Abort (RTA)</b> —R/WC. 1 = ICH2 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. 0 = Software clears this bit by writing a 1 to the bit location.
11	<b>Signaled Target Abort (STA)</b> —R/WC. 1 = ICH2 signals a target abort condition on the hub interface. 0 = Software clears this bit by writing a 1 to the bit location.
10:9	<b>DEVSEL# Timing Status</b> —RO. 00h = Fast timing. This register applies to the hub interface; therefore, this field does not matter.
8	<b>Data Parity Error Detected (DPD)</b> —R/WC. Since this register applies to the hub interface, the ICH2 must interpret this bit differently than it is in the PCI specification. 1 = ICH2 detects a parity error on the hub interface and the Parity Error Response bit in the Command Register (offset 04h, bit 6) is set. 0 = Software clears this bit by writing a 1 to the bit location.
7	Fast Back to Back—RO. Hardwired to 1.
6	User Definable Features (UDF)—RO. Hardwired to 0.
5	66 MHz Capable—RO. Hardwired to 0.
4:0	Reserved.

## 8.1.5 REVID—Revision ID Register (HUB-PCI—D30:F0)

Offset Address: 08h Attribute: RO  
 Default Value: See bit description Size: 8 bits

Bit	Description
7:0	<b>Revision Identification Number</b> —RO. 8-bit value that indicates the revision number for the ICH2 hub interface to PCI bridge. Refer to the Specification Update for the value of the Revision ID 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	<b>Sub-Class Code</b> —RO. This 8-bit value indicates the category of bridge for the ICH2 hub interface to PCI bridge. The code is 04h indicating a 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	<b>Base Class Code</b> —RO. This 8-bit value indicates the type of device for the ICH2 hub interface to PCI bridge. The code is 06h indicating a 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 Count. Not implemented.
2:0	Reserved.

### 8.1.9 HEADTYP—Header Type Register (HUB-PCI—D30:F0)

Offset Address:	0Eh	Attribute:	RO
Default Value:	01h	Size:	8 bits

Bit	Description
7	Multi-function Device—RO. This bit is 0 to indicate a single function device.
6:0	Header Type—RO. 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                      Attribute: RO  
 Default Value: 00h                      Size: 8 bits

Bit	Description
7:0	<b>Primary Bus Number</b> —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                      Attribute: R/W  
 Default Value: 00h                      Size: 8 bits

Bit	Description
7:0	<b>Secondary Bus Number</b> —R/W. This field indicates the bus number of PCI. Note that when this number is equal to the primary bus number (i.e., bus #0), the ICH2 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                      Attribute: R/W  
 Default Value: 00h                      Size: 8 bits

Bit	Description
7:0	<b>Subordinate Bus Number</b> —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 ICH2 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 ICH2 continues to burst data as a master on the PCI bus. When the ICH2 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 results in the deassertion of FRAME#. If the internal grant has not been removed, the ICH2 can continue to own the bus.

Bit	Description
7:3	<b>Master Latency Count</b> —R/W. This 5-bit value indicates the number of PCI clocks, in 8-clock increments, that the ICH2 remains as master of the bus.
2:0	Reserved.



#### 8.1.14 IOBASE—I/O Base Register (HUB-PCI—D30:F0)

Offset Address: 1Ch  
Default Value: F0h

Attribute:	R/W
Size:	8 bits

Bit	Description
7:4	<b>I/O Address Base bits [15:12]</b> —R/W. I/O Base bits corresponding to address lines 15:12 for 4 KB alignment. Bits 11:0 are assumed to be padded to 000h.
3:0	<b>I/O Addressing Capability</b> —RO. This 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 Register and I/O Limit Upper Address registers must be read only.

### 8.1.15 IOLIM—I/O Limit Register (HUB-PCI—D30:F0)

Offset Address: 1Dh  
Default Value: 00h

Attribute:	R/W
Size:	8 bits

Bit	Description
7:4	<b>I/O Address Limit bits [15:12]</b> —R/W. I/O Base bits corresponding to address lines 15:12 for 4 KB alignment. Bits 11:0 are assumed to be padded to FFFh.
3:0	<b>I/O Addressing Capability</b> —RO. This 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 Register and I/O 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/W  
 Default Value: 0280h      Size: 16 bits

For the writable bits in this register, writing a 1 will clear the bit. Writing a 0 to the bit will have no effect.

Bit	Description
15	<b>Detected Parity Error (DPE)</b> —R/WC. 1 = ICH2 detected a parity error on the PCI bus. 0 = Software clears this bit by writing a 1 to the bit position.
14	<b>Received System Error (SSE)</b> —R/WC. 1 = SERR# assertion is received on PCI. 0 = Software clears this bit by writing a 1 to the bit position.
13	<b>Received Master Abort (RMA)</b> —R/WC. 1 = Hub interface to PCI cycle is master-aborted on PCI. 0 = Software clears this bit by writing a 1 to the bit position.
12	<b>Received Target Abort (RTA)</b> —R/WC. 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 ICH2 must send the “target abort” status back to the hub interface. 0 = Software clears this bit by writing a 1 to the bit position.
11	Signaled Target Abort (STA)—RO. The ICH2 does not generate target aborts.
10:9	DEVSEL# Timing Status—RO. 01h = Medium timing.
8	<b>Data Parity Error Detected (DPD)</b> —R/WC. 1 = The ICH2 sets this bit when all of the following three conditions are met: <ul style="list-style-type: none"> <li>- The Parity Error Response Enable bit in the Bridge Control Register (bit 0, offset 3Eh) is set</li> <li>- USB, AC'97 or IDE is a Master</li> <li>- PERR# asserts during a write cycle <b>OR</b> a parity error is detected internally during a read cycle</li> </ul> 0 = Software clears this bit by writing a 1 to the bit position.
7	Fast Back to Back—RO. Hardwired to 1 to indicate that the PCI to hub interface target logic is capable of receiving fast back-to-back cycles.
6	User Definable Features (UDF)—RO. Hardwired to 0.
5	66 MHz Capable—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 ICH2 forwards all hub interface memory accesses to PCI, the ICH2 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	<b>Memory Address Base</b> —R/W. 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 ICH2 will forward all hub interface memory accesses to PCI, the ICH2 will only use 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 FFFFh. Thus, the top of the defined memory address range will be aligned to a 1 MB boundary.

Bit	Description
15:4	<b>Memory Address Limit</b> —R/W. 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: 24h–25h      Attribute: R/W  
Default Value: 0000FFF0h      Size: 16-bit

Bit	Description
15:4	<b>Prefetchable Memory Address Base</b> —R/W. Defines the base address of the prefetchable memory address range for PCI. These 12 bits correspond to address bits 31:20.
3:0	Reserved. RO.

### 8.1.20 PREF\_MEM\_MLT—Prefetchable Memory Limit Register (HUB-PCI—D30:F0)

Offset Address: 26h–27h Attribute: R/W  
Default Value: 00000000h Size: 16-bit

Bit	Description
15:4	<b>Prefetchable Memory Address Limit</b> —RW. Defines the limit address of the prefetchable memory address range for PCI. These 12 bits correspond to address bits 31:20.
3:0	Reserved. RO

### 8.1.21 IOBASE\_HI—I/O Base Upper 16 Bits Register (HUB-PCI—D30:F0)

Offset Address: 30–31h Attribute: RO  
Default Value: 0000h 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 Attribute: RO  
Default Value: 0000h 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\_LINE—Interrupt Line Register (HUB-PCI—D30:F0)

Offset Address: 3Ch Attribute: RO  
Default Value: 00h Size: 8 bits

Bit	Description
7:0	Interrupt Line Routing—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      Attribute: R/W  
Default Value: 0000h      Size: 16 bits

Bit	Description
15:8	Reserved.
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 ICH2 does not follow the P2P bridge reset scheme; Software-controlled resets are implemented in the PCI-LPC device.
5	<b>Master Abort Mode</b> —R/W. The ICH2 ignores this bit. However, this bit is read/write for software compatibility. The ICH2 must handle master aborts as if this bit is reset to 0.
4	Reserved.
3	<b>VGA Enable</b> —R/W. 1 = Enable. 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 ICH2 will never take I/O cycles in the VGA range from PCI. 0 = No VGA device on PCI.
2	<b>ISA Enable</b> —R/W. The ICH2 ignores this bit. However, this bit is read/write for software compatibility. Since the ICH2 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	<b>SERR# Enable</b> —R/W. 1 = Enable. If this bit is set AND bit 8 in CMD register (D30:F0 Offset 04h) is also set, the ICH2 sets the SSE bit in PD_STS 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. 0 = Disable
0	<b>Parity Error Response Enable</b> —R/W. 1 = Enable the hub interface to PCI bridge for parity error detection and reporting on the PCI bus. 0 = Disable

## 8.1.25 BRIDGE\_CNT2—Bridge Control Register 2 (HUB-PCI—D30:F0)

Offset Address: 40h      Attribute: R/W  
Default Value: 00h      Size: 8 bits

Bit	Description
7:1	Reserved
0	<b>PCI_DAC_EN</b> —R/W. Allows ICH2 to recognize external PCI masters performing DAC on PCI. 0 = Disable. 1 = Enable.

## 8.1.26 CNF—ICH2 Configuration Register (HUB-PCI—D30:F0)

Offset Address: 50–51h      Attribute: R/W  
 Default Value: 0000h      Size: 16 bits

Bit	Description
15:10	Reserved.
9	<b>HP_PCI_EN</b> —R/W. High Priority PCI Enable. 1 = Enables a mode where the REQ[0]#/GNT[0]# signal pair has a higher arbitration priority. 0 = All PCI REQ#/GNT pairs have the same arbitration priority.
8	<b>Hole Enable (15 MB–16 MB)</b> —R/W. 1 = Enables the 15 MB to 16 MB hole in the DRAM. 0 = Disable
7:2	Reserved.
1	<b>32-Clock Retry Enable</b> —R/W. System BIOS must set this bit for PCI compliance. 1 = When a PCI device is running a locked memory read cycle, while all other bus masters are waiting to run locked cycles, concurrent with a LPC DMA transfer, this bit, when set allows the ICH2 to retry the locked memory read cycle. 0 = If this bit is not set, under the same circumstance, the bus will not be released since all other masters see the lock in use.
0	Reserved.

## 8.1.27 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 ICH2's arbiter allows a PCI initiator to perform multiple back-to-back transactions on the PCI bus. The ICH2'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 grants 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, 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 MTT to a value of 00h disables this function, which could cause starvation issues 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 time-out before the Grant-to-FRAME# trigger causing a re-arbitration.

Bit	Description
7:3	<b>Multi-Transaction Timer Count Value</b> —R/W. This field specifies the amount of time that grant remains 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.

### 8.1.28 PCI\_MAST\_STS—PCI Master Status Register (HUB-PCI—D30:F0)

Offset Address: 82h Attribute: R/W  
Default Value: 00h Size: 8 bits

Bit	Description
7	<b>Internal PCI Master Request Status (INT_MREQ_STS)</b> —R/WC. 1 = The ICH2's internal DMA controller or LPC has requested use of the PCI bus. 0 = Software clears this bit by writing a 1 to the bit position.
6	<b>Internal LAN Master Request Status (LAN_MREQ_STS)</b> —R/WC. 1 = The ICH2's internal LAN controller has requested use of the PCI bus. 0 = Software clears this bit by writing a 1 to the bit position.
5:0	<b>PCI Master Request Status (PCI_MREQ_STS)</b> —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 ICH2 has detected REQ[0]# asserted and bit 5 will be set if ICH2 detected REQ[5]# asserted. 1 = The associated PCI master has requested use of the PCI bus. 0 = Software clears these bits by writing a 1 to the bit position.

### 8.1.29 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 ICH2'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	<b>SERR# enable on receiving target abort (SERR_RTA_EN)</b> —R/W. 1 = Enable. When SERR_EN is set, the ICH2 will report SERR# when SERR_RTA is set. 0 = Disable
1	<b>SERR# enable on Delayed Transaction Time-out (SERR_DTT_EN)</b> —R/W. 1 = Enable. When SERR_EN is set, the ICH2 will report SERR# when SERR_DTT is set. 0 = Disable.
0	Reserved.

### 8.1.30 ERR\_STS—Error Status Register (HUB-PCI—D30:F0)

Offset Address:	92h	Attribute:	R/W
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.

Bit	Description
7:3	Reserved.
2	<b>SERR# Due to Received Target Abort (SERR_RTA)—R/W.</b> 1 = The ICH2 sets this bit when the ICH2 receives a target abort. If SERR_EN, the ICH2 will also generate an SERR# when SERR_RTA is set. 0 = This bit is cleared by writing a 1.
1	<b>SERR# Due to Delayed Transaction Time-out (SERR_DTT)—R/W.</b> 1 = When a PCI master does not return for the data within 1 ms of the cycle's completion, the ICH2 clears the delayed transaction, and sets this bit. If both SERR_DTT_EN and SERR_EN are set, then ICH2 will also generate an SERR# when SERR_DTT is set. 0 = This bit is cleared by writing a 1.
0	Reserved.

# LPC Interface Bridge Registers (D31:F0)9

The LPC Bridge function of the ICH2 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 (power management, GPIO, USB, IDE, etc.) are described in their respective sections.

## 9.1 PCI Configuration Registers (D31:F0)

**Note:** Registers that are not shown should be treated as Reserved (See [Section 6.2](#) for details).

**Table 9-1. PCI Configuration Map (LPC I/F—D31:F0)**

Offset	Mnemonic	Register Name	Default	Type
00h–01h	VID	Vendor ID	8086h	RO
02h–03h	DID	Device ID	2440h	RO
04h–05h	PCICMD	PCI Command Register	000Fh	R/W
06h–07h	PCISTS	PCI Device Status Register	0280h	R/W
08h	RID	Revision ID	See Note	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	01h	RO
0Bh	BCC	Base Class Code	06h	RO
0Eh	HEADT	Header Type	80h	RO
40h–43h	PMBASE	ACPI Base Address Register	00000001h	R/W
44h	ACPI_CNTL	ACPI Control	00h	R/W
4Eh–4Fh	BIOS_CNTL	BIOS Control Register	0000h	R/W
54h	TCO_CNTL	TCO Control	00h	R/W
58h–5Bh	GPIO_BASE	GPIO Base Address Register	00000001h	R/W
5Ch	GPIO_CNTL	GPIO Control Register	00h	R/W
60h–63h	PIRQ[n]_ROUT	PIRQ[A–D] Routing Control	80808080h	R/W
64h	SIRQ_CNTL	Serial IRQ Control Register	10h	R/W
68h–6Bh	PIRQ[n]_ROUT	PIRQ[E–H] Routing Control	80808080h	R/W
88h	D31_ERR_CFG	Device 31 Error configuration Register	00h	R/W
8Ah	D31_ERR_STS	Device 31 Error Status Register	00h	R/W
90h–91h	PCI_DMA_C	PCI DMA Configuration Registers	0000h	R/W
A0h–CFh		Power Management Registers See <a href="#">Section 9.8.1</a>		
D0h–D3h	GEN_CNTL	General Control	00000000h	R/W
D4h–D7h	GEN_STS	General Status	00000F00h	R/W

Table 9-1. PCI Configuration Map (LPC I/F—D31:F0) (Continued)

Offset	Mnemonic	Register Name	Default	Type
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 & LPT Decode Ranges	00h	R/W
E2h	SND_DEC	LPC I/F Sound Decode Ranges	00h	R/W
E3h	FWH_DEC_EN1	FWH Decode Enable 1	FFh	R/W
E4h–E5h	GEN1_DEC	LPC I/F General 1 Decode Range	0000h	R/W
E6h–E7h	LPC_EN	LPC I/F Enables	00h	R/W
E8h–EBh	FWH_SEL1	FWH Select 1	00112233h	R/W
ECh–EDh	GEN2_DEC	LPC I/F General 2 Decode Range	0000h	R/W
EEh–EFh	FWH_SEL2	FWH Select 2	5678h	R/W
F0h	FWH_DEC_EN2	FWH Decode Enable 2	0Fh	R/W
F2h	FUNC_DIS	Function Disable Register	00h	R/W

**NOTE:** Refer to the Specification Update for the value of the Revision ID Register.

### 9.1.1 VID—Vendor ID Register (LPC I/F—D31:F0)

Offset Address: 00–01h      Attribute: RO  
 Default Value: 8086h      Size: 16-bit  
 Lockable: No      Power Well: Core

Bit	Description
15:0	<b>Vendor ID Value.</b> This is a 16 bit value assigned to Intel. Intel VID = 8086h

### 9.1.2 DID—Device ID Register (LPC I/F—D31:F0)

Offset Address: 02–03h      Attribute: RO  
 Default Value: 2440h      Size: 16-bit  
 Lockable: No      Power Well: Core

Bit	Description
15:0	<b>Device ID Value.</b> This is a 16 bit value assigned to the ICH2 LPC Bridge.



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

Offset Address: 04–05h  
Default Value: 000Fh  
Lockable: No

Attribute: R/W  
Size: 16-bit  
Power Well: Core

Bit	Description
15:10	Reserved.
9	Fast Back to Back Enable (FBE)—RO. Hardwired to 0.
8	<b>SERR# Enable (SERR_EN)</b> —R/W. 1 = Enable. Allow SERR# to be generated. 0 = Disable.
7	Wait Cycle Control (WCC)—RO. Hardwired to 0.
6	<b>Parity Error Response (PER)</b> —R/W. 1 = The ICH will take normal action when a parity error is detected. 0 = No action is taken when detecting a parity error.
5	VGA Palette Snoop (VPS)—RO. Hardwired to 0
4	Postable Memory Write Enable (PMWE)—RO. Hardwired to 0
3	Special Cycle Enable (SCE). Hardwired to 1.
2	Bus Master Enable (BME)—RO. Hardwired to 1 to indicate that bus mastering can not be disabled for function 0 (DMA/ISA Master).
1	Memory Space Enable (MSE)—RO. Hardwired to 1 to indicate that memory space can not be disabled for Function 0 (LPC I/F).
0	I/O Space Enable (IOE)—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 Device Status (LPC I/F—D31:F0)

Offset Address: 06–07h      Attribute: R/WC  
 Default Value: 0280h      Size: 16-bit  
 Lockable: No      Power Well: Core

Bit	Description
15	<b>Detected Parity Error (DPE)</b> —R/W. 1 = PERR# signal goes active. Set even if the PER bit is 0. 0 = This bit is cleared by software writing a 1 to the bit position.
14	<b>Signaled System Error (SSE)</b> —R/W. 1 = Set by the ICH2 if the SERR_EN bit is set and the ICH2 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. 0 = This bit is cleared by software writing a 1 to the bit position.
13	<b>Master Abort Status (RMA)</b> —R/W. 1 = ICH2 generated a master abort on PCI due to LPC I/F master or DMA cycles. 0 = This bit is cleared by software writing a 1 to the bit position.
12	<b>Received Target Abort (RTA)</b> —R/W. 1 = ICH2 received a target abort during LPC I/F master or DMA cycles to PCI. 0 = This bit is cleared by software writing a 1 to the bit position.
11	<b>Signaled Target Abort (STA)</b> —R/W. 1 = ICH2 generated a target abort condition on PCI cycles claimed by the ICH2 for ICH2 internal registers or for going to LPC I/F. 0 = This bit is cleared by software writing a 1 to the bit position.
10:9	<b>DEVSEL# Timing Status (DEV_STS)</b> —RO. 01 = Medium Timing.
8	<b>Data Parity Error Detected (DPED)</b> —R/WC. 1 = Set when all three of the following conditions are true: - The ICH2 is the initiator of the cycle, - The ICH2 asserted PERR# (for reads) or observed PERR# (for writes), and - The PER bit is set. 0 = This bit is cleared by software writing a 1 to the bit position.
7	<b>Fast Back to Back (FB2B)</b> —RO. Always 1. Indicates ICH2 as a target can accept fast back-to-back transactions.
6	<b>User Definable Features (UDF)</b> . Hardwired to 0
5	<b>66 MHz Capable (66MHZ_CAP)</b> —RO. Hardwired to 0
4:0	Reserved.

### 9.1.5 REVID—Revision ID Register (LPC I/F—D31:F0)

Offset Address: 08h      Attribute: RO  
 Default Value: See bit description      Size: 8 bits

Bit	Description
7:0	<b>Revision Identification Number.</b> 8-bit value that indicates the revision number for the LPC bridge. For the A-0 stepping, this value is 00h. Refer to the Specification Update for the value of the Revision ID Register

### 9.1.6 PI—Programming Interface (LPC I/F—D31:F0)

Offset Address: 09h      Attribute: RO  
Default Value: 00h      Size: 8 bits

Bit	Description
7:0	Programming Interface Value.

### 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	<b>Sub-Class Code.</b> This 8-bit value 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	<b>Base Class Code.</b> This 8-bit value indicates the type of device for the LPC bridge. The code is 06h indicating a 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	<b>Multi-function Device</b> —RO. This bit is 1 to indicate a multi-function device.
6:0	<b>Header Type</b> —RO. This 8-bit field identifies the header layout of the configuration space.

### 9.1.10 PMBASE—ACPI Base Address (LPC I/F—D31:F0)

Offset Address:	40–43h	Attribute:	R/W
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. Can be mapped anywhere in the 64 KB I/O space on 128-byte boundaries.

Bit	Description
31:16	Reserved.
15:7	<b>Base Address</b> —R/W. 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	<b>Resource Indicator</b> —RO. Tied to 1 to indicate I/O space.

### 9.1.11 ACPI\_CNTL—ACPI Control (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	<b>ACPI Enable (ACPI_EN)</b> —R/W. 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. 0 = Disable.
3	Reserved.
2:0	<b>SCI IRQ Select (SCI_IRQ_SEL)</b> —R/W. Specifies on which IRQ the SCI will internally appear. If not using the APIC, the SCI must be routed to IRQ[9:11], 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 IRQ[20:23], and can be shared with other interrupts. 000 = IRQ9 001 = IRQ10 010 = IRQ11 011 = Reserved 100 = IRQ20 (Only available if APIC enabled) 101 = IRQ21 (Only available if APIC enabled) 110 = RQ22 (Only available if APIC enabled) 111 = IRQ23 (Only available if APIC enabled)

### 9.1.12 BIOS\_CNTL (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	<b>BIOS Lock Enable (BLE)</b> —R/W. 1 = Enables setting the BIOSWE bit to cause SMLs. 0 = Setting the BIOSWE will not cause SMLs. Once set, this bit can only be cleared by a PCIRST#.
0	<b>BIOS Write Enable (BIOSWE)</b> —R/W. 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 SMM code can update BIOS. 0 = Only read cycles result in FWH interface cycles.

### 9.1.13 TCO\_CNTL—TCO Control (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	<b>TCO Interrupt Enable (TCO_INT_EN)</b> —R/W. This bit enables/disables the TCO interrupt. 1 = Enables TCO Interrupt, as selected by the TCO_INT_SEL field. 0 = Disables TCO interrupt.
2:0	<b>TCO Interrupt Select (TCO_INT_SEL)</b> —R/W. Specifies which IRQ the TCO internally appears. If not using the APIC, the TCO interrupt must be routed to IRQ[9:11], 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 IRQ[20:23], and can be shared with other interrupt. Note that if the TCOSCI_EN bit is set (bit 6 in the GPE0_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 10 or 11, the signal is, in fact, active high. When the TCO interrupt is mapped to IRQ[20, 21, or 22], the signal is active low and can be shared with PCI interrupts that may be mapped to the same signals (IRQs). 000 = IRQ9 001 = IRQ10 010 = IRQ11 011 = Reserved 100 = IRQ20 (Only available if APIC enabled) 101 = IRQ21 (Only available if APIC enabled) 110 = IRQ22 (Only available if APIC enabled) 111 = IRQ23 (Only available if APIC enabled)

### 9.1.14 GPIOBASE—GPIO Base Address (LPC I/F—D31:F0)

Offset Address: 58h–5Bh      Attribute: R/W  
 Default Value: 00000001h      Size: 32-bit  
 Lockable: No      Power Well: Core

Bit	Description
31:16	Reserved.
15:6	<b>Base Address</b> —R/W. Provides the 64 bytes of I/O space for GPIO.
5:1	Reserved.
0	<b>Resource Indicator</b> —RO. Tied to 1 to indicate I/O space.

### 9.1.15 GPIO\_CNTL—GPIO Control (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	<b>GPIO Enable (GPIO_EN)</b> —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. 1 = Enable 0 = Disable
3:0	Reserved.

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

Offset Address: PIRQA–60h, PIRQB–61h,      Attribute: R/W  
 PIRQC–62h, PIRQD–63h  
 Default Value: 80h      Size: 8-bit  
 Lockable: No      Power Well: Core

Bit	Description
7	<b>Interrupt Routing Enable (IRQEN)</b> —R/W. Note that 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. 0 = The corresponding PIRQ is routed to one of the ISA-compatible interrupts specified in bits[3:0]. 1 = The PIRQ is not routed to the 8259.
6:4	Reserved.
3:0	<b>IRQ Routing</b> —R/W. (ISA compatible) 0000 = Reserved      1000 = Reserved 0001 = Reserved      1001 = IRQ9 0010 = Reserved      1010 = IRQ10 0011 = IRQ3      1011 = IRQ11 0100 = IRQ4      1100 = IRQ12 0101 = IRQ5      1101 = Reserved 0110 = IRQ6      1110 = IRQ14 0111 = IRQ7      1111 = IRQ15

### 9.1.17 SERIRQ\_CNTL—Serial IRQ Control (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	<b>Serial IRQ Enable (SIRQEN)</b> —R/W. 1 = Serial IRQs will be recognized. The SERIRQ pin will be configured as SERIRQ. 0 = The buffer is input only and internally SERIRQ will be a 1.
6	<b>Serial IRQ Mode Select (SIRQMD)</b> —R/W. 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 ICH2 not recognizing SERIRQ interrupts. 1 = The serial IRQ machine will be in continuous mode. 0 = The serial IRQ machine will be in quiet mode.
5:2	<b>Serial IRQ Frame Size (SIRQSZ)</b> —R/W. Fixed field that indicates the size of the SERIRQ frame. In the ICH2, 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	<b>Start Frame Pulse Width (SFPW)</b> —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 ICH2 will drive the start frame for the number of clocks specified. In quiet mode, the ICH2 will drive the start frame for the number of clocks specified minus one, as the first clock was driven by the peripheral. 00 = 4 clocks 01 = 6 clocks 10 = 8 clocks 11 = Reserved

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

Offset Address: PIRQE—68h, PIRQF—69h, Attribute: R/W  
PIRQG—6Ah, PIRQH—6Bh  
Default Value: 80h Size: 8-bit  
Lockable: No Power Well: Core

Bit	Description																
7	<b>Interrupt Routing Enable (IRQEN)</b> —R/W. Note that 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. 0 = The corresponding PIRQ is routed to one of the ISA-compatible interrupts specified in bits[3:0]. 1 = The PIRQ is not routed to the 8259.																
6:4	Reserved.																
3:0	<b>IRQ Routing</b> —R/W. (ISA compatible) <table> <tr> <td>0000 = Reserved</td><td>1000 = Reserved</td></tr> <tr> <td>0001 = Reserved</td><td>1001 = IRQ9</td></tr> <tr> <td>0010 = Reserved</td><td>1010 = IRQ10</td></tr> <tr> <td>0011 = IRQ3</td><td>1011 = IRQ11</td></tr> <tr> <td>0100 = IRQ4</td><td>1100 = IRQ12</td></tr> <tr> <td>0101 = IRQ5</td><td>1101 = Reserved</td></tr> <tr> <td>0110 = IRQ6</td><td>1110 = IRQ14</td></tr> <tr> <td>0111 = IRQ7</td><td>1111 = IRQ15</td></tr> </table>	0000 = Reserved	1000 = Reserved	0001 = Reserved	1001 = IRQ9	0010 = Reserved	1010 = IRQ10	0011 = IRQ3	1011 = IRQ11	0100 = IRQ4	1100 = IRQ12	0101 = IRQ5	1101 = Reserved	0110 = IRQ6	1110 = IRQ14	0111 = IRQ7	1111 = IRQ15
0000 = Reserved	1000 = Reserved																
0001 = Reserved	1001 = IRQ9																
0010 = Reserved	1010 = IRQ10																
0011 = IRQ3	1011 = IRQ11																
0100 = IRQ4	1100 = IRQ12																
0101 = IRQ5	1101 = Reserved																
0110 = IRQ6	1110 = IRQ14																
0111 = IRQ7	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 ICH2'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	<b>SERR# on Received Target Abort Enable (SERR_RTA_EN)—R/W.</b> 1 = The ICH2 will generate SERR# when SERR_RTA is set if SERR_EN is set. 0 = Disable. No SERR# assertion on Received Target Abort.
1	<b>SERR# on Delayed Transaction Time-out Enable (SERR_DTT_EN)—R/W.</b> 1 = The ICH2 will generate SERR# when SERR_DTT bit is set if SERR_EN is set. 0 = Disable. No SERR# assertion on Delayed Transaction Time-out.
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 ICH2'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	<b>SERR# Due to Received Target Abort (SERR_RTA)—R/WC.</b> 1 = The ICH2 sets this bit when it receives a target abort. If SERR_EN, the ICH2 will also generate an SERR# when SERR_RTA is set. 0 = Software clears this bit by writing a 1 to the bit location.
1	<b>SERR# Due to Delayed Transaction Time-out (SERR_DTT)—R/WC.</b> 1 = When a PCI master does not return for the data within 1 ms of the cycle's completion, the ICH2 clears the delayed transaction and sets this bit. If both SERR_DTT_EN and SERR_EN are set, then ICH2 will also generate an SERR# when SERR_DTT is set. 0 = Software clears this bit by writing a 1 to the bit location.
0	Reserved.

### 9.1.21 PCI\_DMA\_CFG—PCI DMA Configuration (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	<b>Channel 7 Select</b> —R/W. 00 = Reserved 01 = PC/PCI DMA 10 = Reserved 11 = LPC I/F DMA
13:12	<b>Channel 6 Select</b> —R/W. Same bit decode as for Channel 7
11:10	<b>Channel 5 Select</b> —R/W. Same bit decode as for Channel 7
9:8	Reserved.
7:6	<b>Channel 3 Select</b> —R/W. Same bit decode as for Channel 7
5:4	<b>Channel 2 Select</b> —R/W. Same bit decode as for Channel 7
3:2	<b>Channel 1 Select</b> —R/W. Same bit decode as for Channel 7
1:0	<b>Channel 0 Select</b> —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
Default Value:	00000000h	Size:	32-bit
Lockable:	No	Power Well:	Core

Bit	Description
31:26	Reserved.
25	<b>REQ[5]#/GNT[5]# PC/PCI protocol select (PCPCIB_SEL)</b> —R/W. 1 = When this bit is set to a 1, the PCI REQ[5]#/GNT[5]# signal pair will use the PC/PCI protocol as REQ[B]#/GNT[B]. 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. 0 = The REQ[5]#/GNT[5]# pins will function as a standard PCI REQ/GNT signal pair.
24	<b>Hide ISA Bridge (HIDE_ISA)</b> —R/W. 1 = Software sets this bit to 1 to disable configuration cycle from being claimed by a PCI-to-ISA bridge. This prevents the operating system PCI PnP from getting confused by seeing two ISA bridges. It is required for the ICH2 PCI address line AD22 to connect to the PCI-to-ISA bridge's IDSEL input. When this bit is 1, the ICH2 does not assert AD22 during configuration cycles to the PCI-to-ISA bridge. 0 = The ICH2 does not prevent AD22 from asserting during configuration cycles to the PCI-to-ISA bridge.
23:14	Reserved.
13	<b>Coprocessor Error Enable (COPR_ERR_EN)</b> —R/W. 1 = When FERR# is low, ICH2 generates IRQ13 internally and holds it until an I/O write to port F0h. It will also drive IGNNE# active. 0 = FERR# will not generate IRQ13 nor IGNNE#.
12	<b>Keyboard IRQ1 Latch Enable (IRQ1LEN)</b> —R/W. 1 = The active edge of IRQ1 will be latched and held until a port 60h read. 0 = IRQ1 will bypass the latch.

Bit	Description
11	<b>Mouse IRQ12 Latch Enable (IRQ12LEN)</b> —R/W. 1 = The active edge of IRQ12 will be latched and held until a port 60h read. 0 = IRQ12 will bypass the latch.
10:9	Reserved
8 <sup>1</sup>	<b>APIC Enable (APIC_EN)</b> —R/W. 1 = Enables the internal I/O (x) APIC and its address decode. 0 = Disables internal I/O (x) APIC.
7 <sup>1</sup>	<b>Enables I/O (x) Extension Enable (XAPIC_EN)</b> —R/W. Note that this bit is only valid if the APIC_EN bit (bit 8) is also set to 1. 1 = Enables the extra features (beyond standard I/O APIC) associated with the I/O (x) APIC. 0 = The I/O (x) APIC extensions are not supported.
6	<b>Alternate Access Mode Enable (ALTACC_EN)</b> —R/W. 1 = Alternate Access Mode Enable 0 = Alternate Access Mode Disabled (default). Alternate Access Mode allows reads to otherwise unreadable registers and writes otherwise unwritable registers.
5:3	Reserved.
2	<b>DMA Collection Buffer Enable (DCB_EN)</b> —R/W. 1 = Enables DMA Collection Buffer (DCB) for LPC I/F and PC/PCI DMA. 0 = DCB disabled.
1	<b>Delayed Transaction Enable (DTE)</b> —R/W. 1 = ICH2 enables delayed transactions for internal register, FWH and LPC interface accesses. 0 = Delayed transactions disabled.
0	<b>Positive Decode Enable (POS_DEC_EN)</b> —R/W. 1 = Enables ICH2 to only perform positive decode on the PCI bus. 0 = The ICH2 performs subtractive decode on the PCI bus and forward the cycles to LPC interface if not to an internal register or other known target on the LPC interface. Accesses to internal registers and to known LPC interface devices are still be positively decoded.

**NOTES:**

- Rule 1:** If bit 8 is 0, the ICH2 does not decode any of the registers associated with the I/O APIC or I/O (x) APIC. The state of bit 7 is a “Don’t Care” in this case.

**Rule 2:** If bit 8 is 1 and bit 7 is 0, the ICH2 decodes the memory space associated with the I/O APIC, but not the extra registers associated with the I/O (x) APIC.

**Rule 3:** If bit 8 is 1 and bit 7 is 1, the ICH2 decodes 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).

Note that there is no separate way to disable PCI Message Interrupts if the I/O (x) APIC is enabled. This is not considered necessary.



### 9.1.23 GEN\_STS—General Status (LPC I/F—D31:F0)

Offset Address: D4h–D7h      Attribute: R/W  
Default Value: 0000F0Xh      Size: 32-bit  
Lockable: No      Power Well: Core(0:7), RTC (8:15)

Bit	Description
31:14	Reserved.
13	<b>TOP_SWAP</b> —R/W. 1 = ICH2 will invert A16 for cycles targeting FWH BIOS space (Does not affect accesses to FWH feature space). 0 = ICH2 will not invert A16. This bit is cleared by RTCRST# assertion, but not by any other type of reset.
12	<b>CPU BIST Enable (CPU_BIST_EN)</b> —R/W. 1 = The INIT# signal is driven active when CPURST# is active. INIT# goes 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; however, the ICH2 has a hub interface special cycle that allows the ICH2 to control the assertion/deassertion of CPURST#. 0 = Disable.
11:8	<b>Processor Frequency Strap (FREQ_STRAP[3:0])</b> —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. These bits are in the RTC well. This field is only writeable when SAFE_MODE (bit 2) is cleared to 0. SAFE_MODE is only cleared by a PWROK rising edge.
7:3	Reserved
2	<b>SAFE_MODE</b> —RO. 1 = ICH2 sampled AC_SDOUT high on the rising edge of PWROK. ICH2 will force FREQ_STRAP[3:0] bits to all 1s (safe mode multiplier). 0 = ICH2 sampled AC_SDOUT low on the rising edge of PWROK.
1	<b>NO_REBOOT</b> —R/W (special). 1 = ICH2 will disable the TCO Timer system reboot feature. This bit is set either by hardware when SPKR is sampled low on the rising edge of PWROK or by software writing a 1 to the bit. 0 = Normal TCO Timer reboot functionality (reboot after 2nd TCO time-out). Note that this bit cannot be cleared while an external jumper is in place on the SPKR signal.
0	Reserved.

### 9.1.24 RTC\_CONF—RTC Configuration Register (LPC I/F—D31:F0)

Offset Address: D8h      Attribute: R/W  
 Default Value: 00h      Size: 8-bit  
 Lockable: Yes      Power Well: Core

Bit	Description
7:5	Reserved.
4	<b>Upper 128-byte Lock (U128LOCK)</b> —R/W (special). 1 = Lock 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. 0 = Access to these bytes in the upper CMOS RAM range have not been locked.
3	<b>Lower 128-byte Lock (L128LOCK)</b> —R/W (special). 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. 0 = Access to these bytes in the lower CMOS RAM range have not been locked.
2	<b>Upper 128-byte Enable (U128E)</b> —R/W. 1 = Enables access to the upper 128 byte bank of RTC CMOS RAM. 0 = Disable.
1:0	Reserved.

### 9.1.25 COM\_DEC—LPC I/F Communication Port Decode Ranges (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	<b>COMB Decode Range</b> —R/W. This field determines which range to decode for the COMB Port. 000 = 3F8h–3FFh (COM1) 001 = 2F8h–2FFh (COM2) 010 = 220h–227h 011 = 228h–22Fh 100 = 238h–23Fh 101 = 2E8h–2EFh (COM4) 110 = 338h–33Fh 111 = 3E8h–3EFh (COM3)
3	Reserved
2:0	<b>COMA Decode Range</b> —R/W. This field determines which range to decode for the COMA Port. 000 = 3F8h–3FFh (COM1) 001 = 2F8h–2FFh (COM2) 010 = 220h–227h 011 = 228h–22Fh 100 = 238h–23Fh 101 = 2E8h–2EFh (COM4) 110 = 338h–33Fh 111 = 3E8h–3EFh (COM3)

## 9.1.26 FDD/LPT\_DEC—LPC I/F FDD & LPT Decode Ranges (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	<b>FDD Decode Range</b> —R/W. Determines which range to decode for the FDD Port 0 = 3F0h–3F5h, 3F7h (Primary) 1 = 370h–2FFh (Secondary)
3:2	Reserved
1:0	<b>LPT Decode Range</b> —R/W. This field determines which range to decode for the LPT Port. 00 = 378h–37Fh and 778h–77Fh 01 = 278h–27Fh (port 279h is read only) and 678h–67Fh 10 = 3BCh–3BEh and 7BCh–7BEh 11 = Reserved

## 9.1.27 SND\_DEC—LPC I/F Sound Decode Ranges (LPC I/F—D31:F0)

Offset Address: E2h Attribute: R/W  
 Default Value: 00h Size: 8-bit  
 Lockable: No Power Well: Core

Bit	Description
7:6	Reserved
5:4	<b>MSS Decode Range</b> —R/W. This field determines which range to decode for the Microsoft* Sound System (MSS). 00 = 530h–537h 01 = 604h–60Bh 10 = E80h–E87h 11 = F40h–F47h
3	<b>MIDI Decode Range</b> —R/W. This bit determines which range to decode for the Midi Port. 0 = 330h–331h 1 = 300h–301h
2	Reserved
1:0	<b>SB16 Decode Range</b> —R/W. This field determines which range to decode for the Sound Blaster 16 (SB16) Port. 00 = 220h–233h 01 = 240h–253h 10 = 260h–273h 11 = 280h–293h

### 9.1.28 FWH\_DEC\_EN1—FWH Decode Enable 1 Register (LPC I/F—D31:F0)

Offset Address: E3h  
Default Value: FFh

Attribute: R/W  
Size: 8 bits

This register determines which memory ranges will be decoded on the PCI bus and forwarded to the FWH. The ICH2 will subtractively decode cycles on PCI unless POS\_DEC\_EN is set to 1.

Bit	Description
7	<b>FWH Address Range Enable (FWH_F8_EN)</b> —RO. Enables decoding two 512 KB FWH memory ranges and one 128 KB memory range. 1 = Enable the following ranges for the FWH FFF80000h–FFFFFFFFh FFB80000h–FFBFFFFFFh 000E0000h–000FFFFFFh
6	<b>FWH Address Range Enable (FWH_F0_EN)</b> —R/W. Enables decoding two 512 KB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH: FFF00000h–FFF7FFFFh FFB00000h–FFB7FFFFh
5	<b>FWH Address Range Enable (FWH_E8_EN)</b> —R/W. Enables decoding two 512 KB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH: FFE80000h–FFEFFFFFFh FFA80000h–FFAFFFFFFh
4	<b>FWH Address Range Enable (FWH_E0_EN)</b> —R/W. Enables decoding two 512 KB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH: FFE00000h–FFE7FFFFh FFA00000h–FFA7FFFFh
3	<b>FWH Address Range Enable (FWH_D8_EN)</b> —R/W. Enables decoding two 512 KB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FFD80000h–FFDFFFFFFh FF980000h–FF9FFFFFFh
2	<b>FWH Address Range Enable (FWH_D0_EN)</b> —R/W. Enables decoding two 512 KB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FFD00000h–FFD7FFFFh FF900000h–FF97FFFFh
1	<b>FWH Address Range Enable (FWH_C8_EN)</b> —R/W. Enables decoding two 512 KB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FFC80000h–FFCFFFFFFh FF880000h–FF8FFFFFFh
0	<b>FWH Address Range Enable (FWH_C0_EN)</b> —R/W. Enables decoding two 512 KB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FFC00000h–FFC7FFFFh FF800000h–FF87FFFFh

### 9.1.29 GEN1\_DEC—LPC I/F Generic Decode Range 1 (LPC I/F—D31:F0)

Offset Address: E4h–E5h Attribute: R/W  
Default Value: 00h Size: 16-bit  
Lockable: Yes Power Well: Core

Bit	Description
15:7	<b>Generic I/O Decode Range 1 Base Address (GEN1_BASE)</b> —R/W. This address is aligned on a 128-byte boundary, and must have address lines 31:16 as 0. 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	<b>Generic Decode Range 1 Enable (GEN1_EN)</b> —R/W. 0 = Disable. 1 = Enable the GEN1 I/O range to be forwarded to the LPC I/F

### 9.1.30 LPC\_EN—LPC I/F Enables (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	<b>Microcontroller Address Range Enable (CNF2_LPC_EN)</b> —R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 4Eh and 4Fh to the LPC interface. This range is used for a microcontroller.
12	<b>Super I/O Address Range Enable (CNF1_LPC_EN)</b> —R/W. 0 = Disable. 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	<b>Microcontroller Address Range Enable (MC_LPC_EN)</b> —R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 62h and 66h to the LPC interface. This range is used for a microcontroller.
10	<b>Microcontroller Address Range Enable (KBC_LPC_EN)</b> —R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 60h and 64h to the LPC interface. This range is used for a microcontroller.
9	<b>Game Port Address Range Enable (GAMEH_LPC_EN)</b> —R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 208h to 20Fh to the LPC interface. This range is used for a gameport.
8	<b>Game Port Address Range Enable (GAMEL_LPC_EN)</b> —R/W. 0 = Disable. 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	<b>ADLIB Address Range Enable (ADLIB_LPC_EN)</b> —R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 388h–38Bh to the LPC interface.
6	<b>MSS Address Range Enable (MSS_LPC_EN)</b> —R/W. 0 = Disable. 1 = Enables the decoding of the MSS range to the LPC interface. This range is selected in the LPC_Sound Decode Range Register.
5	<b>MIDI Address Range Enable (MIDI_LPC_EN)</b> —R/W. 0 = Disable. 1 = Enables the decoding of the MIDI range to the LPC interface. This range is selected in the LPC_Sound Decode Range Register.
4	<b>Sound Blaster Address Range Enable (SB16_LPC_EN)</b> —R/W. 0 = Disable. 1 = Enables the decoding of the SB16 range to the LPC interface. This range is selected in the LPC_Sound Decode Range Register.
3	<b>FDD Address Range Enable (FDD_LPC_EN)</b> —R/W. 0 = Disable. 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	<b>LPT Address Range Enable (LPT_LPC_EN)</b> —R/W. 0 = Disable. 1 = Enables the decoding of the LPT range to the LPC interface. This range is selected in the LPC_FDD/LPT Decode Range Register.
1	<b>COM B Address Range Enable (COMB_LPC_EN)</b> —R/W. 0 = Disable. 1 = Enables the decoding of the COMB range to the LPC interface. This range is selected in the LPC_COM Decode Range Register.
0	<b>Com A Address Range Enable (COMA_LPC_EN)</b> —R/W. 0 = Disable. 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 FWH\_SEL1—FWH Select 1 Register (LPC I/F—D31:F0)

Offset Address: E8h Attribute: R/W  
Default Value: 00112233h Size: 32 bits

Bit	Description
31:28	<b>FWH Address Range Select (FWH_F8_IDSEL)</b> —RO. IDSEL for two 512 KB FWH memory ranges and one 128KB memory range. This field is fixed at 0000. The IDSEL in this field addresses the following memory ranges: FFF8 0000h–FFFF FFFFh FFB8 0000h–FFBF FFFFh 000E 0000h–000F FFFFh
27:24	<b>FWH Address Range Select (FWH_F0_IDSEL)</b> —R/W. IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFF0 0000h–FFF7 FFFFh FFB0 0000h–FFB7 FFFFh
23:20	<b>FWH Address Range Select (FWH_E8_IDSEL)</b> —R/W. IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFE8 0000h–FFE7 FFFFh FFA8 0000h–FFAF FFFFh
19:16	<b>FWH Address Range Select (FWH_E0_IDSEL)</b> —R/W. IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFE0 0000h–FFE7 FFFFh FFA0 0000h–FFA7 FFFFh
15:12	<b>FWH Address Range Select (FWH_D8_IDSEL)</b> —R/W. IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFD8 0000h–FFDF FFFFh FF98 0000h–FF9F FFFFh
11:8	<b>FWH Address Range Select (FWH_D0_IDSEL)</b> —R/W. IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFD0 0000h–FFD7 FFFFh FF90 0000h–FF97 FFFFh
7:4	<b>FWH Address Range Select (FWH_C8_IDSEL)</b> —R/W. IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFC8 0000h–FFCF FFFFh FF88 0000h–FF8F FFFFh
3:0	<b>FWH Address Range Select (FWH_C0_IDSEL)</b> —R/W. IDSEL for two 512 KB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFC0 0000h–FFC7 FFFFh FF80 0000h–FF87 FFFFh

### 9.1.32 GEN2\_DEC—LPC I/F Generic Decode Range 2 (LPC I/F—D31:F0)

Offset Address: ECh–EDh      Attribute: R/W  
 Default Value: 00h      Size: 16-bit  
 Lockable: Yes      Power Well: Core

Bit	Description
15:4	<b>Generic I/O Decode Range 2 Base Address (GEN2_BASE)</b> —R/W. This address is aligned on a 64-byte boundary and must have address lines 31:16 as 0. 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	<b>Generic I/O Decode Range 2 Enable (GEN2_EN)</b> —R/W. 0 = Disable. 1 = Accesses to the GEN2 I/O range will be forwarded to the LPC interface.

### 9.1.33 FWH\_SEL2—FWH 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	<b>FWH Address Range Select (FWH_70_IDSEL)</b> —R/W. IDSEL for two 1 MB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF70 0000h–FF7F FFFFh FF30 0000h–FF3F FFFFh
11:8	<b>FWH Address Range Select (FWH_60_IDSEL)</b> —R/W. IDSEL for two 1 MB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF60 0000h–FF6F FFFFh FF20 0000h–FF2F FFFFh
7:4	<b>FWH Address Range Select (FWH_50_IDSEL)</b> —R/W. IDSEL for two 1 MB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF50 0000h–FF5F FFFFh FF10 0000h–FF1F FFFFh
3:0	<b>FWH Address Range Select (FWH_40_IDSEL)</b> —R/W. IDSEL for two 1 MB FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF40 0000h–FF4F FFFFh FF00 0000h–FF0F FFFFh

### 9.1.34 FWH\_DEC\_EN2—FWH 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 are decoded on the PCI bus and forwarded to the FWH. The ICH2 subtractively decodes cycles on PCI unless POS\_DEC\_EN is set to 1.

Bit	Description
7:4	Reserved.
3	<b>FWH Address Range Enable (FWH_70_EN)</b> —R/W. Enables decoding two 1 MB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FF70 0000h–FF7F FFFFh FF30 0000h–FF3F FFFFh
2	<b>FWH Address Range Enable (FWH_60_EN)</b> —R/W. Enables decoding two 1 MB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FF60 0000h–FF6F FFFFh FF20 0000h–FF2F FFFFh
1	<b>FWH Address Range Enable (FWH_50_EN)</b> —R/W. Enables decoding two 1 MB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FF50 0000h–FF5F FFFFh FF10 0000h–FF1F FFFFh
0	<b>FWH Address Range Enable (FWH_40_EN)</b> —R/W. Enables decoding two 1 MB FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FF40 0000h–FF4F FFFFh 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-bit
Lockable:	No	Power Well:	Core

Bit	Description
15:9	Reserved
8	<b>SMBus For BIOS (SMB_FOR_BIOS)</b> —R/W. This bit is used in conjunction with bit 3 in this register. 0 = No effect. 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.
7	Reserved
6	<b>AC'97 Modem Disable (F6_Disable)</b> —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 for functions that are to be disabled. 0 = AC'97 Modem is enabled 1 = AC'97 Modem is disabled
5	<b>AC'97 Audio Controller Disable (F5_Disable)</b> —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 for functions that are to be disabled. 0 = AC'97 audio controller is enabled 1 = AC'97 audio controller is disabled
4	<b>USB Controller 2 Disable (F4_Disable)</b> —R/W. Software sets this bit to disable the USB Controller #2 function. BIOS must not enable I/O or memory address space decode, interrupt generation or any other functionality for functions that are to be disabled. 0 = USB Controller #2 is enabled 1 = USB Controller #2 is disabled
3	<b>SMBus Controller Disable (F3_Disable)</b> —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 for functions that are to be disabled. 0 = SMBus controller is enabled 1 = SMBus controller is disabled
2	<b>USB Controller 1 Disable (F2_Disable)</b> —R/W. Software sets this bit to disable the USB Controller #1 function. BIOS must not enable I/O or memory address space decode, interrupt generation or any other functionality for functions that are to be disabled. 0 = USB Controller #1 is enabled 1 = USB Controller #1 is disabled
1	<b>IDE Controller Disable (F1_Disable)</b> —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 for functions that are to be disabled. 0 = IDE controller is enabled 1 = IDE controller is disabled
0	Reserved.

## 9.2 DMA I/O Registers

**Table 9-2. DMA Registers**

Port	Alias	Register Name/Function	Default	Type
00h	10h	Channel 0 DMA Base & Current Address Register	Undefined	R/W
01h	11h	Channel 0 DMA Base & Current Count Register	Undefined	R/W
02h	12h	Channel 1 DMA Base & Current Address Register	Undefined	R/W
03h	13h	Channel 1 DMA Base & Current Count Register	Undefined	R/W
04h	14h	Channel 2 DMA Base & Current Address Register	Undefined	R/W
05h	15h	Channel 2 DMA Base & Current Count Register	Undefined	R/W
06h	16h	Channel 3 DMA Base & Current Address Register	Undefined	R/W
07h	17h	Channel 3 DMA Base & Current Count Register	Undefined	R/W
08h	18h	Channel 0–3 DMA Command Register	Undefined	WO
		Channel 0–3 DMA Status Register	Undefined	RO
0Ah	1Ah	Channel 0–3 DMA Write Single Mask Register	000001XXb	WO
0Bh	1Bh	Channel 0–3 DMA Channel Mode Register	000000XXb	WO
0Ch	1Ch	Channel 0–3 DMA Clear Byte Pointer Register	Undefined	WO
0Dh	1Dh	Channel 0–3 DMA Master Clear Register	Undefined	WO
0Eh	1Eh	Channel 0–3 DMA Clear Mask Register	Undefined	WO
0Fh	1Fh	Channel 0–3 DMA Write All Mask Register	0Fh	R/W
80h	90h	Reserved Page Register	Undefined	R/W
81h	91h	Channel 2 DMA Memory Low Page Register	Undefined	R/W
82h	–	Channel 3 DMA Memory Low Page Register	Undefined	R/W
83h	93h	Channel 1 DMA Memory Low Page Register	Undefined	R/W
84h–86h	94h–96h	Reserved Page Registers	Undefined	R/W
87h	97h	Channel 0 DMA Memory Low Page Register	Undefined	R/W
88h	98h	Reserved Page Register	Undefined	R/W
89h	99h	Channel 6 DMA Memory Low Page Register	Undefined	R/W
8Ah	9Ah	Channel 7 DMA Memory Low Page Register	Undefined	R/W
8Bh	9Bh	Channel 5 DMA Memory Low Page Register	Undefined	R/W
8Ch–8Eh	9Ch–9Eh	Reserved Page Registers	Undefined	R/W
8Fh	9Fh	Refresh Low Page Register	Undefined	R/W
C0h	C1h	Channel 4 DMA Base & Current Address Register	Undefined	R/W
C2h	C3h	Channel 4 DMA Base & Current Count Register	Undefined	R/W
C4h	C5h	Channel 5 DMA Base & Current Address Register	Undefined	R/W
C6h	C7h	Channel 5 DMA Base & Current Count Register	Undefined	R/W
C8h	C9h	Channel 6 DMA Base & Current Address Register	Undefined	R/W
CAh	CBh	Channel 6 DMA Base & Current Count Register	Undefined	R/W
CCh	CDh	Channel 7 DMA Base & Current Address Register	Undefined	R/W

Table 9-2. DMA Registers (Continued)

Port	Alias	Register Name/Function	Default	Type
CEh	CFh	Channel 7 DMA Base & Current Count Register	Undefined	R/W
D0h	D1h	Channel 4–7 DMA Command Register	Undefined	WO
		Channel 4–7 DMA Status Register	Undefined	RO
D4h	D5h	Channel 4–7 DMA Write Single Mask Register	000001XXb	WO
D6h	D7h	Channel 4–7 DMA Channel Mode Register	000000XXb	WO
D8h	D9h	Channel 4–7 DMA Clear Byte Pointer Register	Undefined	WO
DAh	DBh	Channel 4–7 DMA Master Clear Register	Undefined	WO
DCh	DDh	Channel 4–7 DMA Clear Mask Register	Undefined	WO
DEh	DFh	Channel 4–7 DMA Write All Mask Register	0Fh	R/W

## 9.2.1 DMABASE\_CA—DMA Base and Current Address Registers

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:	RO
Default Value:	Undef	Size:	16-bit (per channel), but accessed in two 8-bit quantities
Lockable:	No	Power Well:	Core

Bit	Description
15:0	<p><b>Base and Current Address</b>—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 <i>Base Address</i> register and copied to the <i>Current Address</i> register. On reads, the value is returned from the <i>Current Address</i> register.</p> <p>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.</p> <p>For transfers to/from a 16-bit slave (channels 5–7), the address is shifted left one bit location. Bit 15 will be shifted out. Therefore, if bit 15 was a 1, it will be lost.</p> <p>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.</p>

## 9.2.2 DMABASE\_CC—DMA Base and Current Count Registers

I/O Address:	Ch. #0 = 01h; Ch. #1 = 03h	Attribute:	R/W
	Ch. #2 = 05h; Ch. #3 = 07h		
Default Value:	Ch. #5 = C6h; Ch. #6 = CAh	Size:	16-bit (per channel), but accessed in two 8-bit quantities
	Ch. #7 = CEh;		
Lockable:	Undefined	Power Well:	Core
	No		

Bit	Description
15:0	<p><b>Base and Current Count</b>—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 <i>Base Count</i> register and copied to the <i>Current Count</i> register. On reads the value is returned from the <i>Current Count</i> register.</p> <p>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 zero 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.</p> <p>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.</p> <p>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.</p>

## 9.2.3 DMAMEM\_LP—DMA Memory Low Page Registers

I/O Address:	Ch. #0 = 87h; Ch. #1 = 83h	Attribute:	R/W
	Ch. #2 = 81h; Ch. #3 = 82h		
Default Value:	Ch. #5 = 8Bh; Ch. #6 = 89h	Size:	8-bit
	Ch. #7 = 8Ah;		
Lockable:	Undefined	Power Well:	Core
	No		

Bit	Description
7:0	<p><b>DMA Low Page (ISA Address bits [23:16])</b>—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.</p>

## 9.2.4 DMACMD—DMA Command Register

I/O Address: Ch. #0–3 = 08h;  
 Ch. #4–7 = D0h  
 Default Value: Undefined  
 Lockable: No

Attribute: WO  
 Size: 8-bit  
 Power Well: Core

Bit	Description
7:5	Reserved. Must be 0.
4	<b>DMA Group Arbitration Priority</b> —WO. Each channel group is individually assigned either fixed or rotating arbitration priority. At part reset, each group is initialized in fixed priority. 0 = Fixed priority to the channel group 1 = Rotating priority to the group.
3	Reserved. Must be 0
2	<b>DMA Channel Group Enable</b> —WO. Both channel groups are enabled following part reset. 0 = Enable the DMA channel group. 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 DMASTS—DMA Status Register

I/O Address: Ch. #0–3 = 08h;  
 Ch. #4–7 = D0h  
 Default Value: Undefined  
 Lockable: No

Attribute: RO  
 Size: 8-bit  
 Power Well: Core

Bit	Description
7:4	<b>Channel Request Status</b> —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. 4 = Channel 0 5 = Channel 1 (5) 6 = Channel 2 (6) 7 = Channel 3 (7)
3:0	<b>Channel Terminal Count Status</b> —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. 0 = Channel 0 1 = Channel 1 (5) 2 = Channel 2 (6) 3 = Channel 3 (7)

## 9.2.6 DMA\_WRSMSK—DMA Write Single Mask Register

I/O Address:	Ch. #0–3 = 0Ah; 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	<b>Channel Mask Select</b> —WO. 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. 1 = Disable DREQ for the selected channel.
1:0	<b>DMA Channel Select</b> —WO. These bits select the DMA Channel Mode Register to program. 00 = Channel 0 (4) 01 = Channel 1 (5) 10 = Channel 2 (6) 11 = Channel 3 (7)

## 9.2.7 DMACH\_MODE—DMA Channel Mode Register

I/O Address:	Ch. #0–3 = 0Bh; Ch. #4–7 = D6h	Attribute:	WO
Default Value:	0000 00xx	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:6	<b>DMA Transfer Mode</b> —WO. Each DMA channel can be programmed in one of four different modes: 00 = Demand mode 01 = Single mode 10 = Reserved 11 = Cascade mode
5	<b>Address Increment/Decrement Select</b> —WO. This bit controls address increment/decrement during DMA transfers. 0 = Address increment. (default after part reset or Master Clear) 1 = Address decrement.
4	<b>Autoinitialize Enable</b> —WO. 0 = Autoinitialize feature is disabled and DMA transfers terminate on a terminal count. A part reset or Master Clear disables autoinitialization. 1 = DMA restores the Base Address and Count registers to the current registers following a terminal count (TC).
3:2	<b>DMA Transfer Type</b> —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. 00 = Verify - No I/O or memory strobes generated 01 = Write - Data transferred from the I/O devices to memory 10 = Read - Data transferred from memory to the I/O device 11 = Illegal
1:0	<b>DMA Channel Select</b> —WO. These bits select the DMA Channel Mode Register that will be written by bits [7:2]. 00 = Channel 0 (4) 01 = Channel 1 (5) 10 = Channel 2 (6) 11 = Channel 3 (7)

## 9.2.8 DMA Clear Byte Pointer Register

I/O Address: Ch. #0–3 = 0Ch;  
 Ch. #4–7 = D8h  
 Default Value: xxxx xxxx  
 Lockable: No

Attribute: WO  
 Size: 8-bit  
 Power Well: Core

Bit	Description
7:0	<b>Clear Byte Pointer</b> —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

I/O Address: Ch. #0–3 = 0Dh;  
 Ch. #4–7 = DAh  
 Default Value: xxxx xxxx

Attribute: WO  
 Size: 8-bit

Bit	Description
7:0	<b>Master Clear</b> —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

I/O Address: Ch. #0–3 = 0Eh;  
 Ch. #4–7 = DCh  
 Default Value: xxxx xxxx  
 Lockable: No

Attribute: WO  
 Size: 8-bit  
 Power Well: Core

Bit	Description
7:0	<b>Clear Mask Register</b> —WO. No specific pattern. Command enabled with a write to the port.

## 9.2.11 DMA\_WRMSK—DMA Write All Mask Register

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	<p><b>Channel Mask Bits</b>—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).</p> <p>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.</p> <p>Bit 0 = Channel 0 (4) 1 = Masked, 0 = Not Masked            Bit 1 = Channel 1 (5) 1 = Masked, 0 = Not Masked            Bit 2 = Channel 2 (6) 1 = Masked, 0 = Not Masked            Bit 3 = Channel 3 (7) 1 = Masked, 0 = Not Masked</p> <p><b>Note:</b> Disabling channel 4 also disables channels 0–3 due to the cascade of channels 0–3 through channel 4.</p>

## 9.3 Timer I/O Registers

Port	Aliases	Register Name/Function	Default Value	Type
40h	50h	Counter 0 Interval Time Status Byte Format	0XXXXXXXb	RO
		Counter 0 Counter Access Port Register	Undefined	R/W
41h	51h	Counter 1 Interval Time Status Byte Format	0XXXXXXXb	RO
		Counter 1 Counter Access Port Register	Undefined	R/W
42h	52h	Counter 2 Interval Time Status Byte Format	0XXXXXXXb	RO
		Counter 2 Counter Access Port Register	Undefined	R/W
43h	53h	Timer Control Word Register	Undefined	WO
		Timer Control Word Register Read Back	XXXXXXXX0b	WO
		Counter Latch Command	X0h	WO

### 9.3.1 TCW—Timer Control Word Register

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	<b>Counter Select</b> —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. 00 = Counter 0 select 01 = Counter 1 select 10 = Counter 2 select 11 = Read Back Command
5:4	<b>Read/Write Select</b> —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). 00 = Counter Latch Command 01 = Read/Write Least Significant Byte (LSB) 10 = Read/Write Most Significant Byte (MSB) 11 = Read/Write LSB then MSB
3:1	<b>Counter Mode Selection</b> —WO. These bits select one of six possible modes of operation for the selected counter. 000 = Mode 0      Out signal on end of count (=0) 001 = Mode 1      Hardware retriggerable one-shot x10 = Mode 2      Rate generator (divide by n counter) x11 = Mode 3      Square wave output 100 = Mode 4      Software triggered strobe 101 = Mode 5      Hardware triggered strobe
0	<b>Binary/BCD Countdown Select</b> —WO. 0 = Binary countdown is used. The largest possible binary count is $2^{16}$ 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

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	<b>Read Back Command.</b> This field must be “11” to select the Read Back Command.
5	<b>Latch Count of Selected Counters.</b> 0 = Current count value of the selected counters will be latched 1 = Current count will not be latched
4	<b>Latch Status of Selected Counters.</b> 0 = Status of the selected counters will be latched 1 = Status will not be latched
3	<b>Counter 2 Select.</b> 1 = Counter 2 count and/or status will be latched
2	<b>Counter 1 Select.</b> 1 = Counter 1 count and/or status will be latched
1	<b>Counter 0 Select.</b> 1 = Counter 0 count and/or status will be latched.
0	Reserved. Must be 0.

### 9.3.1.2 LTCH\_CMD—Counter Latch Command

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	<b>Counter Selection.</b> These bits select the counter for latching. If “11” is written, then the write is interpreted as a read back command. 00 = Counter 0 01 = Counter 1 10 = Counter 2
5:4	<b>Counter Latch Command.</b> 00 = Selects the Counter Latch Command.
3:0	Reserved. Must be 0.

### 9.3.2 SBYTE\_FMT—Interval Timer Status Byte Format Register

I/O Address:	Counter 0 = 40h, Counter 1 = 41h, Counter 2 = 42h	Attribute:	RO
Default Value:	Bits[6:0] undefined, Bit 7=0	Size:	8 bits per counter

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	<b>Counter OUT Pin State</b> —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. 0 = OUT pin of the counter is also a 0. 1 = OUT pin of the counter is also a 1.
6	<b>Count Register Status</b> —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. 0 = Count has been transferred from CR to CE and is available for reading. 1 = Null Count. Count has not been transferred from CR to CE and is not yet available for reading.
5:4	<b>Read/Write Selection Status</b> —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. 00 = Counter Latch Command 01 = Read/Write Least Significant Byte (LSB) 10 = Read/Write Most Significant Byte (MSB) 11 = Read/Write LSB then MSB
3:1	<b>Mode Selection Status</b> —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. 000 = Mode 0 Out signal on end of count (=0) 001 = Mode 1 Hardware retriggerable one-shot x10 = Mode 2 Rate generator (divide by n counter) x11 = Mode 3 Square wave output 100 = Mode 4 Software triggered strobe 101 = Mode 5 Hardware triggered strobe
0	<b>Countdown Type Status</b> —RO. This bit reflects the current countdown type. 0 = Binary countdown 1 = Binary Coded Decimal (BCD) countdown.

### 9.3.3 Counter Access Ports Register

I/O Address:	Counter 0 —40h, Counter 1 —41h, Counter 2—42h	Attribute:	R/W
Default Value:	All bits undefined	Size:	8 bit

Bit	Description
7:0	<b>Counter Port</b> —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

### 9.4.1 Interrupt Controller I/O MAP

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 on the data written to them. [Table 9-3](#) lists the different register possibilities for each address.

**Table 9-3. PIC Registers**

Port	Aliases	Register Name/Function	Default Value	Type
20h	24h, 28h, 2Ch, 30h, 34h, 38h, 3Ch	Master PIC ICW1 Init. Cmd Word 1 Register	Undefined	WO
		Master PIC OCW2 Op Ctrl Word 2 Register	001XXXXXb	WO
		Master PIC OCW3 Op Ctrl Word 3 Register	X01XXX10b	R/W
21h	25h, 29h, 2Dh, 31h, 35h, 39h, 3Dh	Master PIC ICW2 Init. Cmd Word 2 Register	Undefined	WO
		Master PIC ICW3 Init. Cmd Word 3 Register	Undefined	WO
		Master PIC ICW4 Init. Cmd Word 4 Register	01h	WO
		Master PIC OCW1 Op Ctrl Word 1 Register	00h	R/W
A0h	A4h, A8h, ACh, B0h, B4h, B8h, BCh	Slave PIC ICW1 Init. Cmd Word 1 Register	Undefined	WO
		Slave PIC OCW2 Op Ctrl Word 2 Register	001XXXXXb	WO
		Slave PIC OCW3 Op Ctrl Word 3 Register	X01XXX10b	R/W
A1h	A5h, A9h, ADh, B1h, B5h, B9h, BDh	Slave PIC ICW2 Init. Cmd Word 2 Register	Undefined	WO
		Slave PIC ICW3 Init. Cmd Word 3 Register	Undefined	WO
		Slave PIC ICW4 Init. Cmd Word 4 Register	01h	WO
		Slave PIC OCW1 Op Ctrl Word 1 Register	00h	R/W
4D0h	–	Master PIC Edge/Level Triggered Register	00h	R/W
4D1h	–	Slave PIC Edge/Level Triggered Register	00h	R/W

## 9.4.2 ICW1—Initialization Command Word 1 Register

Offset Address: Master Controller—020h      Attribute: WO  
 Slave Controller—0A0h      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	<b>ICW/OCW select</b> —WO. These bits are MCS-85 specific, and not needed. 000 = Should be programmed to “000”
4	<b>ICW/OCW select</b> —WO. 1 = This bit must be a 1 to select ICW1 and enable the ICW2, ICW3, and ICW4 sequence.
3	<b>Edge/Level Bank Select (LTIM)</b> —WO. Disabled. Replaced by the edge/level triggered control registers (ELCR).
2	<b>ADI</b> —WO. 0 = Ignored for the ICH2. Should be programmed to 0.
1	<b>Single or Cascade (SNGL)</b> —WO. 0 = Must be programmed to a 0 to indicate two controllers operating in cascade mode.
0	<b>ICW4 Write Required (IC4)</b> —WO. 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

Offset Address: Master Controller—021h      Attribute: WO  
Slave Controller—0A1h      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	<b>Interrupt Vector Base Address</b> —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	<b>Interrupt Request Level</b> —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: <table><tr><th>Code</th><th>Master Interrupt</th><th>Slave Interrupt</th></tr><tr><td>000</td><td>IRQ0</td><td>IRQ8</td></tr><tr><td>001</td><td>IRQ1</td><td>IRQ9</td></tr><tr><td>010</td><td>IRQ2</td><td>IRQ10</td></tr><tr><td>011</td><td>IRQ3</td><td>IRQ11</td></tr><tr><td>100</td><td>IRQ4</td><td>IRQ12</td></tr><tr><td>101</td><td>IRQ5</td><td>IRQ13</td></tr><tr><td>110</td><td>IRQ6</td><td>IRQ14</td></tr><tr><td>111</td><td>IRQ7</td><td>IRQ15</td></tr></table>	Code	Master Interrupt	Slave Interrupt	000	IRQ0	IRQ8	001	IRQ1	IRQ9	010	IRQ2	IRQ10	011	IRQ3	IRQ11	100	IRQ4	IRQ12	101	IRQ5	IRQ13	110	IRQ6	IRQ14	111	IRQ7	IRQ15
Code	Master Interrupt	Slave Interrupt																										
000	IRQ0	IRQ8																										
001	IRQ1	IRQ9																										
010	IRQ2	IRQ10																										
011	IRQ3	IRQ11																										
100	IRQ4	IRQ12																										
101	IRQ5	IRQ13																										
110	IRQ6	IRQ14																										
111	IRQ7	IRQ15																										

### 9.4.4 ICW3—Master Controller Initialization Command Word 3 Register

Offset Address: 21h      Attribute: WO  
Default Value: All bits undefined      Size: 8 bits

Bit	Description
7:3	0 = These bits must be programmed to zero.
2	<p><b>Cascaded Interrupt Controller IRQ Connection</b>—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.</p> <p>1 = This bit must always be programmed to a 1.</p>
1:0	0 = These bits must be programmed to zero.

### 9.4.5 ICW3—Slave Controller Initialization Command Word 3 Register

Offset Address: A1h Attribute: WO  
 Default Value: All bits undefined Size: 8 bits

Bit	Description
7:3	0 = These bits must be programmed to zero.
2:0	<b>Slave Identification Code</b> —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

Offset Address: Master Controller—021h Attribute: WO  
 Slave Controller—0A1h Size: 8 bits

Bit	Description
7:5	0 = These bits must be programmed to zero.
4	<b>Special Fully Nested Mode (SFNM)</b> —WO. 0 = Should normally be disabled by writing a 0 to this bit. 1 = Special fully nested mode is programmed.
3	<b>Buffered Mode (BUF)</b> —WO. 0 = Must be programmed to 0 for the ICH2. This is non-buffered mode.
2	<b>Master/Slave in Buffered Mode</b> —WO. Not used. 0 = Should always be programmed to 0.
1	<b>Automatic End of Interrupt (AEOI)</b> —WO. 0 = This bit should normally be programmed to 0. This is the normal end of interrupt. 1 = Automatic End of Interrupt (AEOI) mode is programmed. AEOI is discussed in <a href="#">Section 5.7.4</a> .
0	<b>Microprocessor Mode</b> —WO. 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

Offset Address: Master Controller—021h Attribute: R/W  
 Slave Controller—0A1h Size: 8 bits  
 Default Value: 00h

Bit	Description
7:0	<b>Interrupt Request Mask</b> —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

Offset Address: Master Controller—020h      Attribute: WO  
Slave Controller—0A0h      Size: 8 bits  
Default Value: Bit[4:0]=undefined, Bit[7:5]=001

Following a part reset or ICW initialization, the controller enters the fully nested mode of operation. Non-specific EOI without rotation is the default. Both rotation mode and specific EOI mode are disabled following initialization.

Bit	Description																				
7:5	<b>Rotate and EOI Codes (R, SL, EOI)</b> —WO. These three bits control the Rotate and End of Interrupt modes and combinations of the two. 000 = Rotate in Auto EOI Mode (Clear) 001 = Non-specific EOI command 010 = No Operation 011 = Specific EOI Command 100 = Rotate in Auto EOI Mode (Set) 101 = Rotate on Non-Specific EOI Command 110 = *Set Priority Command 111 = *Rotate on Specific EOI Command *L0–L2 Are Used																				
4:3	<b>OCW2 Select</b> —WO. When selecting OCW2, bits 4:3 = “00”																				
2:0	<b>Interrupt Level Select (L2, L1, L0)</b> —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. <table><tr><th>Bits</th><th>Interrupt Level</th><th>Bits</th><th>Interrupt Level</th></tr><tr><td>000</td><td>IRQ0/8</td><td>100</td><td>IRQ4/12</td></tr><tr><td>001</td><td>IRQ1/9</td><td>101</td><td>IRQ5/13</td></tr><tr><td>010</td><td>IRQ2/10</td><td>110</td><td>IRQ6/14</td></tr><tr><td>011</td><td>IRQ3/11</td><td>111</td><td>IRQ7/15</td></tr></table>	Bits	Interrupt Level	Bits	Interrupt Level	000	IRQ0/8	100	IRQ4/12	001	IRQ1/9	101	IRQ5/13	010	IRQ2/10	110	IRQ6/14	011	IRQ3/11	111	IRQ7/15
Bits	Interrupt Level	Bits	Interrupt Level																		
000	IRQ0/8	100	IRQ4/12																		
001	IRQ1/9	101	IRQ5/13																		
010	IRQ2/10	110	IRQ6/14																		
011	IRQ3/11	111	IRQ7/15																		

### 9.4.9 OCW3—Operational Control Word 3 Register

Offset Address: Master Controller—020h      Attribute: WO  
 Slave Controller—0A0h      Size: 8 bits  
 Default Value: Bit[6,0]=0, Bit[7,4:2]=undefined,  
 Bit[5,1]=1

Bit	Description
7	Reserved. Must be 0.
6	<b>Special Mask Mode (SMM)</b> —WO. 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	<b>Enable Special Mask Mode (ESMM)</b> —WO. 0 = Disable. The SMM bit becomes a "don't care". 1 = Enable the SMM bit to set or reset the Special Mask Mode.
4:3	<b>OCW3 Select</b> —WO. When selecting OCW3, bits 4:3 = "01"
2	<b>Poll Mode Command</b> —WO. 0 = Disable. Poll Command is not issued. 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	<b>Register Read Command</b> —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. 00 = No Action 01 = No Action 10 = Read IRQ Register 11 = Read IS Register



### 9.4.10 ELCR1—Master Controller Edge/Level Triggered Register

Offset Address: 4D0h  
Default Value: 00h

Attribute: R/W  
Size: 8 bits

In edge mode, (bit[x] = 0), the interrupt is recognized by a low to high transition. In level mode (bit[x] = 1), the interrupt is recognized by a high level. The cascade channel, IRQ2, the heart beat timer (IRQ0), and the keyboard controller (IRQ1), cannot be put into level mode.

Bit	Description
7	<b>IRQ7 ECL</b> —R/W. 0 = Edge. 1 = Level.
6	<b>IRQ6 ECL</b> —R/W. 0 = Edge. 1 = Level.
5	<b>IRQ5 ECL</b> —R/W. 0 = Edge. 1 = Level.
4	<b>IRQ4 ECL</b> —R/W. 0 = Edge. 1 = Level.
3	<b>IRQ3 ECL</b> —R/W. 0 = Edge. 1 = Level.
2:0	Reserved. Must be 0.

### 9.4.11 ELCR2—Slave Controller Edge/Level Triggered Register

Offset Address: 4D1h  
Default Value: 00h

Attribute: R/W  
Size: 8 bits

In edge mode (bit[x] = 0) the interrupt is recognized by a low-to-high transition. In level mode (bit[x] = 1) the interrupt is recognized by a high level. The real time clock interrupt (IRQ8#) and the floating point error interrupt (IRQ13) cannot be programmed for level mode.

Bit	Description
7	<b>IRQ15 ECL</b> —R/W. 0 = Edge. 1 = Level.
6	<b>IRQ14 ECL</b> —R/W. 0 = Edge. 1 = Level.
5	Reserved. Must be 0.
4	<b>IRQ12 ECL</b> —R/W. 0 = Edge. 1 = Level.
3	<b>IRQ11 ECL</b> —R/W. 0 = Edge. 1 = Level.
2	<b>IRQ10 ECL</b> —R/W. 0 = Edge. 1 = Level.
1	<b>IRQ9 ECL</b> —R/W. 0 = Edge. 1 = Level.
0	Reserved. Must be 0.

## 9.5 Advanced Interrupt Controller (APIC)

### 9.5.1 APIC Register Map

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 9-4](#).

**Table 9-4. APIC Direct Registers**

Address	Register	Size	Type
FEC0_0000h	Index Register	8 bits	R/W
FEC0_0010h	Data Register	32 bits	R/W
FEC0_0020h	IRQ Pin Assertion Register	8 bits	WO
FEC0_0040h	EOI Register	8 bits	WO

[Table 9-5](#) 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 9-5. APIC Indirect Registers**

Index	Register	Size	Type
00h	ID	32 bits	R/W
01h	Version	32 bits	RO
02h	Arbitration ID	32 bits	RO
03h	Boot Configuration	32 bits	R/W
03h–0Fh	Reserved		RO
10h–11h	Redirection Table 0	64 bits	R/W
12h–13h	Redirection Table 1	64 bits	R/W
...	...	...	...
3Eh–3Fh	Redirection Table 23	64 bits	R/W
40h–FFh	Reserved		RO

### 9.5.2 IND—Index Register

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 9-5](#). Software programs this register to select the desired APIC internal register

Bit	Description
7:0	<b>APIC Index</b> —R/W. This is an 8 bit pointer into the I/O APIC register table.

### 9.5.3 DAT—Data Register

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	<b>APIC Data</b> —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

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 5.8.4](#) 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 ICH2's PCI bus. Writes by devices on PCI buses above the ICH2 (e.g., a PCI segment on a P64H) are not supported.

Bit	Description
31:5	Reserved. Bits 31:5 are ignored.
4:0	<b>IRQ Number</b> —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

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:** This is similar to what already occurs when the APIC sees the EIO message on the serial bus. Note that 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 ICH2.

**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	<b>Redirection Entry Clear</b> —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

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 zero on power up reset.

Bit	Description
31:28	Reserved.
27:24	<b>APIC ID</b> —R/W. Software must program this value before using the APIC.
23:0	Reserved.

## 9.5.7 VER—Version Register

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	<b>Maximum Redirection Entries</b> —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 ICH2 this field is hardwired to 17h to indicate 24 interrupts.
15	<b>PRQ</b> —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	<b>Version</b> —RO. This is a version number that identifies the implementation version.

## 9.5.8 ARBID—Arbitration ID Register

Index Offset: 02h      Attribute: RO  
 Default Value: 00000000h      Size: 32 bits

This register contains the bus arbitration priority for the APIC. This register is loaded whenever the APIC ID register is loaded. A rotating priority scheme is used for APIC bus arbitration. The winner of the arbitration becomes the lowest priority agent and assumes an arbitration ID of 0.

Bit	Description
31:28	Reserved.
27:24	<b>I/O APIC Identification</b> —RO. This 4 bit field contains the I/O APIC Arbitration ID.
23:0	Reserved.

## 9.5.9 BOOT\_CONFIG—Boot Configuration Register

Index Offset: 03h      Attribute: R/W  
 Default Value: 00000000h      Size: 32 bits

This register is used to control the interrupt delivery mechanism for the APIC.

Bit	Description
31:1	Reserved.
0	<b>Delivery Type (DT)</b> —R/W. 0 = Interrupt delivery mechanism is via the APIC serial bus (default). 1 = Interrupt delivery mechanism is a front-side bus message.

## 9.5.10 Redirection Table

Index Offset:	10h–11h (vector 0) through 3E–3Fh (vector 23)	Attribute:	R/W
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 bus unit that the interrupt message was sent over the APIC bus. 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	<b>Destination</b> —R/W. If bit 11 of this entry is 0 [Physical], then bits [59:56] specifies an APIC ID. If bit 11 of this entry is 1 [Logical], then bits [63:56] specify the logical destination address of a set of processors.
55:17	Reserved.
16	<b>Mask</b> —R/W. 0 = Not masked: An edge or level on this interrupt pin results in the delivery of the interrupt to the destination. 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	<b>Trigger Mode</b> —R/W. This field indicates the type of signal on the interrupt pin that triggers an interrupt. 0 = Edge triggered. 1 = Level triggered.
14	<b>Remote IRR</b> —R/W. This bit is used for level triggered interrupts; its meaning is undefined for edge triggered interrupts. 0 = Reset when an EOI message is received from a local APIC. 1 = Set when Local APIC/s accept the level interrupt sent by the I/O APIC.
13	<b>Interrupt Input Pin Polarity</b> —R/W. This bit specifies the polarity of each interrupt signal connected to the interrupt pins. 0 = Active high. 1 = Active low.
12	<b>Delivery Status</b> —RO. This field contains the current status of the delivery of this interrupt. Writes to this bit have no effect. 0 = Idle. No activity for this interrupt. 1 = Pending. Interrupt has been injected, but delivery is held up due to the APIC bus being busy or the inability of the receiving APIC unit to accept the interrupt at this time.
11	<b>Destination Mode</b> —R/W. This field determines the interpretation of the Destination field. 0 = Physical. Destination APIC ID is identified by bits [59:56]. 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.

Bit	Description
10:8	<p><b>Delivery Mode</b>—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:</p> <p>000 = Fixed. Deliver the signal on the INTR signal of all processor cores listed in the destination. Trigger Mode can be edge or level.</p> <p>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.</p> <p>010 = SMI (System Management Interrupt). Requires the interrupt to be programmed as edge triggered. The vector information is ignored but must be programmed to all zeroes for future compatibility.</p> <p>011 = Reserved</p> <p>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. Once the interrupt is detected, it will be sent over the APIC bus.</p> <p>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 over the APIC bus again.</p> <p>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. Once the interrupt is detected, it will be sent over the APIC bus.</p> <p>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 over the APIC bus again</p> <p>110 = Reserved</p> <p>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.</p>
7:0	<p><b>Vector</b>—R/W. This field contains the interrupt vector for this interrupt. Values range between 10h and FEh.</p>

## 9.6 Real Time Clock Registers

### 9.6.1 I/O Register Address Map

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 9-6](#).

**Table 9-6. RTC I/O Registers**

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 9-7](#). Locations 72h and 73h are for accessing the extended RAM. The extended RAM bank is also accessed using an indexed scheme. I/O address 72h is used as the address pointer and I/O address 73h is used as the data register. Index addresses above 127h are not valid. If the extended RAM is not needed, it may be disabled.
2. Software must preserve the value of bit 7 at I/O addresses 70h and 74h. When writing to these addresses, software must first read the value, and then write the same value for bit 7 during the sequential address write.

### 9.6.2 Indexed Registers

The RTC contains two sets of indexed registers that are accessed using the two separate Index and Target registers (70h/71h or 72h/73h), as shown in [Table 9-7](#).

**Table 9-7. RTC (Standard) RAM Bank**

Index	Name	Index	Name
00h	Seconds	08h	Month
01h	Seconds Alarm.	09h	Year
02h	Minutes	0Ah	Register A
03h	Minutes Alarm	0Bh	Register B
04h	Hours	0Ch	Register C
05h	Hours Alarm	0Dh	Register D
06h	Day of Week	0Eh–7Fh	114 Bytes of User RAM
07h	Day of Month		

### 9.6.2.1 RTC\_REGA—Register A

RTC Index:	0A	Attribute:	R/W
Default Value:	Undefined	Size:	8-bit
Lockable:	No	Power Well:	RTC

This register is used for general configuration of the RTC functions. None of the bits are affected by RSMRST# or any other ICH2 reset signal.

Bit	Description																
7	<b>Update In Progress (UIP)</b> —R/W. This bit may be monitored as a status flag. 0 = The update cycle will not start for at least 492us. The time, calendar, and alarm information in RAM is always available when the UIP bit is 0. 1 = The update is soon to occur or is in progress.																
6:4	<b>Division Chain Select (DV[2:0])</b> —R/W. These three bits control the divider chain for the oscillator, and are not affected by RSMRST# or any other reset signal. DV[2] corresponds to bit 6. 010 = Normal Operation 11X = Divider Reset 101 = Bypass 15 stages (test mode only) 100 = Bypass 10 stages (test mode only) 011 = Bypass 5 stages (test mode only) 001 = Invalid 000 = Invalid																
3:0	<b>RS[3:0] Rate Select</b> —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 zero. RS3 corresponds to bit 3. <table> <tr> <td>0000 = Interrupt never toggles</td><td>1000 = 3.90625 ms</td></tr> <tr> <td>0001 = 3.90625 ms</td><td>1001 = 7.8125 ms</td></tr> <tr> <td>0010 = 7.8125 ms</td><td>1010 = 15.625 ms</td></tr> <tr> <td>0011 = 122.070 us</td><td>1011 = 31.25 ms</td></tr> <tr> <td>0100 = 244.141 us</td><td>1100 = 62.5 ms</td></tr> <tr> <td>0101 = 488.281 us</td><td>1101 = 125 ms</td></tr> <tr> <td>0110 = 976.5625 us</td><td>1110 = 250 ms</td></tr> <tr> <td>0111 = 1.953125 ms</td><td>1111 = 500 ms</td></tr> </table>	0000 = Interrupt never toggles	1000 = 3.90625 ms	0001 = 3.90625 ms	1001 = 7.8125 ms	0010 = 7.8125 ms	1010 = 15.625 ms	0011 = 122.070 us	1011 = 31.25 ms	0100 = 244.141 us	1100 = 62.5 ms	0101 = 488.281 us	1101 = 125 ms	0110 = 976.5625 us	1110 = 250 ms	0111 = 1.953125 ms	1111 = 500 ms
0000 = Interrupt never toggles	1000 = 3.90625 ms																
0001 = 3.90625 ms	1001 = 7.8125 ms																
0010 = 7.8125 ms	1010 = 15.625 ms																
0011 = 122.070 us	1011 = 31.25 ms																
0100 = 244.141 us	1100 = 62.5 ms																
0101 = 488.281 us	1101 = 125 ms																
0110 = 976.5625 us	1110 = 250 ms																
0111 = 1.953125 ms	1111 = 500 ms																

### 9.6.2.2 RTC\_REGB—Register B (General Configuration)

RTC Index: 0Bh Attribute: R/W  
Default Value: U0U00UUU (U: Undefined) Size: 8-bit  
Lockable: No Power Well: RTC

Bit	Description
7	<b>Update Cycle Inhibit (SET)</b> —R/W. Enables/Inhibits the update cycles. This bit is not affected by RSMRST# nor any other reset signal. 0 = Update cycle occurs normally once each second. 1 = A current update cycle will abort and subsequent update cycles will not occur until SET is returned to zero. When set is one, the BIOS may initialize time and calendar bytes safely.
6	<b>Periodic Interrupt Enable (PIE)</b> —R/W. This bit is cleared by RSMRST#, but not on any other reset. 0 = Disable. 1 = Allows an interrupt to occur with a time base set with the RS bits of register A.
5	<b>Alarm Interrupt Enable (AIE)</b> —R/W. This bit is cleared by RSMRST#, but not on any other reset. 0 = Disable. 1 = 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	<b>Update-ended Interrupt Enable (UIE)</b> —R/W. This bit is cleared by RSMRST#, but not on any other reset. 0 = Disable. 1 = Allows an interrupt to occur when the update cycle ends.
3	<b>Square Wave Enable (SQWE)</b> —R/W. This bit serves no function in the ICH2. It is left in this register bank to provide compatibility with the Motorola* 146818B. The ICH2 has no SQW pin. This bit is cleared by RSMRST#, but not on any other reset.
2	<b>Data Mode (DM)</b> —R/W. Specifies either binary or BCD data representation. This bit is not affected by RSMRST# nor any other reset signal. 0 = BCD 1 = Binary
1	<b>Hour Format (HOURFORM)</b> —R/W. Indicates the hour byte format. This bit is not affected by RSMRST# nor any other reset signal. 0 = Twelve-hour mode. In twelve hour mode, the seventh bit represents AM as zero and PM as one. 1 = Twenty-four hour mode.
0	<b>Daylight Savings Enable (DSE)</b> —R/W. 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. 0 = Daylight Savings Time updates do not occur. 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.

### 9.6.2.3 RTC\_REGC—Register C (Flag Register)

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	<b>Interrupt Request Flag (IRQF)</b> —RO. $IRQF = (PF * PIE) + (AF * AIE) + (UF * UFE)$ . This also causes the CH_IRQ_B signal to be asserted. This bit is cleared upon RSMRST# or a read of Register C.
6	<b>Periodic Interrupt Flag (PF)</b> —RO. This bit is cleared upon RSMRST# or a read of Register C. 0 = If no taps are specified via the RS bits in Register A, this flag will not be set. 1 = Periodic interrupt Flag will be 1 when the tap specified by the RS bits of register A is 1.
5	<b>Alarm Flag (AF)</b> —RO. 0 = This bit is cleared upon RTCRST# or a read of Register C. 1 = Alarm Flag will be set after all Alarm values match the current time.
4	<b>Update-ended Flag (UF)</b> —RO. 0 = The bit is cleared upon RSMRST# or a read of Register C. 1 = Set immediately following an update cycle for each second.
3:0	Reserved. Will always report 0.

### 9.6.2.4 RTC\_REGD—Register D (Flag Register)

RTC Index:	0Dh	Attribute:	R/W
Default Value:	10UUUUUU (U: Undefined)	Size:	8-bit
Lockable:	No	Power Well:	RTC

Bit	Description
7	<b>Valid RAM and Time Bit (VRT)</b> —R/W. 0 = This bit should always be written as a 0 for write cycle; however, it will return a 1 for read cycles. 1 = The Valid Ram and Time bit is set to 1 when the PWRGD (power good) signal provided is high. This feature is not typically used.
6	Reserved. This bit always returns a 0 and should be set to 0 for write cycles.
5:0	<b>Date Alarm</b> —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 zeros to mimic the functionality of the Motorola* 146818B. These bits are not affected by RESET.

## 9.7 Processor Interface Registers

### 9.7.1 NMI\_SC—NMI Status and Control Register

I/O Address:	61h	Attribute:	R/W (some bits RO)
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7	<b>SERR# NMI Source Status (SERR#_NMI_STS)</b> —RO. 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	<b>IOCHK# NMI Source Status (IOCHK#_NMI_STS)</b> —RO. 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	<b>Timer Counter 2 OUT Status (TMR2_OUT_STS)</b> —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	<b>Refresh Cycle Toggle (REF_TOGGLE)</b> —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	<b>IOCHK# NMI Enable (IOCHK#_NMI_EN)</b> —R/W. 0 = Enabled. 1 = Disabled and cleared.
2	<b>PCI SERR# Enable (PCI_SERR_EN)</b> —R/W. 0 = SERR# NMIs are enabled. 1 = SERR# NMIs are disabled and cleared.
1	<b>Speaker Data Enable (SPKR_DAT_EN)</b> —R/W. 0 = SPKR output is a 0. 1 = SPKR output is equivalent to the Counter 2 OUT signal value.
0	<b>Timer Counter 2 Enable (TIM_CNT2_EN)</b> —R/W. 0 = Disable. 1 = Enable

### 9.7.2 NMI\_EN—NMI Enable (and Real Time Clock Index)

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 Table 19-2), and all bits are readable at that address.

Bits	Description
7	<b>NMI Enable (NMI_EN)</b> —R/W. 0 = Enable NMI sources. 1 = Disable All NMI sources.
6:0	<b>Real Time Clock Index Address (RTC_INDX)</b> —R/W. 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

I/O Address:	92h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:2	Reserved.
1	<b>Alternate A20 Gate (ALT_A20_GATE)</b> —R/W. This bit is ORed with the A20GATE input signal to generate A20M# to the processor. 0 = A20M# signal can potentially go active. 1 = This bit is set when INIT# goes active.
0	<b>Interrupt Now (INIT_NOW)</b> —R/W. When this bit transitions from a 0 to a 1, the ICH2 will force INIT# active for 16 PCI clocks.

### 9.7.4 COPROC\_ERR—Coprocessor Error Register

I/O Address:	F0h	Attribute:	WO
Default Value:	00h	Size:	8-bits
Lockable:	No	Power Well:	Core

Bits	Description
7:0	<b>Coprocessor Error (COPROC_ERR)</b> —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

I/O Address:	CF9h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:4	Reserved.
3	<b>Full Reset (FULL_RST)</b> —R/W. This bit is used to determine the states of SLP_S3# 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 time-outs. 1 = ICH2 will drive SLP_S3# and SLP_S5# low for 3–5 seconds. 0 = ICH2 will keep SLP_S3# and SLP_S5# high.
2	<b>Reset Processor (RST_CPU)</b> —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	<b>System Reset (SYS_RST)</b> —R/W. This bit is used to determine a hard or soft reset to the processor. 1 = When RST_CPU bit goes from 0 to 1, the ICH2 performs a hard reset by activating PCIRST# for 1 millisecond. 0 = When RST_CPU bit goes from 0 to 1, the ICH2 performs a soft reset by activating INIT# for 16 PCI clocks.
0	Reserved.

## 9.8 Power Management Registers (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.

### 9.8.1 Power Management PCI Configuration Registers (D31:F0)

Table 9-8 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 9-8. PCI Configuration Map (PM—D31:F0)**

Offset	Mnemonic	Register Name/Function	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
A2h	GEN_PMCON_2	General Power Management Configuration 2	0000h	R/W
A4h	GEN_PMCON_3	General Power Management Configuration 3	00h	R/W
B8–BBh	GPI_ROUT	GPI Route Control	00000000h	R/W
C0	TRP_FWD_EN	I/O Monitor Trap Forwarding Enable		
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

### 9.8.1.1 GEN\_PMCON\_1—General PM Configuration 1 Register (PM—D31:F0)

Offset Address:	A0h	Attribute:	R/W
Default Value:	00h	Size:	16-bit
Lockable:	No	Usage:	ACPI, Legacy
		Power Well:	Core

Bit	Description
15:11	Reserved
10	<b>Software SMI Rate Select (SWSMI_RATE_SEL)</b> —R/W. 0 = SWSMI Timer will time out in 64 ms $\pm$ 4 ms (default). 1 = SWSMI Timer will time out in 1.5 ms $\pm$ 0.5 ms.
9	<b>PWRBTN# Level (PWRBTN_LVL)</b> —RO. This read-only bit indicates the current state of the PWRBTN# signal. 0 = Low. 1 = High.
8:7	Reserved.
6	<b>iiA64 Processor Mode Enable (A64_EN)</b> —R/W. Set by software to indicate the presence of an iA64 processor. 0 = iA32 processor mode. 1 = iA64 processor mode.
5	<b>CPU SLP# Enable (CPUSLP_EN)</b> —R/W. 0 = Disable. 1 = Enables the CPUSLP# signal to go active in the S1 state. This reduces the processor power. Note that CPUSLP# will go active on entry to S3, S4 and S5 even if this bit is not set.
4:2	Reserved.
1:0	<b>Periodic SMI# rate Select (PER_SMI_SEL)</b> —R/W. Set by software to control the rate at which periodic SMI# is generated. 00 = 1 minute 01 = 32 seconds 10 = 16 seconds 11 = 8 seconds

### 9.8.1.2 GEN\_PMCON\_2—General PM Configuration 2 Register (PM—D31:F0)

Offset Address:	A2h	Attribute:	R/WC
Default Value:	00h	Size:	16-bit
Lockable:	No	Usage:	ACPI, Legacy
		Power Well:	Resume

Bit	Description
7:2	Reserved.
1	<b>CPU Power Failure (CPUPWR_FLR)—R/WC.</b> 0 = Software clears this bit by writing a 1 to the bit position. 1 = Indicates that the VRMPWRGD signal from the processor's VRM went low.
0	<b>PWROK Failure (PWROK_FLR)—R/WC.</b> 0 = Software clears this bit by writing a 1 to the bit position, or when the system goes into a G3 state. 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. Note: Traditional designs have a reset button logically ANDed with the PWROK signal from the power supply and the processor's voltage regulator module. If this is done with the ICH2, the PWROK_FLR bit will be set. The ICH2 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 ICH2 will reboot (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.

### 9.8.1.3 GEN\_PMCON\_3—General PM Configuration 3 Register (PM—D31:F0)

Offset Address:	A4h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Usage:	ACPI, Legacy
		Power Well:	RTC

Bit	Description
7:3	Reserved.
2	<b>RTC Power Status (RTC_PWR_STS)—R/WC.</b> 0 = Software clears this bit by writing a 0 to the bit position. 1 = Indicates that the RTC battery was removed or failed. This bit is set when RTCRST# signal is low. 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.
1	<b>Power Failure (PWR_FLR)—R/WC.</b> This bit is in the RTC well, and is not cleared by any type of reset except RTCRST#. 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 the bit position. 1 = Indicates that the trickle current (from the main battery or trickle supply) was removed or failed. 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	<b>After G3 State Select (AFTERG3_EN)—R/W.</b> 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#. 0 = System will return to S0 state (boot) after power is re-applied. 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.

### 9.8.1.4 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	<b>GPI[15] Route</b> —R/W. See bits 1:0 for description.
<b>Same pattern for GPI[14] through GPI[3]</b>	
5:4	<b>GPI[2] Route</b> —R/W. See bits 1:0 for description.
3:2	<b>GPI[1] Route</b> —R/W. See bits 1:0 for description.
1:0	<b>GPI[0] Route</b> —R/W. GPIO[13:11,8:6,4:3,1:0] 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. If the system is in an S1–S5 state and if the GPE1_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. 00 = No effect. 01 = SMI# (if corresponding GPE1_EN bit is also set) 10 = SCI (if corresponding GPE1_EN bit is also set) 11 = Reserved

**Note:** GPIOs that are not implemented will not have the corresponding bits implemented in this register.

### 9.8.1.5 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 ICH2 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	<b>Monitor 7 Forward Enable (MON7_FWD_EN)</b> —R/W. 0 = Disable. Cycles trapped by I/O Monitor 7 will not be forwarded to LPC. 1 = Enable. Cycles trapped by I/O Monitor 7 will be forwarded to LPC.
6	<b>Monitor 6 Forward Enable (MON6_FWD_EN)</b> —R/W. 0 = Disable. Cycles trapped by I/O Monitor 6 will not be forwarded to LPC. 1 = Enable. Cycles trapped by I/O Monitor 6 will be forwarded to LPC.
5	<b>Monitor 5 Forward Enable (MON5_FWD_EN)</b> —R/W. 0 = Disable. Cycles trapped by I/O Monitor 5 will not be forwarded to LPC. 1 = Enable. Cycles trapped by I/O Monitor 5 will be forwarded to LPC.
4	<b>Monitor 4 Forward Enable (MON4_FWD_EN)</b> —R/W. 0 = Disable. Cycles trapped by I/O Monitor 4 will not be forwarded to LPC. 1 = Enable. Cycles trapped by I/O Monitor 4 will be forwarded to LPC.
3:0	Reserved.

### 9.8.1.6 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 ICH2 generates an SMI#. This SMI# occurs if the address is positively decoded by another device on PCI or by the ICH2 (because it would be forwarded to LPC or some other ICH2 internal registers). The trap ranges should not point to registers in the ICH2's internal IDE, USB, AC'97 or LAN I/O space. If the cycle is to be claimed by the ICH2 and targets one of the permitted ICH2 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 ICH2 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	<b>Monitor Trap Base Address (MON[n]_TRAP_BASE)</b> —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). 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).

### 9.8.1.7 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	<b>Monitor 7 Forward Mask (MON7_MASK)</b> —R/W. Selects low 4-bit mask for the I/O locations that MON7 will trap. Similar to MON4_MASK.
11:8	<b>Monitor 6 Forward Mask (MON6_MASK)</b> —R/W. Selects low 4-bit mask for the I/O locations that MON6 will trap. Similar to MON4_MASK.
7:4	<b>Monitor 5 Forward Mask (MON5_MASK)</b> —R/W. Selects low 4-bit mask for the I/O locations that MON5 will trap. Similar to MON4_MASK.
3:0	<b>Monitor 4 Forward Mask (MON4_MASK)</b> —R/W. 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. For example, if MON4_TRAP_BASE = 1230h, and MON4_MSK = 0011b, the ICH2 will decode 1230h, 1231h, 1232h, and 1233h for Monitor 4.

## 9.8.2 APM I/O Decode

Table 9-9 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 9-9. APM Register Map**

Address	Mnemonic	Register Name/Function	Default	Type
B2h	APM_CNT	Advanced Power Management Control Port	00h	R/W
B3h	APM_STS	Advanced Power Management Status Port	00h	R/W

### 9.8.2.1 APM\_CNT—Advanced Power Management Control Port Register

I/O Address:	B2h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Usage:	Legacy Only
Power Well:	Core		

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.

### 9.8.2.2 APM\_STS—Advanced Power Management Status Port Register

I/O Address:	B3h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Usage:	Legacy Only
Power Well:	Core		

Bit	Description
7:0	Used to pass data between the OS and the SMI handler. Basically, this is a scratchpad register and is not effected by any other register or function (other than a PCI reset).

### 9.8.3 Power Management I/O Registers

Table 9-10 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 ACPI 1.0 specification, and use the same bit names.

**Note:** All reserved bits and registers will always return 0 when read, and will have no effect when written.

**Table 9-10. ACPI and Legacy I/O Register Map**

PMBASE+ Offset	Register Name	ACPI Pointer	Default	Attributes
00–01h	PM1 Status	PM1a_EVT_BLK	0000h	R/W
02–03h	PM1 Enable	PM1a_EVT_BLK+2	0000h	R/W
04–07h	PM1 Control	PM1a_CNT_BLK	00000000h	R/W
08–0Bh	PM1 Timer	PMTMR_BLK	00000000h	RO
0Ch	Reserved	—	—	—
10h–13h	Processor Control	P_BLK	00000000h	R/W
14h	Level 2 Register	P_BLK+4	00h	RO
15–20h	Reserved	—	—	—
28–29h	General Purpose Event 0 Status	GPE0_BLK	0000h	R/W
2A–2Bh	General Purpose Event 0 Enables	GPE0_BLK+2	0000h	R/W
2C–2D	General Purpose Event 1 Status	GPE1_BLK	0000h	R/W
2E–2F	General Purpose Event 1 Enables	GPE1_BLK+2	0000h	R/W
30–31h	SMI# Control and Enable	—	0000h	R/W
34–35h	SMI Status Register	—	0000h	R/W
36–3Fh	Reserved	—	0000h	RO
40h	Monitor SMI Status	—	0000h	R/W
42h	Reserved	—	—	—
44h	Device Trap Status	—	0000h	R/W
48h	Trap Enable register	—	0000h	R/W
4Ch–4Dh	Bus Address Tracker	—	Last Cycle	RO
4Eh	Bus Cycle Tracker	—	Last Cycle	RO
50–5Fh	Reserved	—	—	—
60h–7Fh	Reserved for TCO Registers	—	—	—

### 9.8.3.1 PM1\_STS—Power Management 1 Status Register

I/O Address:	PMBASE + 00h (ACPI PM1a_EVT_BLK)	Attribute:	R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI or Legacy
Power Well:	Bits 0–7: Core, Bits 8–15: Resume, except Bit 11 in RTC		

If bit 10 or 8 in this register is 1 and the corresponding \_EN bit is set in the PM1\_EN register, ICH2 generates a Wake Event. Once back in an S0 state (or if already in S0 state when the event occurs), ICH2 also generates 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	<p><b>Wake Status (WAK_STS)</b>—R/WC. This bit is not affected by hard resets caused by a CF9 write but is reset by RSMRST#.</p> <p>0 = Software clears this bit by writing a 1 to the bit position. 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 ICH2 will transition the system to the ON state.</p> <p>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.</p> <p>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).</p>
14:12	Reserved
11	<p><b>Power Button Override Status (PRBTNOR_STS)</b>—R/WC. This bit is not affected by hard resets caused by a CF9 write and is not reset by RSMRST#. Thus, this bit will be preserved through a power failure.</p> <p>0 = The BIOS or SCI handler can clear this bit by writing a 1 to it. 1 = Set by hardware anytime a Power Button Override Event occurs which occurs when the power button is pressed for at least 4 consecutive seconds. The power button override causes an unconditional transition to the S5 state and sets the AFTERG3 bit. This bit can also be set by the SMBus Slave logic.</p>
10	<p><b>RTC Status (RTC_STS)</b>—R/WC. This bit is not affected by hard resets caused by a CF9 write but is reset by RSMRST#.</p> <p>0 = Software clears this bit by writing a 1 to the bit position. 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.</p>
9	Reserved
8	<p><b>Power Button Status (PWRBTN_STS)</b>—R/WC. This bit is not affected by hard resets caused by a CF9 write.</p> <p>1 = This bit is set by hardware when the PWRBTN# signal is asserted Low, independent of any other enable bit. 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. In any sleeping state S1–S5, while PWRBTN_EN and PWRBTN_STS are both set, a wake event is generated.</p> <p>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. This bit can be cleared by software by writing a one to the bit position.</p>

Bit	Description
7:6	Reserved
5	<b>Global Status (GBL_STS)</b> —R/W. 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. 0 = The SCI handler should then clear this bit by writing a 1 to the bit location.
4:1	Reserved
0	<b>Timer Overflow Status (TMROF_STS)</b> —R/W. 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). 0 = The SCI or SMI# handler clears this bit by writing a 1 to the bit location.

### 9.8.3.2 PM1\_EN—Power Management 1 Enable Register

I/O Address:	PMBASE + 02h (ACPI PM1a_EVT_BLK + 2)	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI or Legacy
Power Well:	Bits 0–7: Core, Bits 8–15: Resume		

Bit	Description												
15:11	Reserved.												
10	<b>RTC Event Enable (RTC_EN)</b> —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. 1 = An SCI (or SMI#) or wake event will occur when this bit is set and the RTC_STS bit goes active. 0 = No SCI (or SMI#) or wake event is generated then RTC_STS goes active.												
8	<b>Power Button Enable (PWRBTN_EN)</b> —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. 0 = Disable. 1 = Enable.												
5	<b>Global Enable (GBL_EN)</b> —R/W. When both the GBL_EN and the GBL_STS are set, an SCI is raised. 0 = Disable. 1 = Enable SCI on GBL_STS going active.												
0	<b>Timer Overflow Interrupt Enable (TMROF_EN)</b> —R/W. Works in conjunction with the SCI_EN bit as described below: <table><tr><th>TMROF_EN</th><th>SCI_EN</th><th>Effect when TMROF_STS is set</th></tr><tr><td>0</td><td>x</td><td>No SMI# or SCI</td></tr><tr><td>1</td><td>0</td><td>SMI#</td></tr><tr><td>1</td><td>1</td><td>SCI</td></tr></table>	TMROF_EN	SCI_EN	Effect when TMROF_STS is set	0	x	No SMI# or SCI	1	0	SMI#	1	1	SCI
TMROF_EN	SCI_EN	Effect when TMROF_STS is set											
0	x	No SMI# or SCI											
1	0	SMI#											
1	1	SCI											

### 9.8.3.3 PM1\_CNT—Power Management 1 Control Register

I/O Address:	PMBASE + 04h (ACPI PM1a_CNT_BLK)	Attribute:	R/W
Default Value:	0000h	Size:	32-bit
Lockable:	No	Usage:	ACPI or Legacy
Power Well:	Bits 0–7: Core, Bits 8–15: Resume		

Bit	Description
13	<b>Sleep Enable (SLP_EN)</b> —WO. Setting this bit causes the system to sequence into the Sleep state defined by the SLP_TYP field.
12:10	<b>Sleep Type (SLP_TYP)</b> —R/W. This 3-bit field defines the type of Sleep the system should enter when the SLP_EN bit is set to 1. 000 = ON: Typically maps to S0 state. 001 = Assert STPCLK#. Puts processor in Stop-Grant state. Optional to assert CPUSLP# to put processor in sleep state: Typically maps to S1 state. 010 = Reserved 011 = Reserved 100 = Reserved 101 = Suspend-To-RAM. Assert SLP_S1# and SLP_S3#; typically, maps to S3 state. 110 = Suspend-To-Disk. Assert SLP_S1#, SLP_S3#, and SLP_S5# SLP_S3# and, SLP_S5#; typically, maps to S4 state. 111 = Soft Off. Assert SLP_S1#, SLP_S3#, and SLP_S5# SLP_S3#, and SLP_S5#; typically, maps to S5 state.
2	<b>Global Release (GBL_RLS)</b> —WO. 1 = ACPI software writes a 1 to this bit to raise an event to the BIOS. BIOS software has corresponding enable and status bits to control its ability to receive ACPI events. 0 = This bit always reads as 0.
1	Reserved
0	<b>SCI Enable (SCI_EN)</b> —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. 0 = These events will generate an SMI#. 1 = These events will generate an SCI.

### 9.8.3.4 PM1\_TMR—Power Management 1 Timer Register

I/O Address:	PMBASE + 08h (ACPI PMTMR_BLK)	Attribute:	RO
Default Value:	xx000000h	Size:	32-bit
Lockable:	No	Usage:	ACPI
Power Well:	Core		

Bit	Description
31:24	Reserved
23:0	<b>Timer Value (TMR_VAL)</b> —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 zero during a PCI reset and then continues counting as long as the system is in the S0 state. 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.

### 9.8.3.5 PROC\_CNT—Processor Control Register

I/O Address:	PMBASE + 10h ( <b>ACPI P_BLK</b> )	Attribute:	R/W
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	<b>Throttle Status (THTL_STS)</b> —RO. 0 = No clock throttling is occurring (maximum processor performance). 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	<b>Force Thermal Throttling (FORCE_THTL)</b> —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. 0 = No forced throttling. 1 = Throttling at the duty cycle specified in THRM_DTY starts immediately (no 2 second delay), and no SMI# is generated.																											
7:5	<b>Thermal Duty Cycle (THRM_DTY)</b> . This write-once 3-bit 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. If in the C2 state, no throttling occurs.  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. <table><tr><th>THRM_DTY</th><th>Throttle Mode</th><th>PCI Clocks</th></tr><tr><td>000</td><td>RESERVED (Default) (Will be 50%)</td><td>512</td></tr><tr><td>001</td><td>87.5%</td><td>896</td></tr><tr><td>010</td><td>75.0%</td><td>768</td></tr><tr><td>011</td><td>62.5%</td><td>640</td></tr><tr><td>100</td><td>50%</td><td>512</td></tr><tr><td>101</td><td>37.5%</td><td>384</td></tr><tr><td>110</td><td>25%</td><td>256</td></tr><tr><td>111</td><td>12.5%</td><td>128</td></tr></table>	THRM_DTY	Throttle Mode	PCI Clocks	000	RESERVED (Default) (Will be 50%)	512	001	87.5%	896	010	75.0%	768	011	62.5%	640	100	50%	512	101	37.5%	384	110	25%	256	111	12.5%	128
THRM_DTY	Throttle Mode	PCI Clocks																										
000	RESERVED (Default) (Will be 50%)	512																										
001	87.5%	896																										
010	75.0%	768																										
011	62.5%	640																										
100	50%	512																										
101	37.5%	384																										
110	25%	256																										
111	12.5%	128																										
4	<b>Throttling Enable (THT_EN)</b> : When this bit is set and the system is in a C0 state, processor-controlled STPCLK# throttling is enabled. The duty cycle is selected in the THTL_DTY field. 0 = Disable 1 = Enable																											
3:1	<b>Throttling Duty Cycle (THTL_DTY)</b> : This 3-bit field determines the duty cycle of the throttling when the THTL_EN bit is set. The duty cycle indicates the approximate percentage of time the STPCLK# signal is asserted (low) while in the throttle mode. The STPCLK# throttle period is 1024 PCICLKs. <table><tr><th>THTL_DTY</th><th>Throttle Mode</th><th>PCI Clocks</th></tr><tr><td>000</td><td>RESERVED (Default) (Will be 50%)</td><td>512</td></tr><tr><td>001</td><td>87.5%</td><td>896</td></tr><tr><td>010</td><td>75.0%</td><td>768</td></tr><tr><td>011</td><td>62.5%</td><td>640</td></tr><tr><td>100</td><td>50%</td><td>512</td></tr><tr><td>101</td><td>37.5%</td><td>384</td></tr><tr><td>110</td><td>25%</td><td>256</td></tr><tr><td>111</td><td>12.5%</td><td>128</td></tr></table>	THTL_DTY	Throttle Mode	PCI Clocks	000	RESERVED (Default) (Will be 50%)	512	001	87.5%	896	010	75.0%	768	011	62.5%	640	100	50%	512	101	37.5%	384	110	25%	256	111	12.5%	128
THTL_DTY	Throttle Mode	PCI Clocks																										
000	RESERVED (Default) (Will be 50%)	512																										
001	87.5%	896																										
010	75.0%	768																										
011	62.5%	640																										
100	50%	512																										
101	37.5%	384																										
110	25%	256																										
111	12.5%	128																										
0	Reserved																											

### 9.8.3.6 LV2—Level 2 Register

I/O Address:	PMBASE + 14h ( <b>ACPI P_BLK+4</b> )	Attribute:	RO
Default Value:	00h	Size:	8-bit
Lockable:	No	Usage:	ACPI or Legacy
Power Well:	Core		

Bit	Description
7:0	Reads to this register return all zeros; writes have no effect. Reads to this register generate a “enter a level 2 power state” (C2) to the clock control logic. This causes the STPCLK# signal to go active, and stay active until a break event occurs. Throttling (due either to THTL_EN or THRM# override) will be ignored.

### 9.8.3.7 GPE0\_STS—General Purpose Event 0 Status Register

I/O Address:	PMBASE + 28h ( <b>ACPI GPE0_BLK</b> )	Attribute:	R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI
Power Well:	Resume		

**Note:** This register is symmetrical to the General Purpose Event 0 Enable Register. If the corresponding seen bit is set, then when the \_STS bit get set, ICH2 generates a Wake Event. Once back in an S0 state (or if already in an S0 state when the event occurs), ICH2 also generates 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 x\_EN bit. None of these bits are reset by CF9h write. All are reset by RSMRST#, except for RI\_STS.

Bit	Description
15:12	Reserved.
11	<b>PME Status (PME_STS)—R/WC.</b> 0 = Software clears this bit by writing a 1 to the bit position. 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.
8	<b>RI_STS—R/WC.</b> The value of this bit will be maintained through a G3 state and is not affected by a hard reset caused by RSMRST# or a CF9h write. Assertion of RTCRST# will reset this bit. 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware when the RI# input signal goes active.
7	<b>SMBus Wake Status (SMB_WAK_STS):</b> SMBus Wake Status—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). 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware to indicate that the wake event was caused by the ICH2's SMBus logic.
6	<b>TCO SCI Status (TCOSCI_STS)—R/WC.</b> 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware when the TCO logic causes an SCI.

Bit	Description
5	<b>AC97 Status (AC97_STS)—R/WC.</b> 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. 0 = Software clears this bit by writing a 1 to the bit position. 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: 1. ACSDIN[1] or ACSDIN[0] is high and BITCLK is not oscillating, or 2. The GSCI bit is set (section 13.2.9, NAMBAR +30h, bit 0)
4	<b>USB Controller 2 Status (USB2_STS)—R/WC.</b> 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware when USB Controller 2 needs to cause a wake. Wake event will be generated if the corresponding USB2_EN bit is set.
3	<b>USB Controller 1 Status (USB1_STS)—R/WC.</b> 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware when USB Controller 1 needs to cause a wake. Wake event will be generated if the corresponding USB1_EN bit is set.
2	Reserved.
1	<b>Thermal Interrupt Override Status (THRMOR_STS)—R/WC.</b> 0 = Software clears this bit by writing a 1 to the bit position. 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	<b>Thermal Interrupt Status (THRM_STS)—R/WC.</b> 0 = Software clears this bit by writing a 1 to the bit position. 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#).

### 9.8.3.8 GPE0\_EN—General Purpose Event 0 Enables Register

I/O Address:	PMBASE + 2Ah (ACPI GPE0_BLK + 2)	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI
Power Well:	Bits 0–7 Resume, Bits 8–15 RTC		

**Note:** 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. The resume well bits are all cleared by RSMRST#. The RTC sell bits are cleared by RTCRST#.

Bit	Description
15:12	Reserved.
11	<b>PME# Enable (PME_EN)</b> —R/W. 0 = Disable. 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).
9	Reserved
8	<b>RI_EN</b> —R/W. 0 = Disable. 1 = Enables the setting of the RI_STS to generate a wake event.
7	Reserved
6	<b>TCO SCI Enable (TCOSCI_EN)</b> —R/W. 0 = Disable. 1 = Enables the setting of the TCOSCI_STS to generate an SCI.
5	<b>AC97 Enable (AC97_EN)</b> —R/W. 0 = Disable. 1 = Enables the setting of the AC97_STS to generate a wake event.
4	<b>USB Controller 2 Enable (USB2_EN)</b> —R/W. 0 = Disable. 1 = Enables the setting of the USB2_STS to generate a wake event.
3	<b>USB Controller 1 Enable (USB1_EN)</b> —R/W. 0 = Disable. 1 = Enables the setting of the USB1_STS to generate a wake event.
2	<b>Thermal Pin Polarity (THRM#_POL)</b> —R/W. This bit controls the polarity of the THRM# pin needed to set the THRM_STS bit. 0 = Low value on the THRM# signal will set the THRM_STS bit. 1 = HIGH value on the THRM# signal will set the THRM_STS bit.
1	Reserved.
0	<b>Thermal Signal Reporting Enable (THRM_EN)</b> —R/W. 0 = Disable. 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).

### 9.8.3.9 GPE1\_STS—General Purpose Event 1 Status Register

I/O Address:	PMBASE + 2Ch (ACPI GPE1_BLK)	Attribute:	R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI
Power Well:	Resume		

**Note:** This register is symmetrical to the General Purpose Event 1 Enable Register. GPIOs that are not implemented will not have the corresponding bits implemented in this register.

**Note:** Bits 5 and 2 are not implemented since GPIO5 and GPIO2 are not implemented.

Bit	Description
15:6	<b>GPI[15:6] Status (GPI[15:6]_STS)—R/WC.</b> 0 = Software clears each bit by writing a 1 to the bit position when the corresponding GPIO signal is not active. (The status bit cannot be cleared while the corresponding signal is still active). 1 = These bits are set any time the corresponding GPIO is set up as an input and the corresponding GPIO signal is low (or high if the corresponding GP_INV bit is set).  If the corresponding GPI[n]_EN bit is set in the GPE1_EN register, and the GPI[n]_STS bit is set, then: - If the system is in an S1_S5 state, the event will also wake the system. - If the system is in an S0 state (or upon waking back to an S0 state), an SMI# or SCI will be generated, depending on the GPI_ROUT bits for the corresponding GPI.
5	Reserved
4:3	<b>GPI[4:3] Status (GPI[4:3]_STS)—R/WC.</b> 0 = Software clears each bit by writing a 1 to the bit position when the corresponding GPIO signal is not active. (The status bit cannot be cleared while the corresponding signal is still active). 1 = These bits are set any time the corresponding GPIO is set up as an input and the corresponding GPIO signal is low (or high if the corresponding GP_INV bit is set).  If the corresponding GPI[n]_EN bit is set in the GPE1_EN register, and the GPI[n]_STS bit is set, then: - If the system is in an S1_S5 state, the event will also wake the system. - If the system is in an S0 state (or upon waking back to an S0 state), an SMI# or SCI will be generated, depending on the GPI_ROUT bits for the corresponding GPI.
2	Reserved
1:0	<b>GPI[1:0] Status (GPI[1:0]_STS)—R/WC.</b> 0 = Software clears each bit by writing a 1 to the bit position when the corresponding GPIO signal is not active. (The status bit cannot be cleared while the corresponding signal is still active). 1 = These bits are set any time the corresponding GPIO is set up as an input and the corresponding GPIO signal is low (or high if the corresponding GP_INV bit is set).  If the corresponding GPI[n]_EN bit is set in the GPE1_EN register, and the GPI[n]_STS bit is set, then: - If the system is in an S1_S5 state, the event will also wake the system. - If the system is in an S0 state (or upon waking back to an S0 state), an SMI# or SCI will be generated, depending on the GPI_ROUT bits for the corresponding GPI.

### 9.8.3.10 GPE1\_EN—General Purpose Event 1 Enable Register

I/O Address:	PMBASE + 2Eh (ACPI GPE1_BLK + 2)	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI
Power Well:	Resume		

**Note:** This register is symmetrical to the General Purpose Event 1 Status Register. GPIOs that are not implemented will not have the corresponding bits implemented in this register. All of the bits in this register will be cleared by RSMRST#.

**Note:** Bits 5 and 2 are not implemented since GPIO5 and GPIO2 are not implemented.

Bit	Description
15:6	<b>GPI[15:6] Enable (GPI[15:6]_EN)</b> —R/W. 1 = Enable the corresponding GPI[n]_STS bit being set to cause an SMI#, SCI, and/or wake event. 0 = Disable.
5	Reserved
4:3	<b>GPI[4:3] Enable (GPI[4:3]_EN)</b> —R/W. 1 = Enable the corresponding GPI[n]_STS bit being set to cause an SMI#, SCI, and/or wake event. 0 = Disable.
2	Reserved
1:0	<b>GPI[1:0] Enable (GPI[1:0]_EN)</b> —R/W. 1 = Enable the corresponding GPI[n]_STS bit being set to cause an SMI#, SCI, and/or wake event. 0 = Disable.

### 9.8.3.11 SMI\_EN—SMI Control and Enable Register

I/O Address:	PMBASE + 30h	Attribute:	R/W
Default Value:	0000h	Size:	32 bit
Lockable:	No	Usage:	ACPI or Legacy
Power Well:	Core		

Bit	Description
31:15	Reserved
14	<b>Periodic SMI# Enable (PERIODIC_EN)</b> —R/W. 0 = Disable. 1 = Enables the ICH2 to generate an SMI# when the PERIODIC_STS bit is set in the SMI_STS register.
13	<b>TCO Enable (TCO_EN)</b> —R/W. 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. 1 = Enables the TCO logic to generate SMI#.
12	Reserved
11	<b>Microcontroller SMI# Enable (MCSMI_EN)</b> —R/W. 0 = Disable. 1 = Enables ICH2 to trap accesses to the microcontroller range (62h or 66h) and generate an SMI#. Note that 'trapped' cycles will be claimed by the ICH2 on PCI, but not forwarded to LPC.
10:8	Reserved

Bit	Description
7	<b>BIOS Release (BIOS_RLS)—WO.</b> 0 = This bit will always return 0 on reads. Writes of 0 to this bit have no effect. 1 = Enables the generation of an SCI interrupt for ACPI software when a one is written to this bit position by BIOS software.
6	<b>Software SMI# Timer Enable (SWSMI_TMR_EN)—R/W.</b> 0 = Disable. Clearing the SWSMI_TMR_EN bit before the timer expires will reset the timer and the SMI# will not be generated. 1 = Starts Software SMI# Timer. When the SWSMI timer expires (the time-out 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	<b>APMC Enable (APMC_EN)—R/W.</b> 0 = Disable. Writes to the APM_CNT register will not cause an SMI#. 1 = Enables writes to the APM_CNT register to cause an SMI#.
4	<b>SLP SMI Enable (SLP_SMI_EN)—R/W.</b> 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. 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.
3	<b>Legacy USB Enable (LEGACY_USB_EN)—R/W.</b> 0 = Disable. 1 = Enables legacy USB circuit to cause SMI#.
2	<b>BIOS Enable (BIOS_EN)—R/W.</b> 0 = Disable. 1 = Enables the generation of SMI# when ACPI software writes a 1 to the GBL_RLS bit.
1	<b>End of SMI (EOS)—R/W (special).</b> This bit controls the arbitration of the SMI signal to the processor. This bit must be set for the ICH2 to assert SMI# low to the processor. 1 = When this bit is set, 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. 0 = Once the ICH2 asserts SMI# low, the EOS bit is automatically cleared.
0	<b>Global SMI Enable (GBL_SMI_EN)—R/W.</b> 0 = No SMI# will be generated by ICH2. This bit is reset by a PCI reset event. 1 = Enables the generation of SMI# in the system upon any enabled SMI event.

### 9.8.3.12 SMI\_STS—SMI Status Register

I/O Address:	PMBASE + 34h	Attribute:	R/W
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 ICH2 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).

Bit	Description
31:17	Reserved
16	<b>SMBus SMI Status (SMBUS_SMI_STS)—R/WC.</b> 1 = Indicates that the SMI# was caused by either the SMBus Slave receiving a message, or the SMBALERT# signal going active. This bit will be set on SMBALERT# assertion only if the SMBus Host Controller is programmed to generate SMIs (not interrupts). 0 = This bit is cleared by writing a 1 to its bit position.
15	<b>SERR IRQ SMI Status (SERIRQ_SMI_STS)—RO.</b> 1 = Indicates that the SMI# was caused by the SERIRQ decoder. 0 = SMI# was not caused by SERIRQ decoder. This is not a sticky bit.
14	<b>Periodic Status (PERIODIC_STS)—R/WC.</b> 1 = This bit will be set at the rate determined by the PER_SMI_SEL bits. If the PERIODIC_EN bit is also set, the ICH2 will generate an SMI#. 0 = This bit is cleared by writing a 1 to its bit position.
13	<b>TCO Status (TCO_STS)—RO.</b> 0 = SMI# not caused by TCO logic. 1 = Indicates the SMI# was caused by the TCO logic. Note that this is not a wake event.
12	<b>Device Monitor Status (DEVMON_STS)—RO.</b> 1 = Set under any of the following conditions: - Any of the DEV[7:4]_TRAP_STS bits are set and the corresponding DEV[7:4]_TRAP_EN bits are also set. - Any of the DEVTRAP_STS bits are set and the corresponding DEVTRAP_EN bits are also set. 0 = SMI# not caused by Device Monitor.
11	<b>Microcontroller SMI# Status (MCSMI_STS)—R/WC.</b> 0 = Indicates that there has been no access to the power management microcontroller range (62h or 66h). This bit is cleared by software writing a 1 to the bit position. 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 ICH2 will generate an SMI#.
10	<b>GPE1 Status (GPE1_STS)—RO.</b> This bit is a logical OR of the bits in the GPE1_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 GPE1_EN register. Bits that are not routed to cause an SMI# will have no effect on the GPE1_STS bit. 0 = SMI# was not generated by a GPI assertion. 1 = SMI# was generated by a GPI assertion.
9	<b>GPE0 Status (GPE0_STS)—RO.</b> 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. 0 = SMI# was not generated by a GPE0 event. 1 = SMI# was generated by a GPE0 event.
8	<b>PM1 Status Register (PM1_STS_REG)—RO.</b> This is an OR of the bits in the ACPI PM1 Status Register (offset PMBASE+00h) that can cause an SMI#. 0 = SMI# was not generated by a PM1_STS event. 1 = SMI# was generated by a PM1_STS event.
7	Reserved.

Bit	Description
6	<b>Software SMI Timer Status (SWSMI_TMR_STS)</b> —R/WC. 1 = Set by the hardware when the Software SMI# Timer expires. 0 = Software clears this bit by writing a 1 to the bit location.
5	<b>APM Status (APM_STS)</b> —R/WC. 1 = SMI# was generated by a write access to the APM control register with the APMC_EN bit set. 0 = Software clears this bit by writing a 1 to the bit location.
4	<b>SLP SMI Status (SLP_SMI_STS)</b> —R/WC. 1 = Indicates an SMI# was caused by a write of 1 to SLP_EN bit when SLP_SMI_EN bit is also set. 0 = Software clears this bit by writing a 1 to the bit location.
3	<b>Legacy USB Status (LEGACY_USB_STS)</b> —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. 0 = SMI# was not generated by USB Legacy event. 1 = SMI# was generated by USB Legacy event.
2	<b>BIOS Status (BIOS_STS)</b> —R/WC. 1 = SMI# was generated due to ACPI software requesting attention (writing a 1 to the GBL_RLS bit with the BIOS_EN bit set). 0 = This bit cleared by software writing a 1 to its bit position.
1:0	Reserved.

### 9.8.3.13 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	<b>Device 7:4 Trap Status (DEV[7:4]_TRAP_STS)</b> —R/WC. Bit 12 corresponds to Monitor 4, bit 13 corresponds to Monitor 5 etc. 1 = SMI# was caused by an access to the corresponding device monitor's I/O range. 0 = SMI# was not caused by the associated device monitor.
11:8	<b>Device 7:4 Trap Enable (DEV[7:4]_TRAP_EN)</b> —R/W. Bit 8 corresponds to Monitor 4, bit 9 corresponds to Monitor 5 etc. 1 = Enables SMI# due to an access to the corresponding device monitor's I/O range. 0 = Disable.
7:0	Reserved

### 9.8.3.14 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.

Bit	Description
15:14	Reserved
13	<b>ADLIB Activity Status (ADLIB_ACT_STS)—R/WC.</b> 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
12	<b>Keyboard Controller Activity Status (KBC_ACT_STS)—R/WC.</b> KBC (60/64h). 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
11	<b>MIDI Activity Status (MIDI_ACT_STS)—R/WC.</b> 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
10	<b>Audio Activity Status (AUDIO_ACT_STS)—R/WC.</b> Audio (Sound Blaster "ORed" with MSS). 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
9	<b>PIRQ[D or H] Activity Status (PIRQDH_ACT_STS)—R/WC.</b> 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.
8	<b>PIRQ[C or G] Activity Status (PIRQCG_ACT_STS)—R/WC.</b> 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.
7	<b>PIRQ[B or F] Activity Status (PIRQBF_ACT_STS)—R/WC.</b> 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.
6	<b>PIRQ[A or E] Activity Status (PIRQAE_ACT_STS)—R/WC.</b> 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.
5	<b>Legacy Activity Status (LEG_ACT_STS)—R/WC.</b> Parallel Port, Serial Port 1, Serial Port 2, Floppy Disk Controller. 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
4	Reserved.
3	<b>IDE Secondary Drive 1 Activity Status (IDES1_ACT_STS)—R/WC.</b> 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
2	<b>IDE Secondary Drive 0 Activity Status (IDES0_ACT_STS)—R/WC.</b> 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.

Bit	Description
1	<b>IDE Primary Drive 1 Activity Status (IDEP1_ACT_STS)</b> —R/WC. 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
0	<b>IDE Primary Drive 0 Activity Status (IDEP0_ACT_STS)</b> —R/WC. 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.

### 9.8.3.15 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
15:14	Reserved
13	<b>ADLIB Trap Enable (ADLIB_TRP_EN)</b> —R/W. 0 = Disable. 1 = Enable.
12	<b>KBC Trap Enable (KBC_TRP_EN)</b> —R/W. KBC (60/64h). 0 = Disable. 1 = Enable.
11	<b>MIDI Trap Enable (MIDI_TRP_EN)</b> —R/W. 0 = Disable. 1 = Enable.
10	<b>Audio Trap Enable (AUDIO_TRP_EN)</b> —R/W. Audio (Sound Blaster “ORed” with MSS). 0 = Disable. 1 = Enable.
9:6	Reserved
5	<b>LEG_IO_TRP_EN</b> —R/W. Parallel Port, Serial Port 1, Serial Port 2, Floppy Disk Controller. 0 = Disable. 1 = Enable.
4	Reserved.
3	<b>IDE Secondary Drive 1 Trap Enable (IDES1_TRP_EN)</b> —R/W. 0 = Disable. 1 = Enable.
2	<b>IDE Secondary Drive 0 Trap Enable (IDES0_TRP_EN)</b> —R/W. 0 = Disable. 1 = Enable.
1	<b>IDE Primary Drive 1 Trap Enable (IDEP1_TRP_EN)</b> —R/W. 0 = Disable. 1 = Enable.
0	<b>IDE Primary Drive 0 Trap Enable (IDEP0_TRP_EN)</b> —R/W. 0 = Disable. 1 = Enable.

### 9.8.3.16 BUS\_ADDR\_TRACK—Bus Address Tracker Register

I/O Address:	PMBASE +4Ch	Attribute:	RO
Lockable:	No	Size:	16-bit
Power Well:	Core	Usage:	Legacy Only

This register could be used by the SMI# handler to assist in determining what was the last cycle from the processor.

Bit	Description
15:0	Corresponds to the low 16 bits of the last I/O cycle, as would be defined by the PCI AD[15:0] signals on the PCI bus (even though it may not be a real PCI cycle). The value is latched based on SMI# active. This functionality is useful for figuring out which I/O was last being accessed.

### 9.8.3.17 BUS\_CYC\_TRACK—Bus Cycle Tracker Register

I/O Address:	PMBASE +4Eh	Attribute:	RO
Lockable:	No	Size:	8-bit
Power Well:	Core	Usage:	Legacy Only

This register could be used by the SMM handler to assist in determining what was the last cycle from the processor.

Bit	Description
7:4	Corresponds to the byte enables, as would be defined by the PCI C/BE# signals on the PCI bus (even though it may not be a real PCI cycle). The value is latched based on SMI# going active.
3:0	Corresponds to the cycle type, as would be defined by the PCI C/BE# signals on the PCI bus (even though it may not be a real PCI cycle). The value is latched based on SMI# going active.

## 9.9 System Management TCO Registers (D31:F0)

The TCO logic is accessed via registers mapped to the PCI configuration space (Device 31:Function 0) and the system I/O space. For TCO PCI Configuration registers, see LPC Device 31:Function 0 PCI Configuration registers.

### 9.9.1 TCO Register I/O Map

The TCO I/O registers reside in a 32-byte range pointed to by a TCOBASE value, which is, ACPIBASE + 60h in the PCI configuration space. The following table shows the mapping of the registers within that 32-byte range. Each register is described in the sections below.

**Table 9-11. TCO I/O Register Map**

Offset	Type	Mnemonic	Register Name: Function
00h	R/W	TCO_RLD	TCO Timer Reload and Current Value
01h	R/W	TCO_TMR	TCO Timer Initial Value
02h	R/W	TCO_DAT_IN	TCO Data In
03h	R/W	TCO_DAT_OUT	TCO Data Out
04h–05h	R/W	TCO1_STS	TCO Status
06h–07h	R/W	TCO2_STS	TCO Status
08h–09h	R/W	TCO1_CNT	TCO Control
0Ah–0Bh	R/W	TCO2_CNT	TCO Control
0Ch–0Dh	R/W	TCO_MESSAGE1, TCO_MESSAGE2	Used by BIOS to indicate POST/Boot progress
0Eh	R/W	TCO_WDSTATUS	Watchdog Status Register
0Fh	RO		Reserved
10h	R/W	SW_IRQ_GEN	Software IRQ Generation Register
11h–1Fh	RO		Reserved

### 9.9.2 TCO1\_RLD—TCO Timer Reload and Current Value Register

I/O Address:	TCOBASE +00h	Attribute:	R/W
Default Value:	0000h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:0	<b>TCO Timer Value.</b> 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 time-out. Bits 7:6 will always be 0.

### 9.9.3 TCO1\_TMR—TCO Timer Initial Value Register

I/O Address: TCOBASE +01h      Attribute: R/W  
 Default Value: 0004h      Size: 8-bit  
 Lockable: No      Power Well: Core

Bit	Description
7:6	Reserved
5:0	<b>TCO Timer Initial Value.</b> Value that is loaded into the timer each time the TCO_RLD register is written. Values of 0h–3h will be ignored and should not be attempted. The timer is clocked at approximately 0.6 seconds, and this allows time-outs ranging from 2.4 seconds to 38 seconds.

### 9.9.4 TCO1\_DAT\_IN—TCO Data In Register

I/O Address: TCOBASE +02h      Attribute: R/W  
 Default Value: 0000h      Size: 8-bit  
 Lockable: No      Power Well: Core

Bit	Description
7:0	<b>TCO Data In Value.</b> Data Register for passing commands from the OS to the SMI handler. Writes to this register will cause an SMI and set the OS_TCO_SMI bit in the TCO_STS register.

### 9.9.5 TCO1\_DAT\_OUT—TCO Data Out Register

I/O Address: TCOBASE +03h      Attribute: R/W  
 Default Value: 0000h      Size: 8-bit  
 Lockable: No      Power Well: Core

Bit	Description
7:0	<b>TCO Data Out Value.</b> Data Register 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.

### 9.9.6 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	<b>Hub Interface SERR Status (HUBSERR_STS)—R/WC.</b> 1 = ICH2 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#. 0 = Software clears this bit by writing a 1 to the bit position.
11	<b>Hub Interface NMI Status (HUBNMI_STS)—R/WC.</b> 1 = ICH2 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. 0 = Software clears this bit by writing a 1 to the bit position.

Bit	Description
10	<b>Hub Interface SMI Status (HUBSMI_STS)—R/WC.</b> 1 = ICH2 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#. 0 = Software clears this bit by writing a 1 to the bit position.
9	<b>Hub Interface SCI Status (HUBSCI_STS)—R/WC.</b> 1 = ICH2 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. 0 = Software clears this bit by writing a 1 to the bit position.
8	<b>BIOS Write Status (BIOSWR_STS)—R/WC.</b> 1 = ICH2 sets this bit and generates an SMI# to indicate an illegal attempt to write to the BIOS. This occurs when either: a) The BIOSWP bit is changed from 0 to 1 and the BLD bit is also set, or b) any write is attempted to the BIOS and the BIOSWP bit is also set. 0 = Software clears this bit by writing a 1 to the bit position. Note: On write cycles attempted to the 4 MB lower alias to the BIOS space, the BIOSWR_STS will not be set.
7	<b>New Century Status (NEWCENTURY_STS)—R/WC.</b> This bit is in the RTC well. 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). 0 = Cleared by writing a 1 to the bit position or by RTCRST# going active. 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 (e.g., 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. 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	<b>Time Out Status (TIMEOUT)—R/WC.</b> 1 = Set by ICH2 to indicate that the SMI was caused by the TCO timer reaching 0. 0 = Software clears this bit by writing a 1 to the bit position.
2	<b>TCO Interrupt Status (TCO_INT_STS)—R/WC.</b> 1 = SMI handler caused the interrupt by writing to the TCO_DAT_OUT register. 0 = Software clears this bit by writing a 1 to the bit position.
1	<b>Software TCO SMI Status (SW_TCO_SMI)—R/WC.</b> 1 = Software caused an SMI# by writing to the TCO_DAT_IN register. 0 = Software clears this bit by writing a 1 to the bit position.
0	<b>NMI to SMI Status (NMI2SMI_STS)—RO.</b> 1 = Set by the ICH2 when an SMI# occurs because an event occurred that would otherwise have caused an NMI (because NMI2SMI_EN is set). 0 = Cleared by clearing the associated NMI status bit.

### 9.9.7 TCO2\_STS—TCO2 Status Register

I/O Address:	TCOBASE +06h	Attribute:	R/WC, RO
Default Value:	0000h	Size:	16-bit
Lockable:	No	Power Well:	Resume (Except Bit 0, in RTC)

Bit	Description
15:3	Reserved
2	<p><b>Boot Status (BOOT_STS):</b>  1 = Set to 1 when the SECOND_TO_STS bit goes from 0 to 1 and the processor has not fetched the first instruction.  0 = Cleared by ICH2 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.  If rebooting due to a second TCO timer time-out and if the BOOT_STS bit is set, the ICH2 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.</p>
1	<p><b>Second TCO Time-out Status (SECOND_TO_STS)—R/WC.</b>  1 = The ICH2 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 the ICH2 will reboot the system after the second time-out. The reboot is done by asserting PCIRST#.  0 = This bit is cleared by writing a 1 to the bit position or by a RSMRST#.</p>
0	<p><b>Intruder Detect (INTRD_DET)—R/WC.</b>  1 = Set by ICH2 to indicate that an intrusion was detected. This bit is set even if the system is in G3 state.  0 = This bit is only cleared by writing a 1 to the bit position, or by RTCRST# assertion.</p>

## 9.9.8 TCO1\_CNT—TCO1 Control Register

I/O Address:	TCOBASE +08h	Attribute:	R/W, R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Power Well:	Core

Bit	Description															
15:12	Reserved															
11	<b>TCO Timer Halt (TCO_TMR_HLT)</b> —R/W. 0 = The TCO Timer is enabled to count. 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 prevents rebooting and prevents Alert On LAN event messages from being transmitted on the SMLINK (but not Alert On LAN heartbeat messages).															
10	<b>Send Now (SENDNOW)</b> —R/W (special). 1 = Writing a 1 to this bit will cause the ICH to send an Alert On LAN Event message over the SMLINK interface, with the Software Event bit set. 0 = The ICH will clear this bit when it has completed sending the message. Software must not set this bit to 1 again until the ICH has set it back to 0.  Setting the SENDNOW bit causes the ICH2 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	<b>NMI to SMI Enable (NMI2SMI_EN)</b> —R/W. 0 = Normal NMI functionality. 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: <table><tr><th>NMI_EN</th><th>GBL_SMI_EN</th><th>Description</th></tr><tr><td>0</td><td>0</td><td>No SMI# at all because GBL_SMI_EN = 0</td></tr><tr><td>0</td><td>1</td><td>SMI# will be caused due to NMI events</td></tr><tr><td>1</td><td>0</td><td>No SMI# at all because GBL_SMI_EN = 0</td></tr><tr><td>1</td><td>1</td><td>No SMI# due to NMI because NMI_EN = 1</td></tr></table>	NMI_EN	GBL_SMI_EN	Description	0	0	No SMI# at all because GBL_SMI_EN = 0	0	1	SMI# will be caused due to NMI events	1	0	No SMI# at all because GBL_SMI_EN = 0	1	1	No SMI# due to NMI because NMI_EN = 1
NMI_EN	GBL_SMI_EN	Description														
0	0	No SMI# at all because GBL_SMI_EN = 0														
0	1	SMI# will be caused due to NMI events														
1	0	No SMI# at all because GBL_SMI_EN = 0														
1	1	No SMI# due to NMI because NMI_EN = 1														
8	<b>NMI Now (NMI_NOW)</b> —R/WC. 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. 0 = This bit is cleared by writing a 1 to the bit position. The NMI handler is expected to clear this bit. Another NMI will not be generated until the bit is cleared.															
7:0	Reserved															

### 9.9.9 TCO2\_CNT—TCO2 Control Register

I/O Address: TCOBASE +0Ah      Attribute: R/W  
 Default Value: 0000h      Size: 16-bit  
 Lockable: No      Power Well: Resume

Bit	Description
15:3	Reserved.
2:1	<b>INTRUDER# Signal Select (INTRD_SEL)</b> —R/W. Selects the action to take if the INTRUDER# signal goes active. 00 = No interrupt or SMI# 01 = Interrupt (as selected by TCO_INT_SEL). 10 = SMI 11 = Reserved
0	Reserved.

### 9.9.10 TCO\_MESSAGE1 and TCO\_MESSAGE2 Registers

I/O Address: TCOBASE +0Ch (Message 1)      Attribute: R/W  
                   TCOBASE +0Dh (Message 2)  
 Default Value: 00h      Size: 8-bit  
 Lockable: No      Power Well: Resume

Bit	Description
7:0	<b>TCO Message (TCO_MESSAGE[n])</b> —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.

### 9.9.11 TCO\_WDSTATUS—TCO2 Control Register

Offset Address: TCOBASE + 0Eh      Attribute: R/W  
 Default Value: 00h      Size: 8 bits  
 Power Well: Resume

Bit	Description
7:0	<b>Watchdog Status (WDSTATUS)</b> —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).

## 9.9.12 SW\_IRQ\_GEN—Software IRQ Generation Register

Offset Address:	TCOBASE + 10h	Attribute:	R/W
Default Value:	03h	Size:	8 bits
Power Well:	Resume		

Bit	Description
7:2	Reserved.
1	<b>IRQ12 Cause (IRQ12_CAUSE)</b> —R/W. The state of this bit is logically ANDed with the IRQ12 signal as received by the ICH2's SERIRQ logic. This bit must be a "1" (default) if the ICH2 is expected to receive IRQ12 assertions from a SERIRQ device.
0	<b>IRQ1 Cause (IRQ1_CAUSE)</b> —R/W. The state of this bit is logically ANDed with the IRQ1 signal as received by the ICH2's SERIRQ logic. This bit must be a "1" (default) if the ICH2 is expected to receive IRQ1 assertions from a SERIRQ device.

## 9.10 General Purpose I/O Registers (D31:F0)

The control for the general purpose I/O signals is handled through a separate 64-byte I/O space. The base offset for this space is selected by the GPIO\_BAR register. Table 9-12 summarizes the ICH2 GPIO implementation.

**Table 9-12. Summary of GPIO Implementation**

GPIO	Type	Alternate Function (Note 1)	Power Well	Notes
GPIO[0]	Input Only	REQ[A]#	Core	GPIO_USE_SEL bit 0 enables REQ/GNT[A]# pair. Input active status read from GPE1_STS register bit 0. Input active high/low set through GPI_INV register bit 0.
GPIO[1]	Input Only	REQ[B]# or REQ[5]#	Core	GPIO_USE_SEL bit 1 enables REQ/GNT[B]# pair (See note 4). Input active status read from GPE1_STS register bit 1. Input active high/low set through GPI_INV register bit 1.
GPIO[2]	N/A	N/A	N/A	Not implemented
GPIO[3:4]	Input Only	PIRQ[E:H]#	Core	GPIO_USE_SEL bits [3:4] enable PIRQ[F:G]#. Input active status read from GPE1_STS reg. bits [3:4]. Input active high/low set through GPI_INV reg. bit [3:4].
GPIO[5]	N/A	N/A	N/A	Not implemented
GPIO[6]	Input Only	Unmuxed	Core	Input active status read from GPE1_STS register bit 6. Input active high/low set through GPI_INV register bit 6.
GPIO[7]	Input Only	Unmuxed	Core	Input active status read from GPE1_STS register bit 7. Input active high/low set through GPI_INV register bit 7.
GPIO[8]	Input Only	Unmuxed	Resume	Input active status read from GPE1_STS register bit 8. Input active high/low set through GPI_INV register bit 8.
GPIO[9:10]	N/A	N/A	N/A	Not implemented
GPIO[11]	Input Only	SMBALERT#	Resume	GPIO_USE_SEL bit 11 enables SMBALERT#. Input active status read from GPE1_STS register bit 11. Input active high/low set through GPI_INV register bit 11.
GPIO[12]	Input Only	Unmuxed	Resume	Input active status read from GPE1_STS register bit 12. Input active high/low set through GPI_INV register bit 12.

Table 9-12. Summary of GPIO Implementation (Continued)

GPIO	Type	Alternate Function (Note 1)	Power Well	Notes
GPIO[13]	Input Only	Unmuxed	Resume	Input active status read from GPE1_STS register bit 13. Input active high/low set through GPI_INV register bit 13.
GPIO[14:15]	N/A	N/A	N/A	Not Implemented
GPIO[16]	Output Only	GNT[A]#	Core	Output controlled via GP_LVL register bit 16. TTL driver output
GPIO[17]	Output Only	GNT[B]# or GNT[5]#	Core	Output controlled via GP_LVL register bit 17. TTL driver output
GPIO[18:19]	Output Only	Unmuxed	Core	Output controlled via GP_LVL register bits [18:19]. TTL driver output
GPIO[20]	Output Only	Unmuxed	Core	Output controlled via GP_LVL register bit 20. TTL driver output
GPIO[21]	Output Only	Unmuxed	Core	This GPO defaults high. Output controlled via GP_LVL register bit 21. TTL driver output
GPIO[22]	Output Only	Unmuxed	Core	Output controlled via GP_LVL register bit [22]. Open-drain output
GPIO[23]	Output Only	Unmuxed	Core	Output controlled via GP_LVL register bit [23]. TTL driver output
GPIO[24]	Input / Output	Unmuxed	Resume	Input active status read from GP_LVL register bit 24. Output controlled via GP_LVL register bit 24. TTL driver output
GPIO[25]	Input / Output	Unmuxed	Resume	Blink enabled via GPO_BLINK register bit 25. Input active status read from GP_LVL register bit 25. Output controlled via GP_LVL register bit 25. TTL driver output
GPIO[26]	N/A	N/A	N/A	Not implemented
GPIO[27:28]	Input / Output	Unmuxed	Resume	Input active status read from GP_LVL register bits [27:28]. Output controlled via GP_LVL register bits [27:28]. TTL driver output
GPIO[29:31]	N/A	N/A	N/A	Not implemented

**NOTES:**

1. All GPIOs default to their alternate function.
2. All inputs are sticky. The status bit will remain set as long as the input was asserted for 2 clocks. GPIs are sampled on PCI clocks in S0/S1. GPIs are sampled on RTC clocks in S3/S4/S5.
3. GPIO[7:6,4:3,1:0] are 5V 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/GNT[5]# is enabled. See [Section 9.1.22](#).

## 9.10.1 GPIO Register I/O Address Map

Table 9-13. Registers to Control GPIO

Offset	Mnemonic	Register Name	Default	Access
<b>General Registers</b>				
00–03h	GPIO_USE_SEL	GPIO Use Select	1A00 3180h	R/W
04–07h	GP_IO_SEL	GPIO Input/Output Select	0000 FFFFh	R/W
08–0Bh	—	Reserved	00h	RO
0C–0Fh	GP_LVL	GPIO Level for Input or Output	1F1F 0000h	R/W
10–13h	—	Reserved	00h	RO
<b>Output Control Registers</b>				
14–17h	GPO_TTL	GPIO TTL Select	06630000h	RO
18–1Bh	GPO_BLINK	GPIO Blink Enable	00000000h	R/W
1C–1Fh	—	Reserved	0	RO
<b>Input Control Registers</b>				
20–2Bh	—	Reserved	00000000h	RO
2C–2Fh	GPI_INV	GPIO Signal Invert	00000000h	R/W

## 9.10.2 GPIO\_USE\_SEL—GPIO Use Select Register

Offset Address: GPIOBASE + 00h      Attribute: R/W  
Default Value: 1A003180h      Size: 32-bit  
Lockable: Yes      Power Well: Resume

Bit	Description
21,11,5:0	<p><b>GPIO Use Select (GPIO_USE_SEL)</b>—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.</p> <p>0 = Signal used as native function.  1 = Signal used as a GPIO.</p> <p><b>Note:</b> Bits 31:29, 26, 15:14, 10:9 and 7 are not implemented because there is no corresponding GPIO.</p> <p><b>Note:</b> Bits 28:27, 25:22, 20:18, 13:12, 8 and 6 are not implemented because the corresponding GPIOs are not multiplexed.</p> <p><b>Note:</b> Bits 28:27, 25:22, 20:18, 13:12 and 8:6 are not implemented because the corresponding GPIOs are not multiplexed.</p>

### 9.10.3 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 25:24	<b>GPIO[n] Select (GPIO[n]_SEL)</b> —R/W. 0 = Output. The corresponding GPIO signal is an output. 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.

#### NOTES:

- There will be some delay on GPIO[24:28] going to their default state based on the rising edge of RSMRST#. This is the case since these signals are in the resume well and resume well outputs are not valid until after RSMRST# goes high. ICH2 only guarantees that these GPIOs will be stable prior to SLP\_S3# going active.

### 9.10.4 GP\_LVL—GPIO Level for Input or Output Register

Offset Address: GPIOBASE +0Ch      Attribute: R/W, RO  
 Default Value: 1B3F 0000h      Size: 32-bit  
 Lockable: No      Power Well: See bit descriptions

**Note:** Bits 5 and 2 are not implemented since GPIO[5,2] are not implemented.

Bit	Description
31:29, 26	Reserved.
28:27, 25:24	<b>GPIO Level (GP_LVL[n])</b> —R/W. If GPIO[n] 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 GPIO[n] 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# but not by PCIRST#. 0 = Low 1 = High
23:16	<b>GPIO Level GP_LVL[n]</b> —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#. 0 = Low 1 = High
15:6, 4:3, 1:0	Reserved. For GPI[13:11] and [8:6,4:3,1:0], the active status of a GPI is read from the corresponding bit in GPE1_STS register.

## 9.10.5 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	<p><b>GPIO Blink (GP_BLINK[n])</b>—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# but not by PCIRST#.</p> <p>0 = The corresponding GPIO will function normally.  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 50% duty cycle. The GP_LVL bit is not altered when this bit is set.</p>
19:18	<p><b>GPIO Blink (GP_BLINK[n])</b>—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#.</p> <p>0 = The corresponding GPIO will function normally.  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 50% duty cycle. The GP_LVL bit is not altered when this bit is set.</p>

**Note:** GPIO[18] blinks, by default, immediately after reset. This signal could be connected to an LED to indicate a failed boot (by programming BIOS to clear GP\_BLINK[18] after successful POST).

## 9.10.6 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

**Note:** Bits 5 and 2 are not implemented since GPIO[5,2] are not implemented.

Bit	Description
31:14, 10:9	Reserved
13:11, 8	<p><b>GPIO Signal High/Low Select (GP_INV[n])</b>—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 ICH2. 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 will have 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# but not by PCIRST#.</p> <p>0 = The corresponding GPI_STS bit will be set when the ICH2 detects the state of the input pin to be high.</p> <p>1 = The corresponding GPI_STS bit will be set when the ICH2 detects the state of the input pin to be low.</p>
7:6, 4:3, 1:0	<p><b>GPIO Signal High/Low Select (GP_INV[n])</b>—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 ICH2. 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#.</p> <p>0 = The corresponding GPI_STS bit will be set when the ICH2 detects the state of the input pin to be high.</p> <p>1 = The corresponding GPI_STS bit will be set when the ICH2 detects the state of the input pin to be low.</p>

# IDE Controller Registers (D31:F1) 10

## 10.1 PCI Configuration Registers (IDE—D31:F1)

**Note:** Registers that are not shown should be treated as Reserved (See [Section 6.2](#) for details).

All of the IDE registers are in the Core well. None can be locked.

**Table 10-1. PCI Configuration Map (IDE—D31:F1)**

Offset	Mnemonic	Register Name/Function	Default	Type
00h–01h	VID	Vendor ID	8086h	RO
02h–03h	DID	Device ID	244Bh	RO
04h–05h	CMD	Command Register	00h	R/W
06h–07h	STS	Device Status	0280h	R/W
08h	RID	Revision ID	See Note 1	RO
09h	PI	Programming Interface	80h	RO
0Ah	SCC	Sub Class Code	01h	RO
0Bh	BCC	Base Class Code	01h	RO
0Dh	MLT	Master Latency Timer	00	RO
0Eh	HTYPE	Header Type	00h	RO
20h–23h	BAR	Base Address Register	00000001h	R/W
2C–2Dh	SVID	Subsystem Vendor ID	00	R/Write-Once
2E–2Fh	SID	Subsystem ID	00	R/Write-Once
40h–41h	IDE_TIMP	Primary IDE Timing	0000h	R/W
42–43h	ID_TIMS	Secondary IDE Timing	0000h	R/W
44h	SIDETIM	Slave IDE Timing	00h	R/W
48h	SDMAC	Synchronous DMA Control Register	00h	R/W
4Ah–4Bh	SDMATIM	Synchronous DMA Timing Register	0000h	R/W
54h	IDE_CONFIG	IDE I/O Configuration Register	00h	R/W

**NOTES:**

1. Refer to the Specification Update for the value of the Revision ID Register
2. The ICH2 IDE controller is not arbitrated as a PCI device; therefore, it does not need a master latency timer.

### 10.1.1 CMD—Command Register (IDE—D31:F1)

Address Offset: 04h–05h  
Default Value: 00h

Attribute: RO, R/W  
Size: 16 bits

Bit	Description
15:10	Reserved.
9	Fast Back to Back Enable (FBE)—RO. Reserved as 0.
8	SERR# Enable—RO. Reserved as 0.
7	Wait Cycle Control—RO. Reserved as 0.
6	Parity Error Response—RO. Reserved as 0.
5	VGA Palette Snoop—RO. Reserved as 0.
4	Postable Memory Write Enable (PMWE)—RO. Reserved as 0.
3	Special Cycle Enable (SCE)—RO. Reserved as 0.
2	<b>Bus Master Enable (BME)</b> —R/W. Controls the ICH2's ability to act as a PCI master for IDE Bus Master transfers.
1	Memory Space Enable (MSE)—RO. Reserved as 0.
0	<p><b>I/O Space Enable (IOSE)</b>—R/W. This bit controls access to the I/O space registers.</p> <p>0 = Disables access to the Legacy IDE ports (both Primary and Secondary) as well as the Bus Master IO registers.</p> <p>1 = Enable. Note that the Base Address register for the Bus Master registers should be programmed before this bit is set.</p> <p><b>Note:</b> Separate bits are provided (IDE Decode Enable, in the IDE Timing register) to independently disable the Primary or Secondary I/O spaces.</p>

## 10.1.2 STS—Device Status Register (IDE—D31:F1)

Address Offset: 06–07h Attribute: R/WC, RO  
Default Value: 0280h Size: 16 bits

Bit	Description
15	Detected Parity Error (DPE)—RO. Reserved as 0.
14	Signaled System Error (SSE)—RO. Reserved as 0.
13	<b>Received Master-Abort Status (RMA)</b> —R/WC. 1 = Bus Master IDE interface function, as a master, generated a master abort. 0 = Cleared by writing a 1 to it.
12	Reserved as 0—RO.
11	<b>Signaled Target-Abort Status (STA)</b> —R/WC. 1 = ICH2 IDE interface function is targeted with a transaction that the ICH2 terminates with a target abort. 0 = Cleared by writing a 1 to it.
10:9	<b>DEVSEL# Timing Status (DEVT)</b> —RO. 01 = Hardwired; however, the ICH2 does not have a real DEVSEL# signal associated with the IDE unit, so these bits have no effect.
8	Reserved as 0.
7	Reserved as 1.
6	Reserved as 0.
5	Reserved as 0.
4:0	Reserved

## 10.1.3 PI—Programming Interface (IDE—D31:F1)

Address Offset: 09h Attribute: RO  
Default Value: 80h Size: 8 bits

Bit	Description
7:0	<b>Programming Interface Value</b> —RO. 80h = The 1b in bit 7 indicates that this IDE controller is capable of bus master operation.

## 10.1.4 SCC—Sub Class Code (IDE—D31:F1)

Address Offset: 0Ah Attribute: RO  
Default Value: 01h Size: 8 bits

Bit	Description
7:0	<b>Sub Class Code</b> —RO. 01h = IDE device, in the context of a mass storage device.

### 10.1.5 BCC—Base Class Code (IDE—D31:F1)

Address Offset: 0Bh                      Attribute: RO  
 Default Value: 01h                      Size: 8 bits

Bit	Description
7:0	<b>Base Class Code</b> —RO. 01 = Mass storage device

### 10.1.6 MLT—Master Latency Timer (IDE—D31:F1)

Address Offset: 0Dh                      Attribute: RO  
 Default Value: 00h                      Size: 8 bits

Bit	Description
7:0	Bus Master Latency—RO. The IDE controller is implemented internally, and is not arbitrated as a PCI device, so it does not need a Master Latency Timer. Hardwired to 00h.

### 10.1.7 BM\_BASE—Bus Master Base Address Register (IDE—D31:F1)

Address Offset: 20h–23h                      Attribute: R/W  
 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	<b>Base Address</b> —R/W. Base address of the I/O space (16 consecutive I/O locations).
3:1	Reserved.
0	<b>Resource Type Indicator (RTE)</b> —RO. Hardwired to 1, indicating a request for IO space.

### 10.1.8 IDE\_SVID—Subsystem Vendor ID (IDE—D31:F1)

Address Offset: 2Ch–2Dh      Attribute: R/Write-Once  
Default Value: 00h      Size: 16 bits  
Lockable: No      Power Well: Core

Bit	Description
15:0	<b>Subsystem Vendor ID (SVID)</b> —R/Write-Once. 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.

### 10.1.9 IDE\_SID—Subsystem ID (IDE—D31:F1)

Address Offset: 2Eh–2Fh      Attribute: R/Write-Once  
Default Value: 00h      Size: 16 bits  
Lockable: No      Power Well: Core

Bit	Description
15:0	<b>Subsystem ID (SID)</b> —R/Write-Once. 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.

### 10.1.10 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	<b>IDE Decode Enable (IDE)</b> —R/W. Individually enable/disable 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. 0 = Disable. 1 = Enables the ICH2 to decode the associated Command Blocks (1F0h–1F7h for primary, 170h–177h for secondary) and Control Block (3F6h for primary and 376h for secondary).
14	<b>Drive 1 Timing Register Enable (SITRE)</b> —R/W. 0 = Use bits 13:12, 9:8 for both drive 0 and drive 1. 1 = Use bits 13:12, 9:8 for drive 0, and use the Slave IDE Timing register for drive 1
13:12	<b>IORDY Sample Point (ISP)</b> . The setting of these bits determine the number of PCI clocks between IDE IOR#/IOW# assertion and the first IORDY sample point. 00 = 5 clocks 01 = 4 clocks 10 = 3 clocks 11 = Reserved
11:10	Reserved.
9:8	<b>Recovery Time (RCT)</b> —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. 00 = 4 clocks 01 = 3 clocks 10 = 2 clocks 11 = 1 clock
7	<b>Drive 1 DMA Timing Enable (DTE1)</b> —R/W. 0 = Disable. 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	<b>Drive 1 Prefetch/Posting Enable (PPE1)</b> —R/W. 0 = Disable. 1 = Enable Prefetch and posting to the IDE data port for this drive.
5	<b>Drive 1 IORDY Sample Point Enable (IE1)</b> —R/W. 0 = Disable IORDY sampling for this drive. 1 = Enable IORDY sampling for this drive.
4	<b>Drive 1 Fast Timing Bank (TIME1)</b> —R/W. 0 = Accesses to the data port will use compatible timings for this drive. 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	<b>Drive 0 DMA Timing Enable (DTE0)</b> —R/W. 0 = Disable. 1 = Enable fast timing mode for DMA transfers only for this drive. PIO transfers to the IDE data port will run in compatible timing.

Bit	Description
2	<b>Drive 0 Prefetch/Posting Enable (PPE0)</b> —R/W. 0 = Disable prefetch and posting to the IDE data port for this drive. 1 = Enable prefetch and posting to the IDE data port for this drive.
1	<b>Drive 0 IORDY Sample Point Enable (IE0)</b> —R/W. 0 = Disable IORDY sampling is disabled for this drive. 1 = Enable IORDY sampling for this drive.
0	<b>Drive 0 Fast Timing Bank (TIME0)</b> —R/W. 0 = Accesses to the data port will use compatible timings for this drive. 1 = Accesses to the data port will use bits 13:12 for the IORDY sample point, and bits 9:8 for the recovery time

### 10.1.11 SLV\_IDETIM—Slave (Drive 1) IDE Timing Register (IDE—D31:F1)

Address Offset: 44h      Attribute: R/W  
Default Value: 00h      Size: 8 bits

Bit	Description
7:6	<b>Secondary Drive 1 IORDY Sample Point (SISP1)</b> —R/W. 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. 00 = 5 clocks 01 = 4 clocks 10 = 3 clocks 11 = Reserved
5:4	<b>Secondary Drive 1 Recovery Time (SRCT1)</b> —R/W. 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. 00 = 4 clocks 01 = 3 clocks 10 = 2 clocks 11 = 1 clocks
3:2	<b>Primary Drive 1 IORDY Sample Point (PISP1)</b> —R/W. 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. 00 = 5 clocks 01 = 4 clocks 10 = 3 clocks 11 = Reserved
1:0	<b>Primary Drive 1 Recovery Time (PRCT1)</b> —R/W. 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. 00 = 4 clocks 01 = 3 clocks 10 = 2 clocks 11 = 1 clocks

### 10.1.12 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	<b>Secondary Drive 1 Synchronous DMA Mode Enable (SSDE1)</b> —R/W. 0 = Disable (default). 1 = Enable Synchronous DMA mode for secondary channel drive 1
2	<b>Secondary Drive 0 Synchronous DMA Mode Enable (SSDE0)</b> —R/W. 0 = Disable (default). 1 = Enable Synchronous DMA mode for secondary drive 0.
1	<b>Primary Drive 1 Synchronous DMA Mode Enable (PSDE1)</b> —R/W. 0 = Disable (default). 1 = Enable Synchronous DMA mode for primary channel drive 1
0	<b>Primary Drive 0 Synchronous DMA Mode Enable (PSDE0)</b> —R/W. 0 = Disable (default). 1 = Enable Synchronous DMA mode for primary channel drive 0

### 10.1.13 SDMA\_TIM—Synchronous DMA Timing Register (IDE—D31:F1)

Address Offset: 4A–4Bh  
Default Value: 0000h

Attribute: R/W  
Size: 16 bits

Bit	Description															
15:14	Reserved.															
13:12	<b>Secondary Drive 1 Cycle Time (SCT1)</b> —R/W. For Ultra ATA mode, the setting of these bits determines the minimum write strobe cycle time (CT). The DMARDY#-to-STOP (RP) time is also determined by the setting of these bits. <table><tr><th>SCB1 = 0 (33 MHz clk)</th><th>SCB1 = 1 (66 MHz clk)</th><th>FAST_SCB1 = 1 (133 MHz clk)</th></tr><tr><td>00 = CT 4 clocks, RP 6 clocks</td><td>00 = Reserved</td><td>00 = Reserved</td></tr><tr><td>01 = CT 3 clocks, RP 5 clocks</td><td>01 = CT 3 clocks, RP 8 clocks</td><td>01 = CT 3 clks, RP 16 clks</td></tr><tr><td>10 = CT 2 clocks, RP 4 clocks</td><td>10 = CT 2 clocks, RP 8 clocks</td><td>10 = Reserved</td></tr><tr><td>11 = Reserved</td><td>11 = Reserved</td><td>11 = Reserved</td></tr></table>	SCB1 = 0 (33 MHz clk)	SCB1 = 1 (66 MHz clk)	FAST_SCB1 = 1 (133 MHz clk)	00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved	01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks	10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved	11 = Reserved	11 = Reserved	11 = Reserved
SCB1 = 0 (33 MHz clk)	SCB1 = 1 (66 MHz clk)	FAST_SCB1 = 1 (133 MHz clk)														
00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved														
01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks														
10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved														
11 = Reserved	11 = Reserved	11 = Reserved														
11:10	Reserved.															
9:8	<b>Secondary Drive 0 Cycle Time (SCT0)</b> —R/W. For Ultra ATA mode, the setting of these bits determines the minimum write strobe cycle time (CT). The DMARDY#-to-STOP (RP) time is also determined by the setting of these bits. <table><tr><th>SCB1 = 0 (33 MHz clk)</th><th>SCB1 = 1 (66 MHz clk)</th><th>FAST_SCB1 = 1 (133 MHz clk)</th></tr><tr><td>00 = CT 4 clocks, RP 6 clocks</td><td>00 = Reserved</td><td>00 = Reserved</td></tr><tr><td>01 = CT 3 clocks, RP 5 clocks</td><td>01 = CT 3 clocks, RP 8 clocks</td><td>01 = CT 3 clks, RP 16 clks</td></tr><tr><td>10 = CT 2 clocks, RP 4 clocks</td><td>10 = CT 2 clocks, RP 8 clocks</td><td>10 = Reserved</td></tr><tr><td>11 = Reserved</td><td>11 = Reserved</td><td>11 = Reserved</td></tr></table>	SCB1 = 0 (33 MHz clk)	SCB1 = 1 (66 MHz clk)	FAST_SCB1 = 1 (133 MHz clk)	00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved	01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks	10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved	11 = Reserved	11 = Reserved	11 = Reserved
SCB1 = 0 (33 MHz clk)	SCB1 = 1 (66 MHz clk)	FAST_SCB1 = 1 (133 MHz clk)														
00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved														
01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks														
10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved														
11 = Reserved	11 = Reserved	11 = Reserved														
7:6	Reserved.															

Bit	Description															
5:4	<p><b>Primary Drive 1 Cycle Time (PCT1)</b>—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.</p> <table><tr><th>PCB1 = 0 (33 MHz clk)</th><th>PCB1 = 1 (66 MHz clk)</th><th>FAST_PCB1 = 1 (133 MHz clk)</th></tr><tr><td>00 = CT 4 clocks, RP 6 clocks</td><td>00 = Reserved</td><td>00 = Reserved</td></tr><tr><td>01 = CT 3 clocks, RP 5 clocks</td><td>01 = CT 3 clocks, RP 8 clocks</td><td>01 = CT 3 clks, RP 16 clks</td></tr><tr><td>10 = CT 2 clocks, RP 4 clocks</td><td>10 = CT 2 clocks, RP 8 clocks</td><td>10 = Reserved</td></tr><tr><td>11 = Reserved</td><td>11 = Reserved</td><td>11 = Reserved</td></tr></table>	PCB1 = 0 (33 MHz clk)	PCB1 = 1 (66 MHz clk)	FAST_PCB1 = 1 (133 MHz clk)	00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved	01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks	10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved	11 = Reserved	11 = Reserved	11 = Reserved
PCB1 = 0 (33 MHz clk)	PCB1 = 1 (66 MHz clk)	FAST_PCB1 = 1 (133 MHz clk)														
00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved														
01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks														
10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved														
11 = Reserved	11 = Reserved	11 = Reserved														
3:2	Reserved.															
1:0	<p><b>Primary Drive 0 Cycle Time (PCT0)</b>—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.</p> <table><tr><th>PCB1 = 0 (33 MHz clk)</th><th>PCB1 = 1 (66 MHz clk)</th><th>FAST_PCB1 = 1 (133 MHz clk)</th></tr><tr><td>00 = CT 4 clocks, RP 6 clocks</td><td>00 = Reserved</td><td>00 = Reserved</td></tr><tr><td>01 = CT 3 clocks, RP 5 clocks</td><td>01 = CT 3 clocks, RP 8 clocks</td><td>01 = CT 3 clks, RP 16 clks</td></tr><tr><td>10 = CT 2 clocks, RP 4 clocks</td><td>10 = CT 2 clocks, RP 8 clocks</td><td>10 = Reserved</td></tr><tr><td>11 = Reserved</td><td>11 = Reserved</td><td>11 = Reserved</td></tr></table>	PCB1 = 0 (33 MHz clk)	PCB1 = 1 (66 MHz clk)	FAST_PCB1 = 1 (133 MHz clk)	00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved	01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks	10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved	11 = Reserved	11 = Reserved	11 = Reserved
PCB1 = 0 (33 MHz clk)	PCB1 = 1 (66 MHz clk)	FAST_PCB1 = 1 (133 MHz clk)														
00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved														
01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks														
10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved														
11 = Reserved	11 = Reserved	11 = Reserved														

### 10.1.14 IDE\_CONFIG—IDE I/O Configuration Register

Address Offset: 54h Attribute: R/W  
Default Value: 00h Size: 32 bits

Bit	Description
31:20	Reserved.
19:18	<p><b>Secondary IDE Signal Mode (SEC_SIG_MODE)</b>—R/W. These bits are used to control mode of the Secondary IDE signal pins. These bits should always be set to 00b for desktop implementations.</p> <p>00 = Normal (Enabled). 01 = Tri-state (Disabled). 10 = Drive low (Disabled). 11 = Reserved.</p>
17:16	<p><b>Primary IDE Signal Mode (PRIM_SIG_MODE)</b>—R/W. These bits are used to control mode of the Primary IDE signal pins. These bits should always be set to 00b for desktop implementations.</p> <p>00 = Normal (Enabled). 01 = Tri-state (Disabled). 10 = Drive low (Disabled). 11 = Reserved.</p>
15	<p><b>Fast Secondary Drive 1 Base Clock (FAST_SCB1)</b>—R/W. This bit is used in conjunction with the SCT1 bits to enable/disable Ultra ATA/100 timings for the Secondary Slave drive.</p> <p>0 = Disable Ultra ATA/100 timing for the Secondary Slave drive. 1 = Enable Ultra ATA/100 timing for the Secondary Slave drive (overrides bit 3 in this register).</p>
14	<p><b>Fast Secondary Drive 0 Base Clock (FAST_SCB0)</b>—R/W. This bit is used in conjunction with the SCT0 bits to enable/disable Ultra ATA/100 timings for the Secondary Master drive.</p> <p>0 = Disable Ultra ATA/100 timing for the Secondary Master drive. 1 = Enable Ultra ATA/100 timing for the Secondary Master drive (overrides bit 2 in this register).</p>
13	<p><b>Fast Primary Drive 1 Base Clock (FAST_PCB1)</b>—R/W. This bit is used in conjunction with the PCT1 bits to enable/disable Ultra ATA/100 timings for the Primary Slave drive.</p> <p>0 = Disable Ultra ATA/100 timing for the Primary Slave drive. 1 = Enable Ultra ATA/100 timing for the Primary Slave drive (overrides bit 1 in this register).</p>

Bit	Description
12	<b>Fast Primary Drive 0 Base Clock (FAST_PCB0)</b> —R/W. This bit is used in conjunction with the PCT0 bits to enable/disable Ultra ATA/100 timings for the Primary Master drive. 0 = Disable Ultra ATA/100 timing for the Primary Master drive. 1 = Enable Ultra ATA/100 timing for the Primary Master drive (overrides bit 0 in this register).
11	Reserved.
10	<b>Write Buffer PingPong Enable (WR_PingPong_EN)</b> —R/W. 0 = Disabled. The buffer will behave similar to PIIX4. 1 = Enables the write buffer to be used in a split (ping/pong) manner.
9:8	Reserved.
7	<b>Secondary Slave Channel Cable Reporting</b> —R/W. BIOS should program this bit to tell the IDE driver which cable is plugged into the channel. 0 = 40 conductor cable is present. 1 = 80 conductor cable is present.
6	<b>Secondary Master Channel Cable Reporting</b> —R/W. Same description as bit 7
5	<b>Primary Slave Channel Cable Reporting</b> —R/W. Same description as bit 7
4	<b>Primary Master Channel Cable Reporting</b> —R/W. Same description as bit 7
3	<b>Secondary Drive 1 Base Clock (SCB1)</b> —R/W. 0 = 33 MHz base clock for Ultra ATA timings. 1 = 66 MHz base clock for Ultra ATA timings
2	<b>Secondary Drive 0 Base Clock (SCB0)</b> —R/W. 0 = 33 MHz base clock for Ultra ATA timings. 1 = 66 MHz base clock for Ultra ATA timings
1	<b>Primary Drive 1 Base Clock (PCB1)</b> —R/W. 0 = 33 MHz base clock for Ultra ATA timings. 1 = 66 MHz base clock for Ultra ATA timings
0	<b>Primary Drive 0 Base Clock (PCB0)</b> —R/W. 0 = 33 MHz base clock for Ultra ATA timings. 1 = 66 MHz base clock for Ultra ATA timings

## 10.2 Bus Master IDE I/O Registers (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; writes to reserved bits have no affect (but should not be attempted). The description of the I/O registers is shown in [Table 10-2](#).

**Table 10-2. Bus Master IDE I/O Registers**

Offset	Mnemonic	Register	Default	Type
00h	BMICP	Command Register Primary	00h	R/W
01h		Reserved	00h	RO
02h	BMISP	Status Register Primary	00h	R/WC
03h		Reserved	00h	RO
04h–07h	BMIDP	Descriptor Table Pointer Primary	xxxxxxxh	R/W
08h	BMICS	Command Register Secondary	00h	R/W
09h		Reserved	00h	RO
0Ah	BMISS	Status Register Secondary	00h	R/WC
0Bh		Reserved	00h	RO
0Ch–0Fh	BMIDS	Descriptor Table Pointer Secondary	xxxxxxxh	R/W

### 10.2.1 BMIC[P,S]—Bus Master IDE Command Register

Address Offset:	Primary: 00h Secondary: 08h	Attribute:	R/W
Default Value:	00h	Size:	8 bits

Bit	Description
7:4	Reserved. Returns 0s.
3	<b>Read / Write Control (RWC)</b> —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. 0 = Memory reads. 1 = Memory writes
2:1	Reserved. Returns 0s.
0	<b>Start/Stop Bus Master (START)</b> —R/W. 1 = Enables bus master operation of the controller. Bus master operation begins when this bit is detected changing from a zero to a one. 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. 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.  This bit is intended to be reset after the data transfer is completed, as indicated by either the Bus Master IDE Active bit or the Interrupt bit of the Bus Master IDE Status register for that IDE channel being set, or both.

## 10.2.2 BMIS[P,S]—Bus Master IDE Status Register

Address Offset: Primary: 02h      Attribute: R/W  
Secondary: 0Ah  
Default Value: 00h      Size: 8 bits

Bit	Description
7	Reserved. Returns 0.
6	<b>Drive 1 DMA Capable</b> —R/W. 0 = Not Capable. 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 ICH2 does not use this bit. It is intended for systems that do not attach BMIDE to the PCI bus.
5	<b>Drive 0 DMA Capable</b> —R/W. 0 = Not Capable. 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 ICH2 does not use this bit. It is intended for systems that do not attach BMIDE to the PCI bus.
4:3	Reserved. Returns 0s.
2	<b>Interrupt</b> —R/W. Software can use this bit to determine if an IDE device has asserted its interrupt line (IRQ14 for the Primary channel and IRQ15 for Secondary). 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 one, all data transferred from the drive is visible in system memory. 0 = This bit is cleared by software writing a '1' to the bit position. 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.
1	<b>Error</b> —R/W. 1 = This bit is set when the controller encounters a target abort or master abort when transferring data on PCI. 0 = This bit is cleared by software writing a '1' to the bit position.
0	<b>Bus Master IDE Active (ACT)</b> —RO. 1 = Set by the ICH2 when the Start bit is written to the Command register. 0 = This bit is cleared by the ICH2 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 ICH2 when the Start bit is cleared in the Command register. When this bit is read as a zero, all data transferred from the drive during the previous bus master command is visible in system memory, unless the bus master command was aborted.

## 10.2.3 BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register

Address Offset: Primary: 04h      Attribute: R/W  
Secondary: 0Ch  
Default Value: All bits undefined      Size: 32 bits

Bit	Description
31:2	<b>Base address of Descriptor table (BADDR)</b> —R/W. Corresponds to A[31:2]. The Descriptor Table must be DWord aligned. The Descriptor Table must not cross a 64 KB boundary in memory.
1:0	Reserved.

# USB Controller Registers

# 11

## 11.1 PCI Configuration Registers (D31:F2/F4)

**Note:** Registers that are not shown should be treated as Reserved (See [Section 6.2](#) for details).

**Table 11-1. PCI Configuration Map (USB—D31:F2/F4)**

Offset	Mnemonic	Register Name/Function	Function 2 Default	Function 4 Default	Type
00–01h	VID	Vendor ID	8086h	8086h	RO
02–03h	DID	Device ID	2442h	2444h	RO
04–05h	CMD	Command Register	0000h	0000h	R/W
06–07h	STA	Device Status	0280h	0280h	R/W
08h	RID	Revision ID	See Note	See Note	RO
09h	PI	Programming Interface	00h	00h	RO
0Ah	SCC	Sub Class Code	03h	03h	RO
0Bh	BCC	Base Class Code	0Ch	0Ch	RO
0Eh	HTYPE	Header Type	00h	00h	RO
20–23h	Base	Base Address Register	00000001h	00000001h	R/W
2C–2Dh	SVID	Subsystem Vendor ID	00	00	RO
2E–2Fh	SID	Subsystem ID	00	00	RO
3Ch	INTR_LN	Interrupt Line	00h	00h	R/W
3Dh	INTR_PN	Interrupt Pin	03h	03h	RO
60h	SB_RELNUM	Serial Bus Release Number	10h	10h	RO
C0–C1h	USB_LEGKEY	USB Legacy Keyboard/ Mouse Control	2000h	2000h	R/W
C4h	USB_RES	USB Resume Enable	00h	00h	R/W

**NOTE:** Refer to the Specification Update for the value of the Revision ID Register.

### 11.1.1 VID—Vendor Identification Register (USB—D31:F2/F4)

Address Offset: 00–01h      Attribute: RO  
Default Value: 8086h      Size: 16 bits

Bit	Description
15:0	<b>Vendor ID Value</b> —RO. This is a 16-bit value assigned to Intel.

### 11.1.2 DID—Device Identification Register (USB—D31:F2/F4)

Address Offset: 02–03h      Attribute: RO  
 Default Value: Function 2: 2442h      Size: 16 bits  
 Function 4: 2444h

Bit	Description
15:0	<b>Device ID Value</b> —RO. This is a 16-bit value assigned to the ICH2 USB Host Controllers

### 11.1.3 CMD—Command Register (USB—D31:F2/F4)

Address Offset: 04–05h      Attribute: R/W  
 Default Value: 0000h      Size: 16 bits

Bit	Description
15:10	Reserved.
9	Fast Back to Back Enable (FBE)—RO. Reserved as 0.
8	SERR# Enable—RO. Reserved as 0.
7	Wait Cycle Control—RO. Reserved as 0.
6	Parity Error Response—RO. Reserved as 0.
5	VGA Palette Snoop—RO. Reserved as 0.
4	Postable Memory Write Enable (PMWE)—RO. Reserved as 0.
3	Special Cycle Enable (SCE)—RO. Reserved as 0.
2	<b>Bus Master Enable (BME)</b> —R/W. When set, the ICH2 can act as a master on the PCI bus for USB transfers.
1	Memory Space Enable (MSE)—RO. Reserved as 0.
0	<b>I/O Space Enable (IOSE)</b> —R/W. This bit controls access to the I/O space registers. 1 = Enable accesses to the USB I/O registers. The Base Address register for USB should be programmed before this bit is set. 0 = Disable

### 11.1.4 STA—Device Status Register (USB—D31:F2/F4)

Address Offset: 06–07h Attribute: R/WC  
Default Value: 0280h Size: 16 bits

Bit	Description
15:14	Reserved as '00b'. Read Only.
13	<b>Received Master-Abort Status (RMA)</b> —R/WC. 1 = USB, as a master, generated a master-abort. 0 = Software clears this bit by writing a 1 to the bit location.
12	Reserved. Always read as 0.
11	<b>Signaled Target-Abort Status (STA)</b> —R/WC. 1 = USB function is targeted with a transaction that the ICH2 terminates with a target abort. 0 = Software clears this bit by writing a 1 to the bit location.
10:9	<b>DEVSEL# Timing Status (DEVT)</b> —RO. This 2-bit field defines the timing for DEVSEL# assertion. These read only bits indicate the ICH2's DEVSEL# timing when performing a positive decode. ICH2 generates DEVSEL# with medium timing for USB.
8	Data Parity Error Detected: Reserved as 0. Read Only.
7	Fast Back-to-Back Capable: Reserved as 1. Read Only.
6	User Definable Features (UDF): Reserved as 0. Read Only.
5	66 MHz Capable: Reserved as 0. Read Only.
4:0	Reserved.

### 11.1.5 RID—Revision Identification Register (USB—D31:F2/F4)

Address Offset: 08h Attribute: RO  
Default Value: See bit description Size: 8 bits

Bit	Description
7:0	<b>Revision Identification.</b> These bits contain device stepping information and are hardwired to the default value. Refer to the Specification Update for the value of the Revision ID Register

### 11.1.6 PI—Programming Interface (USB—D31:F2/F4)

Address Offset: 09h Attribute: RO  
Default Value: 00h Size: 8 bits

Bit	Description
7:0	Programming Interface—RO. 00h = No specific register level programming interface defined.

### 11.1.7 SCC—Sub Class Code Register (USB—D31:F2/F4)

Address Offset: 0Ah      Attribute: RO  
 Default Value: 03h      Size: 8 bits

Bit	Description
7:0	<b>Sub Class Code</b> —RO. 03h = Universal Serial Bus Host Controller.

### 11.1.8 BCC—Base Class Code Register (USB—D31:F2/F4)

Address Offset: 0Bh      Attribute: RO  
 Default Value: 0Ch      Size: 8 bits

Bit	Description
7:0	<b>Base Class Code</b> —RO. 0Ch = Serial Bus controller.

### 11.1.9 BASE—Base Address Register (USB—D31:F2/F4)

Address Offset: 20–23h      Attribute: R/W  
 Default Value: 00000001h      Size: 32 bits

Bit	Description
31:16	Reserved.
15:5	<b>Base Address</b> —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	<b>Resource Type Indicator (RTE)</b> —RO. This bit is hardwired to 1 indicating that the base address field in this register maps to I/O space

### 11.1.10 SVID—Subsystem Vendor ID (USB—D31:F2/F4)

Address Offset: 2Ch–2Dh      Attribute: RO  
 Default Value: 00h      Size: 16 bits  
 Lockable: No      Power Well: Core

Bit	Description
15:0	<b>Subsystem Vendor ID (SVID)</b> —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.

### 11.1.11 SID—Subsystem ID (USB—D31:F2/F4)

Address Offset: 2Eh–2Fh Attribute: RO  
 Default Value: 00h Size: 16 bits  
 Lockable: No Power Well: Core

Bit	Description
15:0	<b>Subsystem ID (SID)</b> —R/Write-Once. 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.

### 11.1.12 INTR\_LN—Interrupt Line Register (USB—D31:F2/F4)

Address Offset: 3Ch Attribute: R/W  
 Default Value: 00h Size: 8 bits

Bit	Description
7:0	<b>Interrupt Line</b> —R/W. This data is not used by the ICH2. It is to communicate to software the interrupt line that the interrupt pin is connected to.

### 11.1.13 INTR\_PN—Interrupt Pin Register (USB—D31:F2/F4)

Address Offset: 3Dh Attribute: RO  
 Default Value: Function 2: 03h Size: 8 bits  
 Function 4: 03h

Bit	Description
7:3	Reserved.
2:0	<b>Interrupt Pin</b> —RO. The value of 03h in Function 2 indicates that the ICH2 will drive PIRQD# as its interrupt line for USB Controller 0 (ports 0 and 1). The value of 03h in Function 4 indicates that the ICH2 will drive PIRQC# as its interrupt line for USB Controller 1 (ports 2 and 3). However, in the ICH2 implementation, when the USB Controller 1 interrupt is generated PIRQ[H]# will go active, not PIRQ[C]#.

### 11.1.14 SB\_RELNUM—Serial Bus Release Number Register (USB—D31:F2/F4)

Address Offset: 60h Attribute: RO  
 Default Value: 10h Size: 8 bits

Bit	Description
7:0	<b>Serial Bus Release Number</b> —RO. 10h = Indicates that the USB controller is compliant with the USB specification release 1.0.

### 11.1.15 USB\_LEGKEY—USB Legacy Keyboard/Mouse Control Register (USB—D31:F2/F4)

Address Offset: C0–C1      Attribute: R/W, R/WC, RO  
 Default Value: 2000h      Size: 16 bits

Bit	Description
15	<b>SMI Caused by End of Pass-through (SMIBYENDPS)</b> —R/WC. Indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 0, 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#. 1 = Event Occurred 0 = Software clears this bit by writing a 1 to the bit location.
14	Reserved.
13	<b>PCI Interrupt Enable (USBPIRQEN)</b> —R/W. Used to prevent the USB controller from generating an interrupt due to transactions on its ports. Note that it will probably be configured to generate an SMI using bit 4 of this register. Default to 1 for compatibility with older USB software. 1 = Enable 0 = Disable
12	<b>SMI Caused by USB Interrupt (SMIBYUSB)</b> —RO. Indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 4, 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#. 1 = Event Occurred 0 = Software should clear the IRQ via the USB controller. Writing a 1 to this bit will have no effect.
11	<b>SMI Caused by Port 64 Write (TRAPBY64W)</b> —R/WC. 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#. 1 = Event Occurred 0 = Software clears this bit by writing a 1 to the bit location.
10	<b>SMI Caused by Port 64 Read (TRAPBY64R)</b> —R/WC. 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#. 1 = Event Occurred 0 = Software clears this bit by writing a 1 to the bit location.
9	<b>SMI Caused by Port 60 Write (TRAPBY60W)</b> —R/WC. 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#. 1 = Event Occurred 0 = Software clears this bit by writing a 1 to the bit location.
8	<b>SMI Caused by Port 60 Read (TRAPBY60R)</b> —R/WC. Indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 0, 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#. 1 = Event Occurred 0 = Software clears this bit by writing a 1 to the bit location.
7	<b>SMI at End of Pass-through Enable (SMIATENDPS)</b> —R/W. May need to cause SMI at the end of a pass-through. Can occur if an SMI is generated in the middle of a pass through, and needs to be serviced later. 1 = Enable 0 = Disable
6	<b>Pass Through State (PSTATE)</b> —RO. 1 = Indicates that the state machine is in the middle of an A20GATE pass-through sequence. 0 = If software needs to reset this bit, it should set bit 5 to 0.
5	<b>A20Gate Pass-Through Enable (A20PASSEN)</b> —R/W. 1 = Allows A20GATE sequence Pass-Through function. SMI# will not be generated, even if the various enable bits are set. 0 = Disable

Bit	Description
4	<b>SMI on USB IRQ Enable (USBSMIEN)</b> —R/W. 1 = USB interrupt will cause an SMI event. 0 = Disable
3	<b>SMI on Port 64 Writes Enable (64WEN)</b> —R/W. 1 = A write to port 64h will cause an SMI event. 0 = Disable
2	<b>SMI on Port 64 Reads Enable (64REN)</b> —R/W. 1 = A read to port 64h will cause an SMI event. 0 = Disable
1	<b>SMI on Port 60 Writes Enable (60WEN)</b> —R/W. 1 = A write to port 60h will cause an SMI event. 0 = Disable
0	<b>SMI on Port 60 Reads Enable (60REN)</b> —R/W. 1 = A read to port 60h will cause an SMI event. 0 = Disable

### 11.1.16 USB\_RES—USB Resume Enable Register (USB—D31:F2/F4)

Address Offset: C4h      Attribute: R/W  
Default Value: 00h      Size: 8 bits

Bit	Description
7:2	Reserved.
1	<b>PORT1EN</b> —R/W. Enable the USB controller to respond to wakeup events on this port. For Function 2 this applies to port 1; for Function 4, this applies to port 3. 1 = The USB controller will monitor this port for remote wakeup and connect/disconnect events. 0 = The USB controller will not look at this port for a wakeup event.
0	<b>PORT0EN</b> —R/W. Enable the USB controller to respond to wakeup events on this port. For Function 2 this applies to port 0; for Function 4, this applies to port 2. 1 = The USB controller will monitor this port for remote wakeup and connect/disconnect events. 0 = The USB controller will not look at this port for a wakeup event.

## 11.2 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 11-2. USB I/O Registers**

Offset	Mnemonic	Register	Default	Type
00–01h	USBCMD	USB Command Register	0000h	R/W*
02–03h	USBSTS	USB Status Register	0020h	R/WC
04–05h	USBINTR	USB Interrupt Enable	0000h	R/W
06–07h	FRNUM	USB Frame Number	0000h	R/W (see Note 1)
08–0Bh	FRBASEADD	USB Frame List Base Address	Undefined	R/W
0Ch	SOFMOD	USB Start of Frame Modify	40h	R/W
0D–0Fh	—	Reserved	0	RO
10–11h	PORTSC0	Port 0 Status/Control	0080h	R/WC (see Note 1)
12–13h	PORTSC1	Port 1 Status/Control	0080h	R/WC (see Note 1)
14–17h	—	Reserved	0	RO
18h	LOOPDATA	Loop Back Test Data	00h	RO

**NOTES:**

1. These registers are Word writable only. Byte writes to these registers have unpredictable effects.

### 11.2.1 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	<b>Loop Back Test Mode—R/W.</b> 1 = ICH2 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. 0 = Disable loop back test mode.

Bit	Description
7	<b>Max Packet (MAXP)</b> —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. 1 = 64 bytes 0 = 32 bytes
6	<b>Configure Flag (CF)</b> —R/W. This bit has no effect on the hardware. It is provided only as a semaphore service for software. 1 = HCD software sets this bit as the last action in its process of configuring the Host Controller. 0 = Indicates that software has not completed host controller configuration.
5	<b>Software Debug (SWDBG)</b> —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. 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. 0 = Normal Mode.
4	<b>Force Global Resume (FGR)</b> —R/W. 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. 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.
3	<b>Enter Global Suspend Mode (EGSM)</b> —R/W. 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. 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.
2	<b>Global Reset (GRESET)</b> —R/W. 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. 0 = This bit is reset by the software after a minimum of 10 ms has elapsed as specified in Chapter 7 of the USB Specification.
1	<b>Host Controller Reset (HCRESET)</b> —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. 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 = Reset by the Host Controller when the reset process is complete.

Bit	Description
0	<b>Run/Stop (RS)</b> —R/W. When set to 1, the ICH2 proceeds with execution of the schedule. The ICH2 continues execution as long as this bit is set. When this bit is cleared, the ICH2 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. 1 = Run 0 = Stop

**Table 11-3. 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.

## 11.2.2 USBSTA—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. Software sets a bit to 0 in this register by writing a 1 to it.

Bit	Description
15:6	Reserved.
5	<b>HCHalted</b> —R/WC. 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. 0 = Software resets this bit to 0 by writing a 1 to the bit position.
4	<b>Host Controller Process Error</b> —R/WC. 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. 0 = Software resets this bit to 0 by writing a 1 to the bit position.
3	<b>Host System Error</b> —R/WC. 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. 0 = Software resets this bit to 0 by writing a 1 to the bit position.
2	<b>Resume Detect (RSM_DET)</b> —R/WC. 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). 0 = Software resets this bit to 0 by writing a 1 to the bit position.
1	<b>USB Error Interrupt</b> —R/WC. 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 = Software resets this bit to 0 by writing a 1 to the bit position.
0	<b>USB Interrupt (USBINT)</b> —R/WC. 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. 0 = Software resets this bit to 0 by writing a 1 to the bit position.

### 11.2.3 USBINTR—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	<b>Short Packet Interrupt Enable</b> —R/W. 1 = Enabled. 0 = Disabled.
2	<b>Interrupt On Complete (IOC) Enable</b> —R/W. 1 = Enabled. 0 = Disabled.
1	<b>Resume Interrupt Enable</b> —R/W. 1 = Enabled. 0 = Disabled.
0	<b>Time-out/CRC Interrupt Enable</b> —R/W. 1 = Enabled. 0 = Disabled.

### 11.2.4 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 which 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	<b>Frame List Current Index/Frame Number</b> —R/W. 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].

## 11.2.5 FRBASEADD—Frame List Base Address

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 zero (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	<b>Base Address</b> —R/W. These bits correspond to memory address signals [31:12], respectively.
11:0	Reserved.

## 11.2.6 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 7 least significant bits are used. When a new value is written into these 7 bits, the SOF timing of the next frame will be adjusted. This feature can be used to adjust out any offset from the clock source that generates the clock that drives the SOF counter. This register can also be used to maintain real time synchronization with the rest of the system so that all devices have the same sense of real time. Using this register, the frame length can be adjusted across the full range required by the USB specification. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. It may be reprogrammed by USB system software at any time. Its value will take effect from the beginning of the next frame. This register is reset upon a Host Controller Reset or Global Reset. Software must maintain a copy of its value for reprogramming if necessary.

Bit	Description																								
7	Reserved.																								
6:0	<p><b>SOF Timing Value</b>—R/W. Guidelines for the modification of frame time are contained in Chapter 7 of the USB Specification. The SOF cycle time (number of SOF counter clock periods to generate a SOF frame length) is equal to 11936 + value in this field. The default value is decimal 64 which gives a SOF cycle time of 12000. For a 12 MHz SOF counter clock input, this produces a 1 ms Frame period. The following table indicates what SOF Timing Value to program into this field for a certain frame period.</p> <table> <tr> <th>Frame Length (# 12 MHz Clocks) (decimal)</th><th>SOF Reg. Value (decimal)</th></tr> <tr> <td>11936</td><td>0</td></tr> <tr> <td>11937</td><td>1</td></tr> <tr> <td>.</td><td>.</td></tr> <tr> <td>.</td><td>.</td></tr> <tr> <td>11999</td><td>63</td></tr> <tr> <td>12000</td><td>64</td></tr> <tr> <td>12001</td><td>65</td></tr> <tr> <td>.</td><td>.</td></tr> <tr> <td>.</td><td>.</td></tr> <tr> <td>12062</td><td>126</td></tr> <tr> <td>12063</td><td>127</td></tr> </table>	Frame Length (# 12 MHz Clocks) (decimal)	SOF Reg. Value (decimal)	11936	0	11937	1	.	.	.	.	11999	63	12000	64	12001	65	.	.	.	.	12062	126	12063	127
Frame Length (# 12 MHz Clocks) (decimal)	SOF Reg. Value (decimal)																								
11936	0																								
11937	1																								
.	.																								
.	.																								
11999	63																								
12000	64																								
12001	65																								
.	.																								
.	.																								
12062	126																								
12063	127																								

## 11.2.7 PORTSC[0,1]—Port Status and Control Register

I/O Offset:	Port 0/2: Base + (10–11h) Port 1/3: Base + (12–13h)	Attribute:	R/W (Word writes only)
Default Value:	0080h	Size:	16 bits

**Note:** For Function 2, this applies to ICH2 USB ports 0 and 1. For Function 4, this applies to ICH2 USB ports 2 and 3.

After a Power-up reset, Global reset, or Host Controller reset, the initial conditions of a port are: no device connected, Port disabled, and the bus line status is 00 (single-ended zero).

Bit	Description								
15:13	Reserved—RO.								
12	<p><b>Suspend</b>—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:</p> <table> <tr> <th>Bits [12,2]</th><th>Hub State</th></tr> <tr> <td>X0</td><td>Disable</td></tr> <tr> <td>01</td><td>Enable</td></tr> <tr> <td>11</td><td>Suspend</td></tr> </table> <p>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.</p> <p>1 = Port in suspend state. 0 = Port not in suspend state.</p> <p>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 ICH2 may issue a start-of-frame, and then suspend the port.</p>	Bits [12,2]	Hub State	X0	Disable	01	Enable	11	Suspend
Bits [12,2]	Hub State								
X0	Disable								
01	Enable								
11	Suspend								
11	<p><b>Overcurrent Indicator</b>—R/WC. Set by hardware</p> <p>1 = Overcurrent pin has gone from inactive to active on this port. 0 = Software clears this bit by writing a 1 to the bit position.</p>								
10	<p><b>Overcurrent Active</b>—RO. This bit is set and cleared by hardware.</p> <p>1 = Indicates that the overcurrent pin is active (low). 0 = Indicates that the overcurrent pin is inactive (high).</p>								
9	<p><b>Port Reset</b>—RO.</p> <p>1 = Port is in Reset. When set, the port is disabled and sends the USB Reset signaling. 0 = Port is not in Reset.</p>								
8	<p><b>Low Speed Device Attached (LS)</b>—RO. Writes have no effect.</p> <p>1 = Low speed device is attached to this port. 0 = Full speed device is attached.</p>								
7	Reserved—RO. Always read as 1.								
6	<p><b>Resume Detect (RSM_DET)</b>—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 ICH2 then reflects 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.</p> <p>1 = Resume detected/driven on port. 0 = No resume (K-state) detected/driven on port.</p>								
5:4	<p><b>Line Status</b>—RO. These bits reflect the D+ (bit 4) and D- (bit 5) signals lines' logical levels. These bits are used for fault detect and recovery as well as for USB diagnostics. This field is updated at EOF2 time (See Chapter 11 of the USB Specification).</p>								

Bit	Description
3	<p><b>Port Enable/Disable Change</b>—R/WC. For the root hub, this bit gets set only when a port is disabled due to disconnect on that port or due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification).</p> <p>1 = Port enabled/disabled status has changed.</p> <p>0 = No change. Software clears this bit by writing a 1 to the bit location.</p>
2	<p><b>Port Enabled/Disabled (PORT_EN)</b>—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.</p> <p>1 = Enable.</p> <p>0 = Disable.</p>
1	<p><b>Connect Status Change</b>—R/WC. 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.</p> <p>1 = Change in Current Connect Status.</p> <p>0 = No change. Software clears this bit by writing a 1 to the bit location.</p>
0	<p><b>Current Connect Status</b>—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.</p> <p>1 = Device is present on port.</p> <p>0 = No device is present.</p>

This page is intentionally left blank.

# SMBus Controller Registers (D31:F3) 12

## 12.1 PCI Configuration Registers (SMBUS—D31:F3)

Table 12-1. PCI Configuration Registers (SMBUS—D31:F3)

Offset	Mnemonic	Register Name/Function
00–01h	VID	Vendor ID
02–03h	DID	Device ID
04–05h	CMD	Command Register
06–07h	STA	Device Status
08h	RID	Revision ID
09h	PI	Programming Interface
0Ah	SCC	Sub Class Code
0Bh	BCC	Base Class Code
20–23h	SMB_BASE	SMBus Base Address Register
2C–2Dh	SVID	Subsystem Vendor ID
2E–2Fh	SID	Subsystem ID
3Ch	INTR_LN	Interrupt Line
3Dh	INTR_PN	Interrupt Pin
40h	HOSTC	Host Configuration

**Note:** Registers that are not shown should be treated as Reserved (See [Section 6.2](#) for details).

### 12.1.1 VID—Vendor Identification Register (SMBUS—D31:F3)

Address: 00–01h Attributes: RO  
Default Value: 8086h Size: 16 bits

Bit	Description
15:0	<b>Vendor ID Value</b> —RO. This is a 16 bit value assigned to Intel

### 12.1.2 DID—Device Identification Register (SMBUS—D31:F3)

Address: 02–03h Attributes: RO  
Default Value: 2443h Size: 16 bits

Bit	Description
15:0	<b>Device ID value</b> —RO.

### 12.1.3 CMD—Command Register (SMBUS—D31:F3)

Address: 04–05h Attributes: RO, R/W  
 Default Value: 0000h Size: 16 bits

Bit	Description
15:10	Reserved.
9	Fast Back to Back Enable (FBE)—RO. Reserved as 0.
8	SERR# Enable (SERREN)—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)—RO. Reserved as 0.
1	Memory Space Enable (MSE)—RO. Reserved as 0.
0	<b>I/O Space Enable (IOSE)—R/W.</b> 0 = Disable. 1 = Enables access to the SM Bus I/O space registers as defined by the Base Address Register.

### 12.1.4 STA—Device Status Register (SMBUS—D31:F3)

Address: 06–07h Attributes: RO, R/W  
 Default Value: 0280h Size: 16 bits

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)—RO. Reserved as 0.
12	Received Target Abort (RTA)—RO. Reserved as 0.
11	<b>Signaled Target-Abort Status (STA)—R/W.</b> 1 = Function is targeted with a transaction that the ICH2 terminates with a target abort. 0 = Software resets STA to 0 by writing a 1 to this bit location.
10:9	<b>DEVSEL# Timing Status (DEVT)—RO.</b> This 2-bit field defines the timing for DEVSEL# assertion for positive decode. 01 = Medium timing.
8	Data Parity Error Detected—RO. Reserved as 0.
7	Fast Back-to-Back Capable—RO. Reserved as 1.
6	User Definable Features (UDF)—RO. Reserved as 0.
5	66 MHz Capable—RO. Reserved as 0.
4:0	Reserved.

### 12.1.5 SCC—Sub Class Code Register (SMBUS—D31:F3)

Address Offset: 0Ah      Attributes: RO  
Default Value: 05h      Size: 8 bits

Bit	Description
7:0	<b>Sub Class Code</b> —RO. 05h = SM Bus serial controller

### 12.1.6 BCC—Base Class Code Register (SMBUS—D31:F3)

Address Offset: 0Bh      Attributes: RO  
Default Value: 0Ch      Size: 8 bits

Bit	Description
7:0	<b>Base Class Code</b> —RO. 0Ch = Serial controller.

### 12.1.7 SMB\_BASE—SMBus Base Address Register (SMBUS—D31:F3)

Address Offset: 20–23h      Attribute: R/W  
Default Value: 00000001h      Size: 32-bits

Bit	Description
31:16	Reserved.
15:4	<b>Base Address</b> —R/W. Provides the 16-bit system I/O base address for the ICH2 SMB logic.
3:1	Reserved.
0	<b>IO Space Indicator</b> —RO. This read-only bit is always 1, indicating that the SMB logic is I/O mapped.

### 12.1.8 SVID—Subsystem Vendor ID (SMBUS—D31:F2/F4)

Address Offset: 2Ch–2Dh      Attribute: RO  
Default Value: 00h      Size: 16 bits  
Lockable: No      Power Well: Core

Bit	Description
15:0	<b>Subsystem Vendor ID (SVID)</b> —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.

### 12.1.9 SID—Subsystem ID (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	<b>Subsystem ID (SID)</b> —R/Write-Once. 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.

### 12.1.10 INTR\_LN—Interrupt Line Register (SMBUS—D31:F3)

Address Offset: 3Ch      Attributes: R/W  
 Default Value: 00h      Size: 8 bits

Bit	Description
7:0	<b>Interrupt line</b> —R/W. This data is not used by the ICH2. It is to communicate to software the interrupt line that the interrupt pin is connected to PIRQB#.

### 12.1.11 INTR\_PN—Interrupt Pin Register (SMBUS—D31:F3)

Address Offset: 3Dh      Attributes: RO  
 Default Value: 02h      Size: 8 bits

Bit	Description
7:0	<b>Interrupt PIN</b> —RO. 02h = Indicates that the ICH2 SMBus Controller will drive PIRQB# as its interrupt line.

### 12.1.12 HOSTC—Host Configuration Register (SMBUS—D31:F3)

Address Offset: 40h      Attribute: R/W  
 Default Value: 00h      Size: 8 bits

Bit	Description
7:3	Reserved.
2	<b>I<sup>2</sup>C Enable (I2C_EN)</b> —R/W. 0 = SMBus behavior. 1 = The ICH2 is enabled to communicate with I <sup>2</sup> C devices. This will change the formatting of some commands.
1	<b>SMBus to SMI Enable (SMB_SMI_EN)</b> —R/W. 0 = SMBus interrupts will not generate an SMI#. 1 = Any source of an SMB interrupt will instead be routed to generate an SMI#. This bit will only take effect if the INTREN bit is set in I/O space. This bit needs to be set for SMBALERT# to be enabled.
0	<b>SMBus Host Enable (HST_EN)</b> —R/W. 0 = Disable the SMBus Host Controller. 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 serviced.

## 12.2 SMBus I/O Registers

**Table 12-2. SMB I/O Registers**

Offset	Mnemonic	Register Name/Function	Default	Access
00h	HST_STS	Host Status	00h	R/W
02h	HST_CNT	Host Control	00h	R/W
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	BLOCK_DB	Block Data Byte	00h	R/W
08h	—	Reserved	00h	RO
09h	RCV_SLVA	Receive Slave Address	12h	R/W
0Ah	SLV_DATA	Slave Data	0000h	R/W
0Bh–0Dh	—	Reserved	00h	RO
0Eh	SMLINK_PIN_CTL	SMLINK Pin Control	See Register Description	R/W
0Fh	SMBUS_PIN_CTL	SMBus Pin Control	See Register Description	R/W

## 12.2.1 HST\_STS—Host Status Register

Register Offset: 00h      Attribute: R/WC  
 Default Value: 00h      Size: 8-bits

All status bits are set by hardware and cleared by the software writing a one to the particular bit position. Writing a zero to any bit position has no effect.

Bit	Description
7	<b>Byte Done Status (BYTE_DONE_STA)—R/WC.</b> 1 = The ICH2 has received a byte (for Block Read commands) or if it has completed transmission of a byte (for Block Write commands). This bit will be set even on the last byte of the transfer. It will not be set when transmission is due to the Alert On LAN* heartbeat. 0 = Cleared by writing a 1 to the bit position.
6	<b>In Use Status (INUSE_STA)—R/WC (special).</b> This bit is used as semaphore among various independent software threads that may need to use the ICH2's SMBus logic and has no other effect on Hardware. 0 = After a full PCI reset, a read to this bit returns a 0. 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	<b>SMBus Alert Status (SMBALERT_STA)—R/WC.</b> 0 = Interrupt or SMI# was not generated by SMBALERT#. 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. If the signal is programmed as a GPIO, then this bit will never be set.
4	<b>Interrupt/SMI# was Failed Bus Transaction (FAILED)—R/WC.</b> 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. 0 = Cleared by writing a 1 to the bit position.
3	<b>Bus Error (BUS_ERR)—R/WC.</b> 1 = The source of the interrupt of SMI# was a transaction collision. 0 = Cleared by writing a 1 to the bit position.
2	<b>Device Error (DEV_ERR)—R/WC.</b> 1 = The source of the interrupt or SMI# was due to one of the following: <ul style="list-style-type: none"> <li>• Illegal Command Field,</li> <li>• Unclaimed Cycle (host initiated),</li> <li>• Host Device Time-out Error.]</li> </ul> 0 = Software resets this bit by writing a 1 to this location. The ICH2 will then deassert the interrupt or SMI#.
1	<b>Interrupt/SMI# was Successful Completion (INTR)—R/WC (special).</b> 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. 1 = The source of the interrupt or SMI# was the successful completion of its last command. 0 = Software resets this bit by writing 1 to this location. The ICH2 will then deassert the interrupt or SMI#.
0	<b>Host Busy (HOST_BUSY)—RO.</b> 1 = Indicates that the ICH2 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 <sup>2</sup> C Read command. This is necessary in order to check the BYTE_DONE_STS bit. 0 = Cleared by the ICH2 when the current transaction is completed.

## 12.2.2 HST\_CNT—Host Control Register

Register Offset: 02h  
Default Value: 00h

Attribute: R/W  
Size: 8-bits

Bit	Description
7	Reserved.
6	<b>START</b> —WO. 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. 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 ICH2 has finished the command.
5	<b>LAST BYTE</b> —WO. This bit is used for Block Read commands. 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 ICH2 to send a NACK (instead of an ACK) after receiving the last byte.
4:2	<b>SMBus Command (SMB_CMD)</b> —R/W. The bit encoding below indicates which command the ICH2 is to perform. If enabled, the ICH2 generates an interrupt or SMI# when the command has completed. If the value is for a non-supported or reserved command, the ICH2 sets the device error (DEV_ERR) status bit and generates an interrupt when the START bit is set. The ICH2 performs no command and does not operate until DEV_ERR is cleared. 000 = Quick: The slave address and read/write value (bit 0) are stored in the transmit slave address register. 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. 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. 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. 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. 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. 110 = I <sup>2</sup> C 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 ICH2 will continue reading data until the NAK is received. 111 = Reserved
1	<b>KILL</b> —R/W. 1 = When set, 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 = Normal SMBus Host Controller functionality.
0	<b>INTREN</b> —R/W. 1 = Enable the generation of an interrupt or SMI# upon the completion of the command. 0 = Disable.

### 12.2.3 HST\_CMD—Host Command Register

Register Offset: 03h      Attribute: R/W  
Default Value: 00h      Size: 8 bits

Bit	Description
7:0	<b>Host Command</b> —R/W. This eight bit field is transmitted by the host controller in the command field of the SMBus protocol during the execution of any command.

### 12.2.4 XMIT\_SLVA—Transmit Slave Address Register

Register Offset: 04h      Attribute: R/W  
Default Value: 00h      Size: 8 bits

This register is transmitted by the host controller in the slave address field of the SMBus protocol.

Bit	Description
7:1	<b>ADDRESS</b> —R/W. 7-bit address of the targeted slave.
0	<b>Read/Write Select</b> —R/W. Direction of the host transfer. 0 = Write 1 = Read

### 12.2.5 HST\_D0—Data 0 Register

Register Offset: 05h      Attribute: R/W  
Default Value: 00h      Size: 8 bits

Bit	Description
7:0	<b>DATA0/COUNT</b> —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.

### 12.2.6 HST\_D1—Data 1 Register

Register Offset: 06h      Attribute: R/W  
Default Value: 00h      Size: 8 bits

Bit	Description
7:0	<b>DATA1</b> —R/W. This eight bit register is transmitted in the DATA1 field of the SMBus protocol during the execution of any command.

## 12.2.7 BLOCK\_DB—Block Data Byte Register

Register Offset: 07h Attribute: R/W  
Default Value: 00h Size: 8 bits

Bit	Description
7:0	<p><b>Block Data Byte</b>—R/W. For Block Writes, software writes the first byte to this register as part of the setup for this command. After the ICH2 has sent the Address, Command, and Byte Count fields, it will send the byte in the Block Data Byte register. After the byte has been sent, the ICH2 sets the BYTE_DONE_STS bit in the Host Status register. If there are more bytes to send, the software then writes in the next byte to the Block Data Byte register and software also clears the BYTE_DONE_STS bit. The ICH2 then sends the next byte. During the time from when a byte has been transmitted to when the next byte has been loaded, the ICH2 inserts wait-states on the SMBus/I<sup>2</sup>C.</p> <p>A similar process will be used for Block Reads. After receiving the byte count (which goes in the DATA 0 register), the first "data byte" goes in the Block Data Byte register and the ICH2 generates an SMI# or interrupt (depending on configuration). The interrupt or SMI# handler then reads the byte and clears the BYTE_DONE_STS bit. This frees room for the next byte. During the time from when a byte is read to when the BYTE_DONE_STS bit is cleared, the ICH2 inserts wait-states on the SMBus/I<sup>2</sup>C.</p>

## 12.2.8 RCV\_SLVA—Receive Slave Address Register

Register Offset: 09h Attribute: R/W  
Default Value: 44h Size: 8 bits  
Lockable: No Power Well: Resume

Bit	Description
7	Reserved
6:0	<p><b>SLAVE_ADDR</b>—R/W. This field is the slave address that the ICH2 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#.</p>

## 12.2.9 SLV\_DATA—Receive Slave Data Register

Register Offset: 0Ah Attribute: RO  
Default Value: 00h Size: 16 bits  
Lockable: No Power Well: Resume

This register contains the 16-bit data value written by the external SMBus master. The CPU can then read the value from this register. This register is reset by RSMRST#, but not PCIRST#.

Bit	Description
15:8	<b>DATA_MSG1</b> : Data Message Byte 1—RO. See <a href="#">Section 5.17.5</a> for a discussion of this field.
7:0	<b>DATA_MSG0</b> : Data Message Byte 0—RO. See <a href="#">Section 5.17.5</a> for a discussion of this field.

## 12.2.10 SMLINK\_PIN\_CTL—SMLINK Pin Control Register

Register Offset: 0Eh      Attribute: Read/Write  
 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	<b>SMLINK Clock Pin Control (SMLINK_CLK_CTL)</b> —R/W. 1 = No functional impact on the SMLINK[0] pin. (default) 0 = ICH2 will drive the SMLINK[0] pin low, independent of the what the other SMLINK logic would otherwise indicate for the SMLINK[0] pin.
1	<b>SMLINK[1] Pin Current Status (SMLINK[1]_CUR_STA)</b> —RO. This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMLINK[1] pin. This allows software to read the current state of the pin. 1 = SMLINK[1] pin is high 0 = SMLINK[1] pin is low
0	<b>SMLINK[0] Pin Current Status (SMLINK[0]_CUR_STA)</b> —RO. This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMLINK[0] pin. This allows software to read the current state of the pin. 1 = SMLINK[0] pin is high 0 = SMLINK[0] pin is low

## 12.2.11 SMBUS\_PIN\_CTL SMBus Pin Control Register

Register Offset: 0Fh      Attribute: Read/Write  
 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	<b>SMBCLK Pin Control (SMBCLK_CTL)</b> —R/W. 1 = No functional impact on the SMBCLK pin. (default) 0 = ICH2 will drive the SMBCLK pin low, independent of the what the other SMB logic would otherwise indicate for the SMBCLK pin.
1	<b>SMBDATA Pin Current Status (SMBDATA_CUR_STA)</b> —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. 1 = SMBDATA pin is high 0 = SMBDATA pin is low
0	<b>SMBCLK Pin Current Status (SMBCLK_CUR_STA)</b> —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. 1 = SMBCLK pin is high 0 = SMBCLK pin is low

# AC'97 Audio Controller Registers (D31:F5)

13

## 13.1 AC'97 Audio PCI Configuration Space (D31:F5)

**Note:** Registers that are not shown should be treated as Reserved (See [Section 6.2](#) for details).

**Table 13-1. PCI Configuration Map (Audio—D31:F5)**

Offset	Mnemonic	Register	Default	Access
00h–01h	VID	Vendor Identification	8086h	RO
02h–03h	DID	Device Identification	2445h	RO
04h–05h	PCICMD	PCI Command	0000	R/W
06h–07h	PCISTS	PCI Device Status	0280h	R/WC
08h	RID	Revision Identification	See Note	RO
09h	PI	Programming Interface	00	RO
0Ah	SCC	Sub Class Code	01h	RO
0Bh	BCC	Base Class Code	04h	RO
0Eh	HEDT	Header Type	00	RO
10h–13h	NAMBAR	Native Audio Mixer Base Address	00000001h	R/W
14h–17h	NABMBAR	Native Audio Bus Mastering Base Address	00000001h	R/W
18h–2Bh	—	Reserved	00h	RO
2Ch–2Dh	SVID	Subsystem Vendor ID	0000h	Write-Once
2Eh–2Fh	SID	Subsystem ID	0000h	Write-Once
30h–3Bh	—	Reserved	—	—
3Ch	INTR_LN	Interrupt Line	00h	R/W
3Dh	INTR_PN	Interrupt Pin	02h	RO
3Eh–FFh	—	Reserved	—	—

**NOTE:** Refer to the Specification Update for the value of the Revision ID Register

### 13.1.1 VID—Vendor Identification Register (Audio—D31:F5)

Offset: 01h-00h Attribute: RO  
Default Value: 8086h Size: 16 Bits  
Lockable: No Power Well: Core

Bit	Description
15:0	<b>Vendor ID Value.</b> This is a 16 bit value assigned to Intel

### 13.1.2 DID—Device Identification Register (Audio—D31:F5)

Offset:	03h–02h	Attribute:	RO
Default Value:	2445h	Size:	16 Bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	<b>Device ID Value.</b>

### 13.1.3 PCICMD—PCI Command Register (Audio—D31:F5)

Address Offset:	05h–04h	Attribute:	R/W
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

PCICMD is a 16-bit control register. Refer to the PCI 2.1 specification for complete details on each bit.

Bit	Description
15:10	Reserved. Read as 0s.
9	Fast Back to Back Enable (FBE). Not implemented. Hardwired to 0.
8	SERR# Enable (SEN). Not implemented. Hardwired to 0.
7	Wait Cycle Control (WCC). Not implemented. Hardwired to 0.
6	Parity Error Response (PER). Not implemented. Hardwired to 0.
5	VGA Palette Snoop (VPS). Not implemented. Hardwired to 0.
4	Memory Write and Invalidate Enable (MWI). Not implemented. Hardwired to 0.
3	Special Cycle Enable (SCE). Not implemented. Hardwired to 0.
2	<b>Bus Master Enable (BME)</b> —R/W. Controls standard PCI bus mastering capabilities. 0 = Disable. 1 = Enable
1	Memory Space (MS). Hardwired to 0, AC '97 does not respond to memory accesses
0	<b>IOS (I/O Space)</b> —R/W. This bit controls access to the AC'97 Audio Controller I/O space registers. 0 = Disable (Default). 1 = Enable access to I/O space. The Native PCI Mode Base Address register should be programmed prior to setting this bit.

### 13.1.4 PCISTS—PCI Device Status Register (Audio—D31:F5)

Offset:	07h–06h	Attribute:	R/WC
Default Value:	0280h	Size:	16 bits
Lockable:	No	Power Well:	Core

PCISTA is a 16-bit status register. Refer to the PCI 2.1 specification for complete details on each bit.

Bit	Description
15	Detected Parity Error (DPE). Not implemented. Hardwired to 0.
14	SERR# Status (SERRS). Not implemented. Hardwired to 0.
13	<b>Master-Abort Status (MAS)</b> —R/WC. 1 = Bus Master AC '97 2.1 interface function, as a master, generates a master abort. 0 = Software clears this bit by writing a 1 to the bit position.
12	Reserved. Will always read as 0.
11	Signaled Target-Abort Status (STA). Not implemented. Hardwired to 0.
10:9	<b>DEVSEL# Timing Status (DEVT)</b> —RO. This 2-bit field reflects the ICH2's DEVSEL# timing when performing a positive decode. 01b = Medium timing.
8	Data Parity Detected (DPD). Not implemented. Hardwired to 0.
7	Fast Back to back Capable (FBC). Hardwired to 1. This bit indicates that the ICH2 as a target is capable of fast back-to-back transactions.
6	UDF Supported. Not implemented. Hardwired to 0.
5	66 MHz Capable. Hardwired to 0.
4:0	Reserved. Read as 0's.

### 13.1.5 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	<b>Revision ID Value</b> —RO. Refer to the ICH2 / ICH2-m Specification Update for the value of the Revision ID Register

### 13.1.6 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	<b>Programming Interface</b> —RO.

### 13.1.7 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	<b>Sub Class Code</b> —RO. 01h = Audio Device

### 13.1.8 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	<b>Base Class Code</b> —RO. 04h = Multimedia device

### 13.1.9 HEDT—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	<b>Header Type Value.</b> Hardwired to 00h.

### 13.1.10 NAMBAR—Native Audio Mixer Base Address Register (Audio—D31:F5)

Address Offset:	10h–13h	Attribute:	R/W
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.

For description of these I/O registers, refer to the AC'97 specification.

Bit	Description
31:16	Hardwired to 0s
15:8	<b>Base Address</b> —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	<b>Resource Type Indicator (RTE)</b> —RO. Hardwired to 1 indicating a request for I/O space.

### 13.1.11 NABMBAR—Native Audio Bus Mastering Base Address Register (Audio—D31:F5)

Address Offset:	14h–17h	Attribute:	R/W
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.

Bit	Description
31:16	Hardwired to 0s
15:6	<b>Base Address</b> —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	<b>Resource Type Indicator (RTE)</b> —RO. This bit is set to 1 indicating a request for I/O space.

### 13.1.12 SVID—Subsystem Vendor ID Register (Audio—D31:F5)

Address Offset:	2Dh–2Ch	Attribute:	Read/Write-Once
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.

Bit	Description
15:0	<b>Subsystem Vendor ID Value</b> —R/Write-Once.

### 13.1.13 SID—Subsystem ID Register (Audio—D31:F5)

Address Offset:	2Fh–2Eh	Attribute:	Read/Write-Once
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.

Bit	Description
15:0	<b>Subsystem ID Value</b> —R/Write-Once.

### 13.1.14 INTR\_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	<b>Interrupt Line</b> —R/W. This data is not used by the ICH2. It is used to communicate to software the interrupt line that the interrupt pin is connected to.

### 13.1.15 INTR\_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'ed to the interrupt controller with the PIRQB# signal.

Bit	Description
7:3	Reserved.
2:0	<b>AC '97 Interrupt Routing</b> —RO. Hardwired to 010b to select PIRQB#.

## 13.2 AC'97 Audio I/O Space (D31:F5)

The AC'97 I/O space includes Native Audio Bus Master Registers and Native Mixer Registers. Table 13-2 shows the register addresses for the audio mixer registers.

**Table 13-2. ICH2 Audio Mixer Register Configuration**

Primary offset	Secondary Offset	NAMBAR Exposed Registers (D31:F5)
00h	80h	Reset
02h	82h	Master Volume Mute
04h	84h	Headphone Volume Mute
06h	86h	Master Volume Mono Mute
08h	88h	Master Tone (R & L)
0Ah	8Ah	PC_BEEP Volume Mute
0Ch	8Ch	Phone Volume Mute
0Eh	8Eh	Mic Volume Mute
10h	90h	Line In Volume Mute
12h	92h	CD Volume Mute
14h	94h	Video Volume Mute
16h	96h	Aux Volume Mute
18h	98h	PCM Out Volume Mute
1Ah	9Ah	Record Select
1Ch	9Ch	Record Gain Mute
1Eh	9Eh	Record Gain Mic Mute
20h	A0h	General Purpose
22h	A2h	3D Control
24h	A4h	AC'97 RESERVED
26h	A6h	Powerdown Ctrl/Stat
28h	A8h	Extended Audio
2Ah	AAh	Extended Audio Ctrl/Stat

Table 13-2. ICH2 Audio Mixer Register Configuration (Continued)

Primary offset	Secondary Offset	NAMBAR Exposed Registers (D31:F5)
2Ch	ACh	PCM Front DAC Rate
2Eh	AEh	PCM Surround DAC Rate
30h	B0h	PCM LFE DAC Rate
32h	B2h	PCM LR ADC Rate
34h	B4h	MIC ADC Rate
36h	B6h	6Ch Vol: C, LFE Mute
38h	B8h	6Ch Vol: L, R Surround Mute
3Ah:56h	BAh–F6h	Intel RESERVED
<b>58h<sup>1</sup></b>		<b>Vendor Reserved</b>
<b>7Ah<sup>1</sup></b>		<b>Vendor Reserved</b>
<b>7Ch<sup>1</sup></b>		<b>Vendor ID1</b>
<b>7Eh<sup>1</sup></b>		<b>Vendor ID2</b>

**NOTE:**

1. Registers in bold are multiplexed between audio and modem functions
2. Software should not try to access reserved registers

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.

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.

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 three channels (PCM in, PCM out, and Mic in) each have their own set of Bus Mastering registers. The following register descriptions apply to all three channels. The register definition section titles use a generic “x\_” in front of the register to indicate that the register applies to all three channels. The naming prefix convention used in Table 13-3 and in the register description I/O address is as follows:

- PI = PCM in channel
- PO = PCM out channel
- MC = Mic in channel.

**Table 13-3. Native Audio Bus Master Control Registers**

Offset	Mnemonic	Name	Default	Access
00h	PI_BDBAR	PCM In Buffer Descriptor list Base Address Register	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 Register	0001h	R/W
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 Register	00h	R/W
10h	PO_BDBAR	PCM Out Buffer Descriptor list Base Address Register	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 Register	0001h	R/W
18h	PO_PICB	PCM Out Position In Current Buffer	0000h	RO
1Ah	PO_PIV	PCM Out Prefetched Index Value	00h	RO
1Bh	PO_CR	PCM Out Control Register	00h	R/W
20h	MC_BDBAR	Mic. In Buffer Descriptor list Base Address Register	00000000h	R/W
24h	PM_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 Register	0001h	R/W
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 Register	00h	R/W
2Ch	GLOB_CNT	Global Control	00000000h	R/W
30h	GLOB_STA	Global Status	00000000h	RO
34h	ACC_SEMA	Codec Write Semaphore Register	00h	R/W

### 13.2.1 x\_BDBAR—Buffer Descriptor Base Address Register

I/O Address: NABMBAR + 00h (PIBDBAR), Attribute: R/W  
 NABMBAR + 10h (POBDBAR),  
 NABMBAR + 20h (MCBDBAR)  
 Default Value: 00000000h Size: 32 bits  
 Lockable: No Power Well: Core

Bit	Description
31:3	<b>Buffer Descriptor Base Address[31:3]</b> —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.

### 13.2.2 x\_CIV—Current Index Value Register

I/O Address:	NABMBAR + 04h (PICIV), NABMBAR + 14h (POCIV), NABMBAR + 24h (MCCIV)	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	<b>Current Index Value[4:0]</b> —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.

### 13.2.3 x\_LVI—Last Valid Index Register

I/O Address:	NABMBAR + 05h (PILVI), NABMBAR + 15h (POLVI), NABMBAR + 25h (MCLVI)	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	<b>Last Valid Index[4:0]</b> —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.

## 13.2.4 x\_SR—Status Register

I/O Address:	NABMBAR + 06h (PISR), NABMBAR + 16h (POSR), NABMBAR + 26h (MCSR)	Attribute:	R/WC, RO
Default Value:	0001h	Size:	16 bits
Lockable:	No	Power Well:	Core

Bit	Description
15:5	Reserved.
4	<p><b>FIFO error (FIFOE)—R/WC.</b>  1 = FIFO error occurs.  0 = Cleared by writing a 1 to this bit position.</p> <p>PISR Register: FIFO error indicates a FIFO overrun. The FIFO pointers do not increment, the incoming data is not written into the FIFO, thus is lost.</p> <p><b>POSR Register:</b> FIFO error indicates a FIFO underrun. The sample transmitted in this case should be the last valid sample.</p> <p>The ICH2 will set the FIFOE bit if the under-run or overrun occurs when there are more valid buffers to process.</p>
3	<p><b>Buffer Completion Interrupt Status (BCIS)—R/WC.</b>  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.  0 = Cleared by writing a 1 to this bit position.</p>
2	<p><b>Last Valid Buffer Completion Interrupt (LVBCI)—R/WC.</b>  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.</p> <p>In the case of <i>Transmits</i> (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 <i>Receives</i>, this bit is set after the data for the last buffer has been written to memory.</p> <p>0 = Cleared by writing a 1 to this bit position.</p>
1	<p><b>Current Equals Last Valid (CELV)—RO.</b>  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 = Cleared by hardware when controller exists state (i.e., until a new value is written to the LVI register.)</p>
0	<p><b>DMA Controller Halted (DCH)—RO.</b>  1 = Halted. This could happen because of the Start/Stop bit being cleared, or it could happen once the controller has processed the last valid buffer (in which case it will set bit 1 and halt).</p>

Software can read the above 3 registers simultaneously by scheduling a single 32 bit read from address offset 04h. Software can also read this individual register by performing a 16 bit read from 06h.

### 13.2.5 x\_PICB—Position In Current Buffer Register

I/O Address: NABMBAR + 08h (PIPICB), Attribute: RO  
 NABMBAR + 18h (POPICB),  
 NABMBAR + 28h (MCPICB)  
 Default Value: 0000h Size: 16 bits  
 Lockable: No Power Well: Core

Bit	Description
15:0	<b>Position In Current Buffer[15:0]</b> —RO. These bits represent the number of DWords remaining to be processed in the current buffer; 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.

### 13.2.6 x\_PIV—Prefetched Index Value Register

I/O Address: NABMBAR + 0Ah (PIPIV), Attribute: RO  
 NABMBAR + 1Ah (POPIV),  
 NABMBAR + 2Ah (MCPIV)  
 Default Value: 00h Size: 8 bits  
 Lockable: No Power Well: Core

Bit	Description
7:5	Hardwired to 0.
4:0	<b>Prefetched Index Value[4:0]</b> —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.

## 13.2.7 x\_CR—Control Register

I/O Address:	NABMBAR + 0Bh (PICR), NABMBAR + 1Bh (POCR), NABMBAR + 2Bh (MCCR)	Attribute:	R/W
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:5	Reserved.
4	<b>Interrupt On Completion Enable (IOCE)</b> —R/W. This bit controls whether or not an interrupt occurs when a buffer completes with the IOC bit set in its descriptor. 0 = Disable. Interrupt will not occur. 1 = Enable.
3	<b>FIFO Error Interrupt Enable (FEIE)</b> —R/W. This bit controls whether the occurrence of a FIFO error will cause an interrupt or not. 0 = Disable. Bit 4 in the Status Register will be set; however, the interrupt will not occur. 1 = Enable. Interrupt will occur.
2	<b>Last Valid Buffer Interrupt Enable (LVBIE)</b> —R/W. This bit controls whether the completion of the last valid buffer will cause an interrupt or not. 0 = Disable. Bit 2 in the Status register will still be set; however, the interrupt will not occur. 1 = Enable.
1	<b>Reset Registers (RR)</b> —R/W (special). 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 = Removes reset condition.
0	<b>Run/Pause Bus master (RPBM)</b> —R/W. 0 = Pause bus master operation. This results in all state information being retained (i.e., master mode operation can be stopped and then resumed). 1 = Run. Bus master operation starts.

## 13.2.8 GLOB\_CNT—Global Control Register

I/O Address:	NABMBAR + 2Ch	Attribute:	R/W
Default Value:	00000000h	Size:	32 bits
Lockable:	No	Power Well:	Core

Bit	Description
31:22	Reserved.
21:20	<b>PCM 4/6 Enable</b> —R/W. Configures PCM Output for 2, 4 or 6 channel mode. 00 = 2-channel mode (default) 01 = 4-channel mode 10 = 6-channel mode 11 = Reserved
19:6	Reserved.
5	<b>Secondary Resume Interrupt Enable</b> —R/W. 0 = Disable. 1 = Enable an interrupt to occur when the secondary codec causes a resume event on the AC-link.
4	<b>Primary Resume Interrupt Enable</b> —R/W. 0 = Disable. 1 = Enable an interrupt to occur when the primary codec causes a resume event on the AC-link.
3	<b>ACLINK Shut Off</b> —R/W. 0 = Normal operation. 1 = Drive all AC'97 outputs low and turn off all AC'97 input buffer enables
2	<b>AC'97 Warm Reset</b> —R/W (special). 0 = Normal operation. 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 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 bit_clk is seen on the ACLink, after which it clears itself).
1	<b>AC '97 Cold Reset#</b> —R/W. 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. 1 = This bit defaults to 0; thus, after reset, the driver needs to set this bit to a 1. The value of this bit is retained after suspends; thus, if this bit is set to a 1 prior to suspending, a cold reset is not generated automatically upon resuming. Note: This bit is in the Resume well, not in the Core well.
0	<b>GPI Interrupt Enable (GIE)</b> —R/W. This bit controls whether the change in status of any GPI causes an interrupt. 0 = Bit 0 of the Global Status Register is set, but no interrupt is generated. 1 = The change on value of a GPI causes an interrupt and sets bit 0 of the Global Status Register.



### 13.2.9 GLOB\_STA—Global Status Register

I/O Address: NABMBAR + 30h      Attribute: RO, R/W, R/WC  
Default Value: 00300000h      Size: 32 bits  
Lockable: No      Power Well: Core

Bit	Description
31:22	Reserved.
21	<b>6 Channel Capability (6CH_CAP)</b> —RO. Hardwired to 1. 0 = The AC '97 Controller does not support 6-channel PCM Audio output. 1 = The AC '97 Controller supports 6-channel PCM Audio output.
20	<b>4 Channel Capability (4CH_CAP)</b> —RO. Hardwired to 1. 0 = The AC '97 Controller does not support 4-channel PCM Audio output. 1 = The AC '97 Controller supports 4-channel PCM Audio output.
19:18	Reserved.
17	<b>MD3</b> —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.
16	<b>AD3</b> —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.
15	<b>Read Completion Status (RCS)</b> —R/WC. This bit indicates the status of codec read completions. 0 = A codec read completes normally. 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.
14	<b>Bit 3 of slot 12</b> —RO. Display bit 3 of the most recent slot 12.
13	<b>Bit 2 of slot 12</b> —RO. Display bit 2 of the most recent slot 12.
12	<b>Bit 1 of slot 12</b> —RO. Display bit 1 of the most recent slot 12.
11	<b>Secondary Resume Interrupt (SRI)</b> —R/WC. This bit indicates that a resume event occurred on AC_SDIN[1]. 1 = Resume event occurred 0 = Cleared by writing a 1 to this bit position.
10	<b>Primary Resume Interrupt (PRI)</b> —R/WC. This bit indicates that a resume event occurred on AC_SDIN[0]. 1 = Resume event occurred 0 = Cleared by writing a 1 to this bit position.
9	<b>Secondary Codec Ready (SCR)</b> —RO. Reflects the state of the codec ready bit in AC_SDIN[1]. 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. 0 = Not Ready. 1 = Ready.
8	<b>Primary Codec Ready (PCR)</b> —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. 0 = Not Ready. 1 = Ready.
7	<b>Mic In Interrupt (MINT)</b> —RO. This bit indicates that one of the Mic in channel interrupts occurred. 1 = Interrupt occurred. 0 = When the specific interrupt is cleared, this bit will be cleared.

Bit	Description
6	<b>PCM Out Interrupt (POINT)</b> —RO. This bit indicates that one of the PCM out channel interrupts occurred. 1 = Interrupt occurred. 0 = When the specific interrupt is cleared, this bit will be cleared.
5	<b>PCM In Interrupt (PIINT)</b> —RO. This bit indicates that one of the PCM in channel interrupts occurred. 1 = Interrupt occurred. 0 = 0 = When the specific interrupt is cleared, this bit will be cleared.
4:3	Reserved
2	<b>Modem Out Interrupt (MOINT)</b> —RO. This bit indicates that one of the modem out channel interrupts occurred. 1 = Interrupt occurred. 0 = When the specific interrupt is cleared, this bit will be cleared.
1	<b>Modem In Interrupt (MIINT)</b> —RO. This bit indicates that one of the modem in channel interrupts occurred. 1 = Interrupt occurred. 0 = When the specific interrupt is cleared, this bit will be cleared.
0	<b>GPI Status Change Interrupt (GSCI)</b> —RWC. This bit reflects the state of bit 0 in slot 12, and is set whenever bit 0 of slot 12 is set. This happens when the value of any of the GPIOs currently defined as inputs changes. 1 = Input changed. 0 = Cleared by writing a 1 to this bit position.

### 13.2.10 CAS—Codec Access Semaphore Register

I/O Address:	NABMBAR + 34h	Attribute:	R/W
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:1	Reserved.
0	<b>Codec Access Semaphore (CAS)</b> —R/W (special). This bit is read by software to check whether a codec access is currently in progress. 0 = No access in progress. 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.

# AC'97 Modem Controller Registers (D31:F6)

14

## 14.1 AC'97 Modem PCI Configuration Space (D31:F6)

**Note:** Registers that are not shown should be treated as Reserved (See [Section 6.2](#) for details).

**Table 14-1. PCI Configuration Map (Modem—D31:F6)**

Offset	Mnemonic	Register	Default	Access
00h–01h	VID	Vendor Identification	8086	RO
02h–03h	DID	Device Identification	2446h	RO
04h–05h	PCICMD	PCI Command	0000	R/W
06h–07h	PCISTA	PCI Device Status	0280h	R/WC
08h	RID	Revision Identification	See Note	RO
09h	PI	Programming Interface	00	RO
0Ah	SCC	Sub Class Code	03h	RO
0Bh	BCC	Base Class Code	07h	RO
0Eh	HEDT	Header Type	00	RO
0Fh	—	Reserved	—	—
10h–13h	MMBAR	Modem Mixer Base Address	00000001h	R/W
14h–17h	MBAR	Modem Base Address	00000001h	R/W
18h–1Bh	—	Reserved	00000001h	—
1Ch–2Bh	—	Reserved	—	—
2Ch–2Dh	SVID	Subsystem Vendor ID	0000h	Write-Once
2Eh–2Fh	SID	Subsystem ID	0000h	Write-Once
30h–3Bh	—	Reserved	—	—
3Ch	INTR_LN	Interrupt Line	00h	RO
3Dh	INT_PN	Interrupt Pin	02h	RO
3Eh–FFh	—	Reserved	—	—

**NOTE:** Refer to the Specification Update for the value of the Revision ID Register

### 14.1.1 VID—Vendor Identification Register (Modem—D31:F6)

Address Offset: 01h–00h      Attribute: RO  
Default Value: 8086      Size: 16 Bits  
Lockable: No      Power Well: Core

Bit	Description
15:0	Vendor ID Value.

### 14.1.2 DID—Device Identification Register (Modem—D31:F6)

Address Offset:	03h–02h	Attribute:	RO
Default Value:	2446h	Size:	16 Bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	<b>Device ID Value.</b>

### 14.1.3 PCICMD—PCI Command Register (Modem—D31:F6)

Address Offset:	05h–04h	Attribute:	R/W
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

PCICMD is a 16-bit control register. Refer to the PCI 2.1 specification for complete details on each bit.

Bit	Description
15:10	Reserved. Read 0.
9	Fast Back to Back Enable (FBE). Not implemented. Hardwired to 0.
8	SERR# Enable (SEN). Not implemented. Hardwired to 0.
7	Wait Cycle Control (WCC). Not implemented. Hardwired to 0.
6	Parity Error Response (PER). Not implemented. Hardwired to 0.
5	VGA Palette Snoop (VPS). Not implemented. Hardwired to 0.
4	Memory Write and Invalidate Enable (MWI). Not implemented. Hardwired to 0.
3	Special Cycle Enable (SCE). Not implemented. Hardwired to 0.
2	<b>Bus Master Enable (BME)</b> —R/W. Controls standard PCI bus mastering capabilities. 0 = Disable. 1 = Enable
1	Memory Space (MS). Hardwired to 0, AC '97 does not respond to memory accesses.
0	<b>I/O Space (IOS)</b> —R/W. This bit controls access to the I/O space registers. 0 = Disable access. (default = 0). 1 = Enable access to I/O space. The Native PCI Mode Base Address register should be programmed prior to setting this bit.

### 14.1.4 PCISTA—Device Status Register (Modem—D31:F6)

Address Offset:	07h–06h	Attribute:	R/WC
Default Value:	0280h	Size:	16 bits
Lockable:	No	Power Well:	Core

PCISTA is a 16-bit status register. Refer to the PCI 2.1 specification for complete details on each bit.

Bit	Description
15	Detected Parity Error (DPE)—RO. Not implemented. Hardwired to 0.
14	SERR# Status (SERRS)—RO. Not implemented. Hardwired to 0.
13	<b>Master-Abort Status (MAS)</b> —R/WC. 1 = Bus Master AC '97 interface function, as a master, generates a master abort. 0 = Software clears this bit by writing a 1 to the bit position.
12	Reserved. Read as “0”.
11	Signaled Target-Abort Status (STA)—RO. Not implemented. Hardwired to 0.
10:9	DEVSEL# Timing Status (DEVT)—RO. This 2-bit field reflects the ICH2's DEVSEL# timing parameter. These read only bits indicate the ICH2's DEVSEL# timing when performing a positive decode.
8	Data Parity Detected (DPD)—RO. Not implemented. Hardwired to 0.
7	Fast Back to back Capable (FBC)—RO. Hardwired to 1. This bit indicates that the ICH2 as a target is capable of fast back-to-back transactions.
6	UDF Supported—RO. Not implemented. Hardwired to 0.
5	66 MHz Capable—RO. Hardwired to 0.
4:0	Reserved. Read as 0s.

### 14.1.5 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	<b>Revision ID Value</b> —RO. Refer to the Specification Update for the value of the Revision ID Register

### 14.1.6 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	<b>Programming Interface Value</b> —RO.

### 14.1.7 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	<b>Sub Class Code Value</b> —RO. 03h = Generic Modem.

### 14.1.8 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	<b>Base Class Code Value</b> —RO. 07h = Simple Communications Controller.

### 14.1.9 HEDT—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	<b>Header Value</b> —RO.

### 14.1.10 MMBAR—Modem Mixer Base Address Register (Modem—D31:F6)

Address Offset:	10h–13h	Attribute:	R/W
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	<b>Base Address</b> —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	<b>Resource Type Indicator (RTE)</b> —RO. This bit is set to one, indicating a request for I/O space.

### 14.1.11 MBAR—Modem Base Address Register (Modem—D31:F6)

Address Offset:	14h–17h	Attribute:	R/W
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	<b>Base Address</b> —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	<b>Resource Type Indicator (RTE)</b> —RO. This bit is set to one, indicating a request for I/O space.

### 14.1.12 SVID—Subsystem Vendor ID (Modem—D31:F6)

Address Offset:	2Dh–2Ch	Attribute:	Write-Once
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 the register, the value can be read back. Any subsequent writes will have no effect.

Bit	Description
15:0	<b>Subsystem Vendor ID Value</b> —Read/Write-Once.

### 14.1.13 SID—Subsystem ID (Modem—D31:F6)

Address Offset:	2Fh–2Eh	Attribute:	Write-Once
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Core

The SID register, in combination with the Subsystem Vendor ID register, makes it possible for the operating environment to distinguish one audio subsystem from another. This register is implemented as a write-once register. Once a value is written to the register, the value can be read back. Any subsequent writes will have no effect.

Bit	Description
15:0	<b>Subsystem ID Value</b> —Read/Write-Once.

### 14.1.14 INTR\_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	<b>Interrupt Line</b> —R/W. This data is not used by the ICH2. It is used to communicate to software the interrupt line that the interrupt pin is connected to.

### 14.1.15 INT\_PIN—Interrupt Pin (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 ORED to the interrupt controller with the PIRQB# signal.

Bit	Description
7:3	Reserved.
2:0	<b>AC '97 Interrupt Routing</b> —RO. Hardwired to 010b to select PIRQB#.

## 14.2 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 14-2 shows the register addresses for the modem mixer registers.

**Table 14-2. ICH2 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 Status/Control
40h	C0h	Line 1 DAC/ADC Rate
42h	C2h	<i>Line 2 DAC/ADC Rate<sup>2</sup></i>
44h	C4h	<i>Handset DAC/ADC Rate<sup>2</sup></i>
46h	C6h	Line 1 DAC/ADC Level Mute
48h	C8h	<i>Line 2 DAC/ADC Level Mute<sup>2</sup></i>
4Ah	CAh	<i>Handset DAC/ADC Level Mute<sup>2</sup></i>
4Ch	CCh	GPIO Pin Configuration
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 <sup>1</sup>	D8h	<b>Vendor Reserved</b>
7Ah <sup>1</sup>	FAh	<b>Vendor Reserved</b>
7Ch <sup>1</sup>	FCh	<b>Vendor ID1</b>
7Eh <sup>1</sup>	FEh	<b>Vendor ID2</b>

**NOTE:**

1. Registers in bold are multiplexed between audio and modem functions
2. Registers in italics are for functions not supported by the ICH2
3. Software should not try to access reserved registers
4. The ICH2 supports a modem codec as either primary or secondary, but does not support two modem codecs.

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.

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 14-3. Modem Registers

Offset	Mnemonic	Name	Default	Access
00h	MI_BDBAR	Modem In Buffer Descriptor List Base Address Register	00000000h	R/W
04h	MI_CIV	Modem In Current Index Value Register	00h	R
05h	MI_LVI	Modem In Last Valid Index Register	00h	R/W
06h	MI_SR	Modem In Status Register	0001h	R/W
08h	MI_PICB	Modem In Position In Current Buffer Register	00h	R
0Ah	MI_PIV	Modem In Prefetch Index Value Register	00h	RO
0Bh	MI_CR	Modem In Control Register	00h	R/W
10h	MO_BDBAR	Modem Out Buffer Descriptor List Base Address Register	00000000h	R/W
14h	MO_CIV	Modem Out Current Index Value Register	00h	RO
15h	MO_LVI	Modem Out Last Valid Register	00h	R/W
16h	MO_SR	Modem Out Status Register	0001h	R/W
18h	MI_PICB	Modem In Position In Current Buffer Register	00h	RO
1Ah	MO_PIV	Modem Out Prefetched Index Register	00h	RO
1Bh	MO_CR	Modem Out Control Register	00h	R/W
3Ch	GLOB_CNT	Global Control	00000000h	R/W
40h	GLOB_STA	Global Status	00000000h	RO
44h	ACC_SEMA	Codec Write Semaphore Register	00h	R/W

**NOTE:**

1. MI = Modem in channel; MO = Modem out channel

## 14.2.1 x\_BDBAR—Buffer Descriptor List Base Address Register

I/O Address: MBAR + 00h (MIBDBAR), Attribute: R/W  
 MBAR + 10h (MOBDBAR)  
 Default Value: 00000000h Size: 32bits  
 Lockable: No Power Well: Core

Bit	Description
31:3	<b>Buffer Descriptor List Base Address[31:3]</b> —R/W. These bits represent address bits 31:3. The entries should be aligned on 8 byte boundaries.
2:0	Hardwired to 0.

### 14.2.2 x\_CIV—Current Index Value Register

I/O Address:	MBAR + 04h (MICIV), MBAR + 14h (MOCIV),	Attribute:	RO
Default Value:	00h	Size:	8bits
Lockable:	No	Power Well:	Core

Bit	Description
7:5	Hardwired to 0.
4:0	<b>Current Index Value [4:0]</b> —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.

### 14.2.3 x\_LVI—Last Valid Index Register

I/O Address:	MBAR + 05h (MILVI), MBAR + 15h (MOLVI)	Attribute:	R/W
Default Value:	00h	Power Well:	Core

Bit	Description
7:5	Hardwired to 0
4:0	<b>Last Valid Index [4:0]</b> —R/W. These bits indicate the last valid descriptor in the list. This value is updated by software as it prepares new buffers and adds to the list.

## 14.2.4 x\_SR—Status Register

I/O Address:	MBAR + 06h (MISR), MBAR + 16h (MOSR)	Attribute:	R/WC
Default Value:	0001h	Size:	16 bits
Lockable:	No	Power Well:	Core

Bit	Description
15:5	Reserved.
4	<p><b>FIFO error (FIFOE)—R/WC.</b>  1 = FIFO error occurs.  0 = Cleared by writing a 1 to this bit position.</p> <p><b>Modem in:</b> FIFO error indicates a FIFO overrun. The FIFO pointers do not increment, the incoming data is not written into the FIFO, thereby being lost.</p> <p><b>Modem out:</b> FIFO error indicates a FIFO underrun. The sample transmitted in this case should be the last valid sample.</p> <p>The ICH2 sets the FIFOE bit if the under-run or overrun occurs when there are more valid buffers to process.</p>
3	<p><b>Buffer Completion Interrupt Status (BCIS)—R/WC.</b>  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.  0 = Cleared by writing a 1 to this bit position.</p>
2	<p><b>Last Valid Buffer Completion Interrupt (LVBCI)—R/WC.</b>  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.</p> <p>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.</p> <p>0 = Cleared by writing a 1 to this bit position</p>
1	<p><b>Current Equals Last Valid (CELV)—RO.</b>  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.</p> <p>0 = Hardware clears when controller exists state (i.e., until a new value is written to the LVI register).</p>
0	<p><b>DMA Controller Halted (DCH)—RO.</b>  1 = DMA controller is halted. This could happen because of the Start/Stop bit being cleared, or it could happen once the controller has processed the last valid buffer (in which case it will set bit 1 and halt).</p>

### 14.2.5 x\_PICB—Position In Current Buffer Register

I/O Address: MBAR + 08h (MIPICB), MBAR + 18h (MOPICB), Attribute: RO  
Default Value: 0000h Size: 16 bits  
Lockable: No Power Well: Core

Bit	Description
15:0	<b>Position In Current Buffer[15:0]</b> —RO. These bits represent the number of dwords left to be processed in the current buffer.

### 14.2.6 x\_PIV—Prefetch Index Value Register

I/O Address: MBAR + 0Ah (MIPIV), MBAR + 1Ah (MOPIV), Attribute: RO  
Default Value: 00h Size: 8 bits  
Lockable: No Power Well: Core

Bit	Description
7:5	Hardwired to 0
4:0	<b>Prefetched Index value [4:0]</b> —RO. These bits represent which buffer descriptor in the list has been prefetched.

### 14.2.7 x\_CR—Control Register

I/O Address: MBAR + 0Bh (MICR), MBAR + 1Bh (MOCR), Attribute: R/W  
Default Value: 00h Size: 8 bits  
Lockable: No Power Well: Core

Bit	Description
7:5	Reserved.
4	<b>Interrupt On Completion Enable (IOCE)</b> —R/W. This bit controls whether or not an interrupt occurs when a buffer completes with the IOC bit set in its descriptor. 0 = Disable. 1 = Enable.
3	<b>FIFO Error Interrupt Enable (FEIE)</b> —R/W. This bit controls whether the occurrence of a FIFO error will cause an interrupt or not. 0 = Disable. Bit 4 in the Status Register will be set, but the interrupt will not occur. 1 = Enable. Interrupt will occur
2	<b>Last Valid Buffer Interrupt Enable (LVBIE)</b> —R/W. This bit controls whether the completion of the last valid buffer will cause an interrupt or not. 0 = Disable. Bit 2 in the Status register will still be set, but the interrupt will not occur. 1 = Enable.
1	<b>Reset Registers (RR)</b> —R/W (special). 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 does not need to clear it). 0 = Removes reset condition.
0	<b>Run/Pause Bus master (RPBM)</b> —R/W. 0 = Pause bus master operation. This results in all state information being retained (i.e., master mode operation can be stopped and then resumed). 1 = Run. Bus master operation starts.

## 14.2.8 GLOB\_CNT—Global Control Register

I/O Address:	MBAR + 3Ch	Attribute:	R/W
Default Value:	00000000h	Size:	32 bits
Lockable:	No	Power Well:	Core

Bit	Description
31:6	Reserved.
5	<b>Secondary Resume Interrupt Enable</b> —R/W. 0 = Disable. 1 = Enable an interrupt to occur when the secondary codec causes a resume event on the AC-link.
4	<b>Primary Resume Interrupt Enable</b> —R/W. 0 = Disable. 1 = Enable an interrupt to occur when the primary codec causes a resume event on the AC-link.
3	<b>ACLINK Shut Off</b> —R/W. 0 = Normal operation. 1 = Disable the AC-link signals (drive all AC'97 outputs low and turn off all AC'97 input buffer enables)
2	<b>AC'97 Warm Reset</b> —R/W (special). 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 BIT_CLK is running, the write will be ignored and the bit will not be changed. A warm reset can only occur in the absence of BIT_CLK. 0 = This bit is self-clearing (it clears itself after the reset has occurred and BIT_CLK has started).
1	<b>AC'97 Cold Reset#</b> —R/W (special). 0 = Writing a 0 to this bit causes a cold reset to occur throughout the AC'97 circuitry. All data in the codec will be lost. Software needs to clear this bit no sooner than after 1usec has elapsed. This bit reflects the state of the AC_RST# pin. The ICH2 clears this bit to "0" upon entering S3/S4/S5 sleep states and PCIRST#.
0	<b>GPI Interrupt Enable (GIE)</b> —R/W. This bit controls whether the change in status of any GPI causes an interrupt. 0 = Bit 0 of the Global Status Register is set, but an interrupt is not generated. 1 = The change on value of a GPI causes an interrupt and sets bit 0 of the Global Status Register.

## 14.2.9 GLOB\_STA—Global Status Register

I/O Address:	MBAR + 40h	Attribute:	RO, R/W, R/WC
Default Value:	00300000h	Size:	32 bits
Lockable:	No	Power Well:	Core

Bit	Description
31:22	Reserved.
21	<b>6 Channel Capability (6CH_CAP)</b> —RO. Hardwired to 1. 0 = The AC '97 Controller does not support 6-channel PCM Audio output. 1 = The AC '97 Controller supports 6-channel PCM Audio output.
20	<b>4 Channel Capability (4CH_CAP)</b> —RO. Hardwired to 1. 0 = The AC '97 Controller does not support 4-channel PCM Audio output. 1 = The AC '97 Controller supports 4-channel PCM Audio output.
19:18	Reserved.
17	<b>MD3</b> —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.
16	<b>AD3</b> —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.
15	<b>Read Completion Status (RCS)</b> —R/W. This bit indicates the status of codec read completions. 0 = A codec read completes normally. 1 = A codec read results in a time-out. The bit remains set until being cleared by software.
14	<b>Bit 3 of slot 12</b> —RO. Display bit 3 of the most recent slot 12
13	<b>Bit 2 of slot 12</b> —RO. Display bit 2 of the most recent slot 12
12	<b>Bit 1 of slot 12</b> —RO. Display bit 1 of the most recent slot 12
11	<b>Secondary Resume Interrupt (SRI)</b> —R/WC. This bit indicates that a resume event occurred on AC_SDIN[1]. 1 = Resume event occurred 0 = Cleared by writing a 1 to this bit position.
10	<b>Primary Resume Interrupt (PRI)</b> —R/WC. This bit indicates that a resume event occurred on AC_SDIN[0]. 1 = Resume event occurred 0 = Cleared by writing a 1 to this bit position.
9	<b>Secondary Codec Ready (SCR)</b> —RO. Reflects the state of the codec ready bit in AC_SDIN[1]. 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. 0 = Not Ready. 1 = Ready.
8	<b>Primary Codec Ready (PCR)</b> —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. 0 = Not Ready. 1 = Ready.
7	<b>Mic In Interrupt (MINT)</b> —RO. This bit indicates that one of the Mic in channel interrupts occurred. 1 = Interrupt occurred. 0 = When the specific interrupt is cleared, this bit will be cleared.
6	<b>PCM Out Interrupt (POINT)</b> —RO. This bit indicates that one of the PCM out channel interrupts occurred. 1 = Interrupt occurred. 0 = When the specific interrupt is cleared, this bit will be cleared.

Bit	Description
5	<b>PCM In Interrupt (PIINT)</b> —RO. This bit indicates that one of the PCM in channel interrupts occurred. 1 = Interrupt occurred. 0 = 0 = When the specific interrupt is cleared, this bit will be cleared.
4:3	Reserved
2	<b>Modem Out Interrupt (MOINT)</b> —RO. This bit indicates that one of the modem out channel interrupts occurred. 1 = Interrupt occurred. 0 = When the specific interrupt is cleared, this bit will be cleared.
1	<b>Modem In Interrupt (MIINT)</b> —RO. This bit indicates that one of the modem in channel interrupts occurred. 1 = Interrupt occurred. 0 = When the specific interrupt is cleared, this bit will be cleared.
0	<b>GPI Status Change Interrupt (GSCI)</b> —RWC. This bit reflects the state of bit 0 in slot 12, and is set when bit 0 of slot 12 is set. This happens when the value of any of the GPIOs currently defined as inputs changes. 1 = Input changed. 0 = Cleared by writing a 1 to this bit position.

**Note:** On reads from a codec, the controller will give the codec a maximum of 4 frames to respond, after which if no response is received, it will return a dummy read completion to the processor (with all pHs on the data) and also set the Read Completion Status bit in the Global Status Register.

## 14.2.10 CAS—Codec Access Semaphore Register

I/O Address:	NABMBAR + 44h	Attribute:	R/W
Default Value:	00h	Size:	8 bits
Lockable:	No	Power Well:	Core

Bit	Description
7:1	Reserved.
0	<b>Codec Access Semaphore (CAS)</b> —R/W (special). This bit is read by software to check whether a codec access is currently in progress. 0 = No access in progress. 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.

# ***Pinout and Package Information*** **15**

---

## **15.1 Pinout**

This section contains the ICH2 ballout information. [Figure 15-1](#) and [Figure 15-2](#) provide a graphical illustration of how the ballout maps to the 360 EPGA package. [Table 15-1](#) provides the ballout, listed alphabetically by signal name.

Figure 15-1. ICH2 Ballout (Top view — Left side)

	1	2	3	4	5	6	7	8	9	10	11
A	VSS	VSS	HLCOMP	HL0	HL2	HL_STB	HL_STB#	HL5	HL7	VSS	IGNNE#
B	VSS	VSS	VSS	HUBREF	HL1	HL3	HL4	HL6	VSS	VSS	NMI
C	#N/A	VSS	VSS	VSS	HL11	HL9	HL10	HL8	VSS	STPCLK#	INTR
D	#N/A	VCC1_8	VSS	CLK66	VSS	VSS	VSS	VSS	VSS	VCC1_8	A20M#
E	#N/A	#N/A	#N/A	#N/A	VCC1_8	VSS	VSS	VSS	VSS		
F	LAN_TXD2	LAN_TXD1	LAN_TXD0	#N/A	VCCSUS3_3						
G	LAN_RXD1	LAN_RXD0	LAN_CLK	#N/A	VCCSUS3_3						
H	LAN_RXD2	LAN_RSTSYNC	#N/A	#N/A	VCCSUS1_8						
J	#N/A	#N/A	EE_SHCLK	EE_DOUT	VCCSUS1_8					VSS	VSS
K	VSS	V5REF1	EE_DIN	EE_CS						VSS	VSS
L	GPIO21	GNTA# / GPIO16	REQB# / REQ5# / GPIO1	GNTB# / GNT5# / GPIO17						VSS	VSS
M	GNT1#	GNT0#	REQA# / GPIO0	PIRQH#						VSS	VSS
N	PIRQG# / GPIO4	PIRQF# / GPIO3	PIRQE#	PIRQD#						VSS	VSS
P	PIRQA#	PIRQB#	PIRQC#	REQ4#	VCC1_8					VSS	VSS
R	GNT4#	REQ0#	REQ1#	GNT2#	VCC3_3						
T	REQ2#	GNT3#	AD30	AD28	VCC3_3						
U	AD26	AD24	AD22	AD20	VCC3_3						
V	AD18	AD16	FRAME#	TRDY#	VCC3_3	VCC3_3	VCC3_3	VCC3_3	VCC1_8		
W	STOP#	PAR	AD11	AD4	AD3	AD10	SERR#	IRDY#	AD21	AD27	PCICLK
Y	AD15	AD13	AD9	AD2	AD5	AD12	PERR#	C/BE2#	AD23	AD29	GPIO6
AA	VSS	VSS	C/BE0#	AD0	AD7	AD14	PLOCK#	AD17	C/BE3#	AD31	GPIO7
AB	VSS	VSS	AD6	AD1	AD8	C/BE1#	DEVSEL#	AD19	AD25	REQ3#	LFRAME# / FWH4
	1	2	3	4	5	6	7	8	9	10	11

**Figure 15-2. ICH2 Ballout (Top view — Right side)**

12	13	14	15	16	17	18	19	20	21	22	
CPUSLP#	CPUPWRGD	GPIO23	GPIO18	SDA0	SIORDY	SDD15	SDD13	SDD3	VSS	VSS	A
SMI#	RCIN#	GPIO22	VRMPWRGD	SDA2	SDDACK#	SDDREQ	SDD1	SDD11	VSS	VSS	B
INIT#	A20GATE	GPIO20	SDCS1#	IRQ15	SDIOW#	SDD14	SDD12	SDD4	SDD5	SDD9	C
V_CPU_IO	V_CPU_IO	GPIO19	SDCS3#	SDA1	SDIOR#	SDD0	SDD2	SDD10	SDD8	SDD6	D
		VCC3_3	VCC3_3	VCC3_3	VCC3_3	VCC3_3	PDCS3#	SDD7	PDCS1#	PDA2	E
						VCC3_3	PDA1	PDA0	IRQ14	PDDACK#	F
						VCC3_3	PDIOR#	PIORDY	PDIOV#	PDDREQ	G
						VCC3_3	PDD0	PDD15	PDD14	PDD1	H
						VCC3_3	PDD2	PDD13	PDD12	PDD3	J
							VCC1_8	PDD11	PDD4	PDD10	K
							VCC1_8	PDD5	PDD9	PDD8	L
							CLK14	V5REF2	PDD6	PDD7	M
							APICD1	APICCLK	SERIRQ	SPKR	N
						VCC3_3	AC_SYNC	CLK48	AC_SDOUT	APICD0	P
						VCC3_3	AC_BITCLK	PWROK	RSMRST#	FERR#	R
						VCCSUS3_3	INTRUDER#	RTCRST#	VBIAS	RTCX2	T
						VCCSUS3_3	SMLINK0	TP0	VCCRTC	RTCX1	U
		VCCSUS1_8	VCCSUS1_8	VCCSUS1_8	VCCSUS3_3	VCCSUS3_3	V5REF_SUS	SMLINK1	GPIO24	AC_RST#	V
LAD1 / FWH1	LDRQ1#	GPIO12	GPIO25	SLP_S3#	USBP0P	USBP2P	OC0#	OC3#	PWRBTN#	AC_SDIN1	W
LAD0 / FWH0	LDRQ0#	GPIO8	PME#	RSM_PWROK	SUSSTAT#	USBP0N	USBP2N	OC1#	OC2#	AC_SDIN0	Y
FS0	THRM#	GPIO28	PCIRST#	SMBDATA	RI#	SUSCLK	USBP1N	USBP3N	VSS	VSS	AA
LAD3 / FWH3	LAD2 / FWH2	GPIO27	GPIO13	SMBCLK	SMBALERT# / GPIO11	SLP_S5#	USBP1P	USBP3P	VSS	VSS	AB
12	13	14	15	16	17	18	19	20	21	22	

VSS	VSS	VSS
VSS	VSS	VSS
VSS	VSS	VSS
VSS	VSS	VSS
VSS	VSS	VSS
VSS	VSS	VSS

Table 15-1. ICH2 EBGA Ball List

Signal Name	Ball Number
SDD03	A20
A20GATE	C13
A20M#	D11
AC_BITCLK	R19
AC_RST#	V22
AC_SDIN0	Y22
AC_SDIN1	W22
AC_SDOUT	P21
AC_SYNC	P19
AD0	AA4
AD1	AB4
AD2	Y4
AD3	W5
AD4	W4
AD5	Y5
AD6	AB3
AD7	AA5
AD8	AB5
AD9	Y3
AD10	W6
AD11	W3
AD12	Y6
AD13	Y2
AD14	AA6
AD15	Y1
AD16	V2
AD17	AA8
AD18	V1
AD19	AB8
AD20	U4
AD21	W9
AD22	U3
AD23	Y9
AD24	U2
AD25	AB9
AD26	U1
AD27	W10
AD28	T4

Table 15-1. ICH2 EBGA Ball List

Signal Name	Ball Number
AD29	Y10
AD30	T3
AD31	AA10
APICCLK	N20
APICD0	P22
APICD1	N19
C/BE0#	AA3
C/BE1#	AB6
C/BE2#	Y8
C/BE3#	AA9
CLK14	M19
CLK48	P20
CLK66	D4
CPUPWRGD	A13
CPUSLP#	A12
DEVSEL#	AB7
EE_CS	K4
EE_DIN	K3
EE_DOUT	J4
EE_SHCLK	J3
FERR#	R22
FRAME#	V3
FS0	AA12
GNT0#	M2
GNT1#	M1
GNT2#	R4
GNT3#	T2
GNT4#	R1
GNTA# / GPIO16	L2
GNTB# / GNT5# / GPIO17	L4
GPIO06	Y11
GPIO07	AA11
GPIO08	Y14
GPIO12	W14
GPIO13	AB15
GPIO18	A15
GPIO19	D14
GPIO20	C14
GPIO21	L1

Table 15-1. ICH2 EPGA Ball List

Signal Name	Ball Number
GPIO22	B14
GPIO23	A14
GPIO24	V21
GPIO25	W15
GPIO27	AB14
GPIO28	AA14
HL_STB	A6
HL_STB#	A7
HL00	A4
HL01	B5
HL02	A5
HL03	B6
HL04	B7
HL05	A8
HL06	B8
HL07	A9
HL08	C8
HL09	C6
HL10	C7
HL11	C5
HLCOMP	A3
HUBREF	B4
IGNNE#	A11
INIT#	C12
INTR	C11
INTRUDER#	T19
IRDY#	W8
IRQ14	F21
IRQ15	C16
LAD0 / FWH0	Y12
LAD1 / FWH1	W12
LAD2 / FWH2	AB13
LAD3 / FWH3	AB12
LAN_CLK	G3
LAN_RSTSYNC	H2
LAN_RXD0	G2
LAN_RXD1	G1
LAN_RXD2	H1
LAN_TXD0	F3

Table 15-1. ICH2 EPGA Ball List

Signal Name	Ball Number
LAN_TXD1	F2
LAN_TXD2	F1
LDRQ0#	Y13
LDRQ1#	W13
LFRAME# / FWH4	AB11
NMI	B11
OC0#	W19
OC1#	Y20
OC2#	Y21
OC3#	W20
PAR	W2
PCICLK	W11
PCIRST#	AA15
PDA0	F20
PDA1	F19
PDA2	E22
PDCS1#	E21
PDCS3#	E19
PDD00	H19
PDD01	H22
PDD02	J19
PDD03	J22
PDD04	K21
PDD05	L20
PDD06	M21
PDD07	M22
PDD08	L22
PDD09	L21
PDD10	K22
PDD11	K20
PDD12	J21
PDD13	J20
PDD14	H21
PDD15	H20
PDDACK#	F22
PDDREQ	G22
PDIOR#	G19
PDIOW#	G21
PERR#	Y7

Table 15-1. ICH2 EBGa Ball List

Signal Name	Ball Number
PIORDY	G20
PIRQA#	P1
PIRQB#	P2
PIRQC#	P3
PIRQD#	N4
PIRQE#	N3
PIRQF# / GPIO3	N2
PIRQG# / GPIO4	N1
PIRQH#	M4
PLOCK#	AA7
PME#	Y15
PWRBTN#	W21
PWROK	R20
RCIN#	B13
REQ0#	R2
REQ1#	R3
REQ2#	T1
REQ3#	AB10
REQ4#	P4
REQA# / GPIO0	M3
REQB# / REQ5# / GPIO1	L3
RI#	AA17
RSM_PWROK	Y16
RSMRST#	R21
RTCRST#	T20
RTCX1	U22
RTCX2	T22
SDA0	A16
SDA1	D16
SDA2	B16
SDCS1#	C15
SDCS3#	D15
SDD00	D18
SDD01	B19
SDD02	D19
SDD04	C20
SDD05	C21
SDD06	D22
SDD07	E20

Table 15-1. ICH2 EBGa Ball List

Signal Name	Ball Number
SDD08	D21
SDD09	C22
SDD10	D20
SDD11	B20
SDD12	C19
SDD13	A19
SDD14	C18
SDD15	A18
SDDACK#	B17
SDDREQ	B18
SDIOR#	D17
SDIOW#	C17
SERIRQ	N21
SERR#	W7
SIORDY	A17
SLP_S3#	W16
SLP_S5#	AB18
SMBALERT# / GPIO11	AB17
SMBCLK	AB16
SMBDATA	AA16
SMI#	B12
SMLINK0	U19
SMLINK1	V20
SPKR	N22
STOP#	W1
STPCLK#	C10
SUSCLK	AA18
SUSSTAT#	Y17
THRM#	AA13
TP0	U20
TRDY#	V4
USBP0N	Y18
USBP0P	W17
USBP1N	AA19
USBP1P	AB19
USBP2N	Y19
USBP2P	W18
USBP3N	AA20
USBP3P	AB20

Table 15-1. ICH2 EPGA Ball List

Signal Name	Ball Number
V_CPU_IO	D12
V_CPU_IO	D13
V5REF_SUS	V19
V5REF1	K2
V5REF2	M20
VBIAS	T21
VCC1_8	D10
VCC1_8	D2
VCC1_8	K19
VCC1_8	L19
VCC1_8	P5
VCC1_8	V9
VCC1_8	E5
VCC3_3	E14
VCC3_3	E15
VCC3_3	E16
VCC3_3	E17
VCC3_3	E18
VCC3_3	F18
VCC3_3	G18
VCC3_3	H18
VCC3_3	J18
VCC3_3	P18
VCC3_3	R18
VCC3_3	R5
VCC3_3	T5
VCC3_3	U5
VCC3_3	V5
VCC3_3	V6
VCC3_3	V7
VCC3_3	V8
VCCRTC	U21
VCCSUS1_8	H5
VCCSUS1_8	J5
VCCSUS1_8	V14
VCCSUS1_8	V15
VCCSUS1_8	V16
VCCSUS3_3	F5
VCCSUS3_3	G5

Table 15-1. ICH2 EPGA Ball List

Signal Name	Ball Number
VCCSUS3_3	T18
VCCSUS3_3	U18
VCCSUS3_3	V17
VCCSUS3_3	V18
VRMPWRGD	B15
VSS	D7
VSS	D8
VSS	D9
VSS	E6
VSS	E7
VSS	E8
VSS	E9
VSS	J10
VSS	J11
VSS	J12
VSS	J13
VSS	J14
VSS	J9
VSS	K1
VSS	K10
VSS	K11
VSS	K12
VSS	K13
VSS	K14
VSS	K9
VSS	L10
VSS	L11
VSS	A1
VSS	A10
VSS	A2
VSS	A21
VSS	A22
VSS	AA1
VSS	AA2
VSS	AA21
VSS	AA22
VSS	AB1
VSS	AB2
VSS	AB21

Table 15-1. ICH2 EPGA Ball List

Signal Name	Ball Number
VSS	AB22
VSS	B1
VSS	B10
VSS	B2
VSS	B21
VSS	B22
VSS	B3
VSS	B9
VSS	C2
VSS	C3
VSS	C4
VSS	C9
VSS	D3
VSS	D5
VSS	D6
VSS	L12
VSS	L13
VSS	L14
VSS	L9

Table 15-1. ICH2 EPGA Ball List

Signal Name	Ball Number
VSS	M10
VSS	M11
VSS	M12
VSS	M13
VSS	M14
VSS	M9
VSS	N10
VSS	N11
VSS	N12
VSS	N13
VSS	N14
VSS	N9
VSS	P10
VSS	P11
VSS	P12
VSS	P13
VSS	P14
VSS	P9

## 15.2 Package Information

Figure 15-3 and Figure 15-4 illustrate the ICH2 360 EPGA package.

Figure 15-3. ICH2 Package (Top and Side Views)

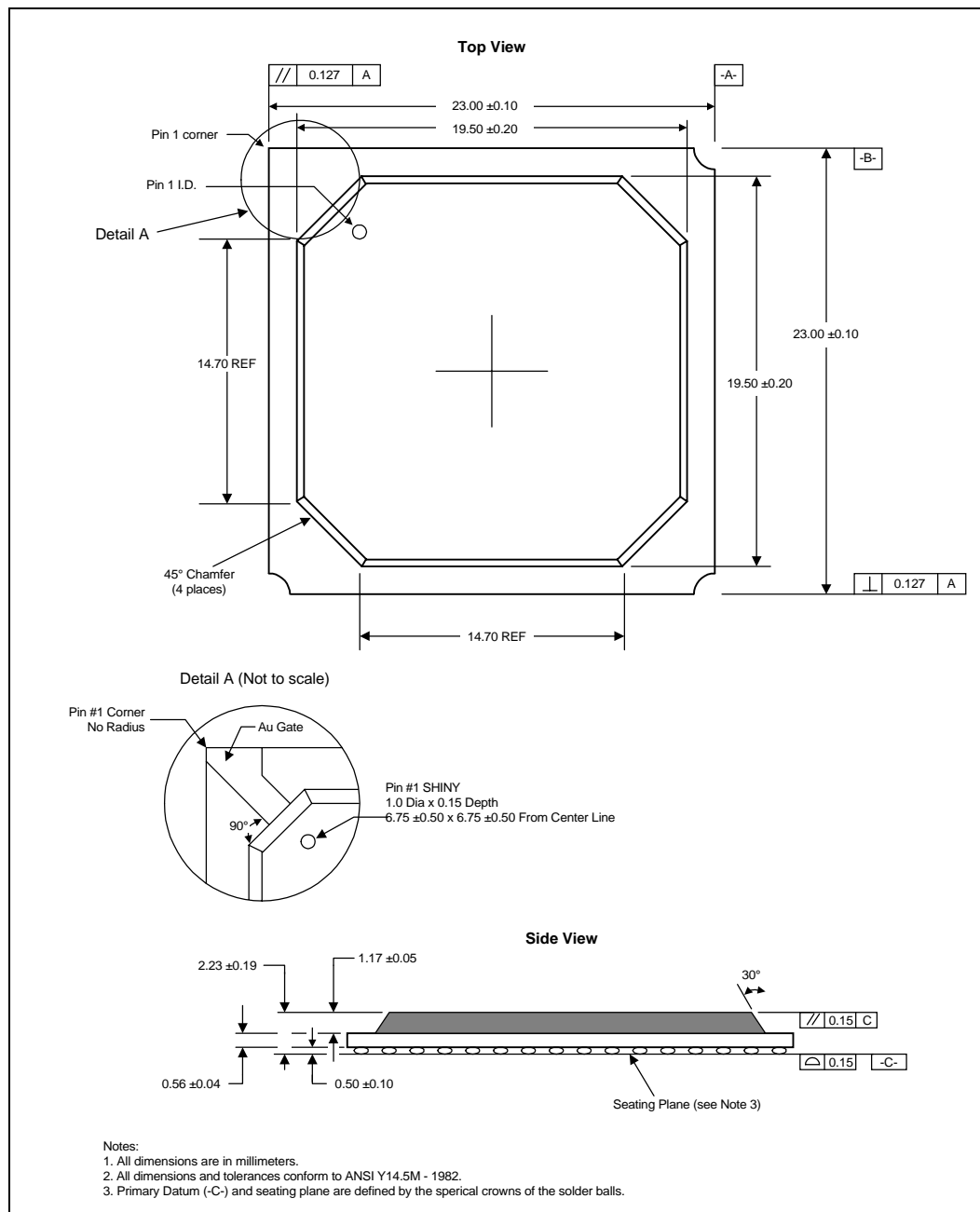
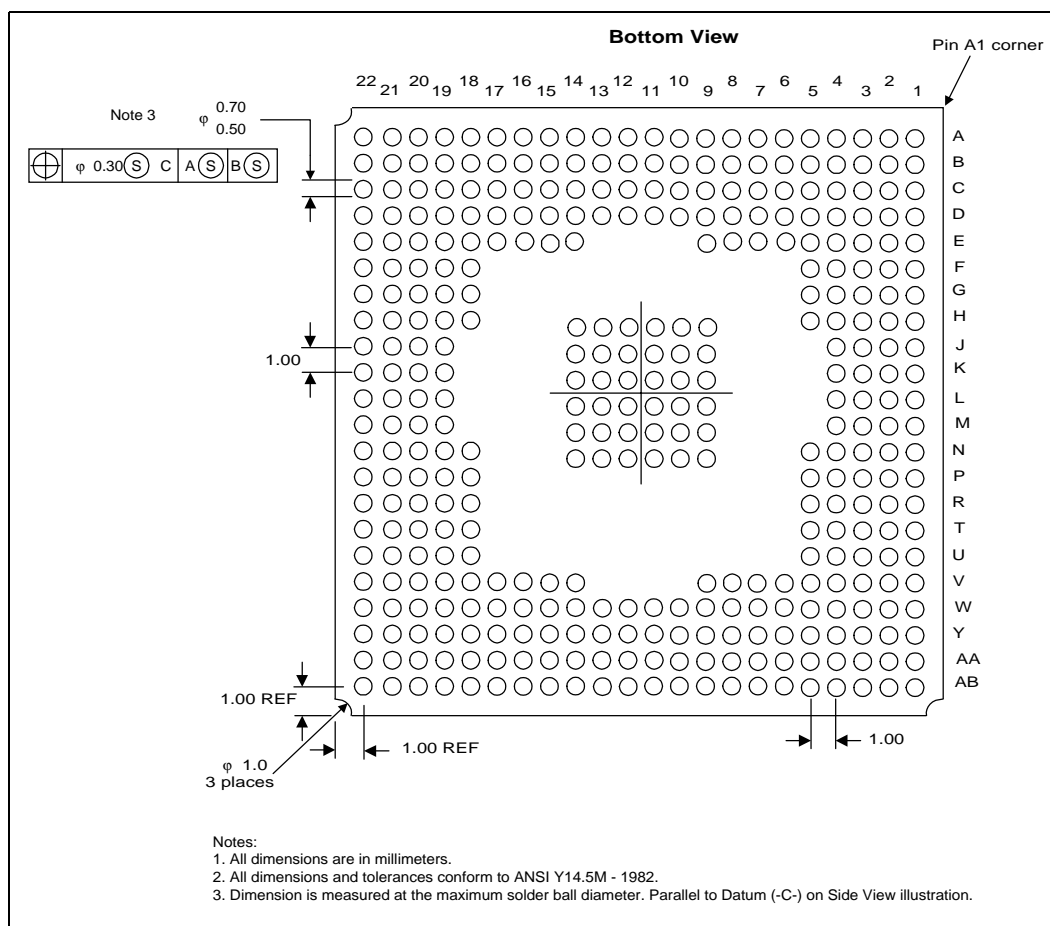


Figure 15-4. ICH2 Package (Bottom View)



# Electrical Characteristics

# 16

**Note:** The data provided in this chapter regarding the Electrical Characteristics of the ICH2 component are preliminary and subject to change.

## 16.1 Absolute Maximum Ratings

Case Temperature under Bias .....	0°C to +109°C
Storage Temperature .....	-55°C to +150°C
Voltage on Any 3.3V Pin with Respect to Ground .....	-0.5 to V <sub>cc</sub> + 0.3 V
Voltage on Any 5V Tolerant Pin with Respect to Ground (V <sub>REF</sub> =5V).....	-0.5 to V <sub>REF</sub> + 0.3 V
1.8V Supply Voltage with Respect to V <sub>ss</sub> .....	-0.5 to +2.7V
3.3V Supply Voltage with Respect to V <sub>ss</sub> .....	-0.5 to +4.6 V
5.0V Supply Voltage (V <sub>ref</sub> ) with Respect to V <sub>ss</sub> .....	-0.5 to +5.5 V
Maximum Power Dissipation .....	2.0 W

**Warning:** Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. See [Section 16.2](#) for the Functional Operating Range of the ICH2.

## 16.2 Functional Operating Range

All of the AC and DC Characteristics specified in this document assume that the ICH2 component is operating within the Functional Operating Range given in this section. Operation outside of the Functional Operating Range is not recommended, and extended exposure outside of the Functional Operating Range may affect component reliability.

Case Temperature under Bias .....	0°C to +109°C
1.8V Supply Voltage (V <sub>CC1_8</sub> ) with respect to V <sub>ss</sub> .....	1.7V to 1.9V
1.8V Supply Voltage (V <sub>ccSus1_8</sub> ) with respect to V <sub>ss</sub> .....	1.6V to 1.9V
3.3V Supply Voltage (V <sub>CC3_3</sub> , V <sub>ccSus3_3</sub> ) with respect to V <sub>ss</sub> .....	3.102V to 3.498V
5V Supply Voltage (V <sub>5REF</sub> , V <sub>5REF_Sus</sub> ) with respect to V <sub>ss</sub> .....	4.75V to 5.25V
V <sub>_CPU_IO</sub> Voltage with respect to V <sub>ss</sub> .....	TBD

## 16.3 D.C. Characteristics

**Table 16-1. D.C. Current Characteristics (ICH2 Power Consumption Measurements)**

Power Plane	Maximum Power Consumption			
	S0	S1	S3	S4/S5
1.8V Core	300 ma	100 ma	N/A	N/A
3.3V I/O	410 ma	5 ma	N/A	N/A
1.8V LAN	36 ma	28 ma	7 ma	7 ma
3.3V LAN (Integrated LAN + 82562ET)	230 ma	210 ma	210 ma; 60 ma when 82562ET is placed in reduced power mode (50 MHz clk → 5 MHz)	
1.8V SUS	45 ma	30 ma	30 ma	30 ma
3.3V SUS	25 ma	0.6 ma	0.6 ma	0.6 ma

**Table 16-2. DC Characteristic Input Signal Association**

Symbol	Associated Signals
$V_{IH1}/V_{IL1}$ (5V Tolerant)	<b>PCI Signals:</b> AD[31:0], C/BE[3:0]#, DEVSEL#, FRAME#, IRDY#, TRDY#, STOP#, PAR, PERR#, PLOCK#, SERR#, REQ[4:0]# <b>PC/PCI Signals:</b> REQ[A]#/GPIO[0], REQB[#]/REQ[5]#/GPIO[1] <b>IDE Signals:</b> PDD[15:0], SDD[15:0], PDDREQ, PIORDY, SDDREQ, SIORDY <b>Interrupt Signals:</b> IRQ[15:14], SERIRQ, PIRQ[D:A]#, PIRQ[H]#, PIRQ[G:F]#/GPIO[4:3], PIRQ[E]# <b>Legacy Signals:</b> RCIN#, A20GATE <b>USB Signals:</b> OC[3:0]# <b>GPIO Signals:</b> GPIO[7:6]
$V_{IH2}/V_{IL2}$	<b>Clock Signals:</b> CLK66, CLK48, CLK14, LAN_CLK, PCICLK
$V_{IH3}/V_{IL3}$	<b>LPC/FWH Signals:</b> LDRQ[1:0]#, LAD[3:0]/FWH[3:0] <b>Power Management Signals:</b> PME#, PWRBTN#, RI#, RSM_PWROK, RTCRST#, THRM#, VRMPWRGD <b>System Management Signals:</b> SMBALERT#/GPIO[11] <b>EEPROM Signals:</b> EE_DIN <b>AC'97 Signals:</b> AC_BITCLK, AC_SDIN[1:0], AC_SYNC <b>GPIO Signals:</b> GPIO[25:24, 13:12, 8]
$V_{IH4}/V_{IL4}$	<b>Clock Signals:</b> APICCLK
$V_{IH5}/V_{IL5}$	<b>SMBus Signals:</b> SMBCLK, SMBDATA <b>System Management Signals:</b> INTRUDER#, SMLINK[1:0] <b>Power Management Signals:</b> RSMRST#, PWROK, <b>GPIO Signals:</b> GPIO[28:27]
$V_{IL6}/V_{IH6}$	<b>LAN Signals:</b> LAN_RXD[2:0]
$V_{IL7}/V_{IH7}$	<b>Processor Signals:</b> FERR#, APICD[1:0]
$V_{IL8}/V_{IH8}$	<b>Hub Interface Signals:</b> HL[11:0], HL_STB#, HL_STB
$V_{DI} / V_{CM} / V_{SE}$	<b>USB Signals:</b> USBP[1:0][P,N]
$V_{IL9}/V_{IH9}$	<b>RTCX1</b>

Table 16-3. DC Input Characteristics

Symbol	Parameter	Min.	Max	Unit	Notes
$V_{IL1}$	Input Low Voltage	-0.5	0.8	V	
$V_{IH1}$	Input High Voltage	2.0	$V_{5REF} + 0.5$	V	
$V_{IL2}$	Input Low Voltage	-0.5	0.8	V	
$V_{IH2}$	Input High Voltage	2.0	$V_{cc3\_3} + 0.5$	V	
$V_{IL3}$	Input Low Voltage	-0.5	$0.3V_{cc3\_3}$	V	
$V_{IH3}$	Input High Voltage	$0.5V_{cc3\_3}$	$V_{cc3\_3} + 0.5$	V	
$V_{IL4}$	Input Low Voltage	-0.5	0.7	V	
$V_{IH4}$	Input High Voltage	1.7	2.625	V	
$V_{IL5}$	Input Low Voltage	-0.5	0.6	V	
$V_{IH5}$	Input High Voltage	2.1	$V_{ccSus3\_3} + 0.5$	V	
$V_{IL6}$	Input Low Voltage	-0.5	$0.3V_{cc3\_3}$	V	
$V_{IH6}$	Input High Voltage	$0.6V_{cc3\_3}$	$V_{cc3\_3} + 0.5$	V	
$V_{IL7}$	Input Low Voltage	-0.5	0.6	V	
$V_{IH7}$	Input High Voltage	1.2	$V_{cc3\_3} + 0.5$	V	
$V_{IL8}$	Input Low Voltage	-0.5	$HUBREF - 0.15$	V	Normal Mode
			$HUBREF - 0.20$		Enhanced Mode
$V_{IH8}$	Input High Voltage	$HUBREF + 0.15$	$V_{cc1\_8} + 0.5$	V	Normal Mode
		$HUBREF + 0.20$			Enhanced Mode
$V_{DI}$	Differential Input Sensitivity	0.2		V	Note 1
$V_{CM}$	Differential Common Mode Range	0.8	2.5	V	Note 2
$V_{SE}$	Single-Ended Receiver Threshold	0.8	2.0	V	
$V_{IL9}$	Input Low Voltage	-0.5	0.10	V	
$V_{IH10}$	Input High Voltage	0.40	2.0	V	

**NOTES:**

- $V_{DI} = |USBx[P] - USBx[N]|$
- Includes  $V_{DI}$  range.

Table 16-4. DC Characteristic Output Signal Association

Symbol	Associated Signals
$V_{OH1}/V_{OL1}$	<b>IDE Signals:</b> 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]#
$V_{OH2}/V_{OL2}$	<b>Processor Signals:</b> A20M#, CPUPWRGD, CPUSLP#, IGNNE#, INIT#, INTR, NMI, SMI#, STPCLK#
$V_{OH3}/V_{OL3}$	<b>PCI Signals:</b> AD[31:0], C/BE[3:0]#, PCIRST#, GNT[4:0]#, PAR, DEVSEL#, PERR#, PLOCK#, STOP#, TRDY#, IRDY#, FRAME#, SERR# <b>Interrupt Signals:</b> SERIRQ, PIRQ[D:A]#, PIRQ[H]#, PIRQ[G:F]#/GPIO[4:3], PIRQ[E]#
$V_{OH4}/V_{OL4}$	<b>PCI Signals:</b> GNT5#/GNTB#/GPIO17, GNTA#/GPIO16 <b>LPC/FWH Signals:</b> LAD[3:0]/FWH[3:0], LFRAME#/FWH[4] <b>AC'97 Signals:</b> AC_RST#, AC_SDOOUT, AC_SYNC <b>LAN Signals:</b> LAN_RSTSYNC, LAN_TXD[2:0] <b>Power Management Signals:</b> PME# <b>GPIO Signals:</b> GPIO[21]
$V_{OL5}/V_{OH5}$	<b>SMBus Signals:</b> SMBCLK, SMBDATA <b>System Management Signals:</b> SMLINK[1:0] <b>Interrupt Signals:</b> APICD[1:0]
$V_{OL6}/V_{OH6}$	<b>Power Management Signals:</b> SLP_S3#, SLP_S5#, SUS_STAT#, SUSCLK <b>EEPROM Signals:</b> EE_CS, EE_DOUT, EE_SHCLK <b>GPIO Signals:</b> GPIO[28:27, 25:22, 20:18] <b>Other Signals:</b> SPKR
$V_{OL7}/V_{OH7}$	<b>USB Signals:</b> USBPO[P:N], USBP1[P:N]
$V_{OL8}/V_{OH8}$	<b>Hub Signals:</b> HL[11:0], HL_STB#, HL_STB

Table 16-5. DC Output Characteristics

Symbol	Parameter	Min.	Max	Unit	$I_{OL} / I_{OH}$	Notes
$V_{OL1}$	Output Low Voltage		0.5	V	4 mA	
$V_{OH1}$	Output High Voltage	2.4		V	-0.4 mA	
$V_{OL2}$	Output Low Voltage		0.4	V	4.0 mA	
$V_{OH2}$	Output High Voltage	$V_{CPU\_IO} - 0.13V$		V	-0.5 mA	Note 1
$V_{OL3}$	Output Low Voltage		0.55	V	6 mA	
$V_{OH3}$	Output High Voltage	2.4		V	-2 mA	Note 1
$V_{OL4}$	Output Low Voltage		0.1V <sub>cc</sub>	V	1.5 mA	
$V_{OH4}$	Output High Voltage	0.9V <sub>cc</sub>		V	-0.5 mA	Note 1
$V_{OL5}$	Output Low Voltage		0.4	V	3.0 mA	
$V_{OH5}$	Output High Voltage	N/A		V		Note 1
$V_{OL6}$	Output Low Voltage		0.4	V	4.0 mA	
$V_{OH6}$	Output High Voltage	$V_{cc3\_3} - 0.5$		V	-2.0 mA	Note 1
$V_{OL7}$	Output Low Voltage		0.4	V	5 mA	
$V_{OH7}$	Output High Voltage	$V_{cc} - 0.5$		V	-2 mA	
$V_{OL8}$	Output Low Voltage		0.1(V <sub>cc1_8</sub> )	V	1 mA	Normal Mode
			0.8	V	20 mA	Enhanced Mode
$V_{OH8}$	Output High Voltage	0.9(V <sub>cc1_8</sub> )		V	-1 mA	Normal Mode
		1.6		V	-1.5 mA	Enhanced Mode

**NOTES:**

1. The CPUPWRGD, SERR#, PIRQ[A:H], PME#, GPIO22/CPUPERF, APIC[1:0], SMBDATA, SMBCLK and SMLINK[1:0] signals have an open drain driver, and the V<sub>OH</sub> specification does not apply. These signals must have external pull-up resistors.

Table 16-6. Other DC Characteristics

Symbol	Parameter	Min.	Max	Unit	Notes
V5REF	ICH2 Core Well Reference Voltage	4.75	5.25	V	
VCC3_3	I/O Buffer Voltage	3.102	3.498	V	
VCC1_8	Internal Logic Voltage	1.7	1.9	V	
HUBREF	Hub Interface Reference Voltage	0.48(V <sub>cc1.8</sub> )	0.52(V <sub>cc1.8</sub> )	V	Normal Mode
		0.64(V <sub>cc1.8</sub> )	0.70(V <sub>cc1.8</sub> )	V	Enhanced Mode
V5REF_Sus	Suspend Well Reference Voltage	4.75	5.25	V	
VccSus3_3	Suspend Well I/O Buffer Voltage	3.102	3.498	V	
VccSus1_8	Suspend Well Logic Voltage	1.6	1.9	V	
Vcc(RTC)	Battery Voltage	2.0	3.6	V	
V <sub>IT+</sub>	Hysteresis Input Rising Threshold	1.9		V	Applied to USBP[3:0][P:N]
V <sub>IT-</sub>	Hysteresis Input Falling Threshold		1.3	V	Applied to USBP[3:0][P:N]
V <sub>DI</sub>	Differential Input Sensitivity	0.2		V	(USBP <sub>x+</sub> , USBP <sub>x-</sub> )
V <sub>CM</sub>	Differential Common Mode Range	0.8	2.5	V	Includes V <sub>DI</sub>
V <sub>CRS</sub>	Output Signal Crossover Voltage	1.3	2.0	V	
V <sub>SE</sub>	Single Ended Rcvr Threshold	0.8	2.0	V	
I <sub>LI1</sub>	Input Leakage Current	-1.0	+1.0	uA	
I <sub>LI2</sub>	Hi-Z State Data Line Leakage	-10	+10	uA	(0 V < V <sub>IN</sub> < 3.3V)
I <sub>LI3</sub>	Input Leakage Current - Clock signals	-100	+100	uA	See Note
C <sub>IN</sub>	Input Capacitance - Hub interface		8	pF	F <sub>C</sub> = 1 MHz
	Input Capacitance - All Other		12	pF	
C <sub>OUT</sub>	Output Capacitance		12	pF	F <sub>C</sub> = 1 MHz
C <sub>I/O</sub>	I/O Capacitance		12	pF	F <sub>C</sub> = 1 MHz
C <sub>L</sub>	Crystal Load Capacitance	7.5	15	pF	

**NOTE:** Includes APICCLK, CLK14, CLK48, CLK66, LAN\_CLK and PCICLK

## 16.4 A.C. Characteristics

Table 16-7. Clock Timings

Sym	Parameter	Min	Max	Unit	Notes	Figure
<b>PCI Clock (PCICLK)</b>						
t1	Period	30	33.3	ns		16-2
t2	High Time	12		ns		16-2
t3	Low Time	12		ns		16-2
t4	Rise Time		3	ns		16-2
t5	Fall Time		3	ns		16-2
<b>Oscillator Clock (OSC)</b>						
t6	Period	67	70	ns		16-2
t7	High Time	20				16-2
t8	Low time	20		ns		16-2
<b>USB Clock (USBCLK)</b>						
f <sub>clk48</sub>	Operating Frequency	48		MHz		
t9	Frequency Tolerance		2500	ppm	1	
t10	High Time	7		ns		16-2
t11	Low time	7		ns		16-2
t12	Rise Time		1.2	ns		16-2
t13	Fall Time		1.2	ns		16-2
<b>Suspend Clock (SUSCLK)</b>						
f <sub>susclk</sub>	Operating Frequency	32		KHz	5	
t14	High time	10		us	5	16-2
t15	Low Time	10		us	5	16-2
<b>SMBus Clock (SMBCLK)</b>						
f <sub>smb</sub>	Operating Frequency	10	16	KHz		
t18	High time	4.0	50	us	2	16-17
t19	Low time	4.7		us		16-17
t20	Rise time		1000	ns		16-17
t21	Fall time		300	ns		16-17
<b>I/O APIC Clock (APICCLK)</b>						
f <sub>ioap</sub>	Operating Frequency	14.32	33.33	MHz		
t22	High time	12	36	ns		16-2
t23	Low time	12	36	ns		16-2
t24	Rise time	1.0	5.0	ns		16-2
t25	Fall time	1.0	5.0	ns		16-2

Table 16-7. Clock Timings (Continued)

Sym	Parameter	Min	Max	Unit	Notes	Figure
<b>AC'97 Clock (BITCLK)</b>						
f <sub>ac97</sub>	Operating Frequency	12.288				
t <sub>26</sub>	Output Jitter		750			
t <sub>27</sub>	High time	32.56	48.84	ns		16-2
t <sub>28</sub>	Low time	32.56	48.84	ns		16-2
t <sub>29</sub>	Rise time	2.0	6.0	ns	4	16-2
t <sub>30</sub>	Fall time	2.0	6.0	ns	4	16-2
<b>Hub Interface Clock</b>						
f <sub>hl</sub>	Operating Frequency	66				
t <sub>31</sub>	High time	6.0		ns		16-2
t <sub>32</sub>	Low time	6.0		ns		16-2
t <sub>33</sub>	Rise time	0.25	1.2	ns		16-2
t <sub>34</sub>	Fall time	0.25	1.2	ns		16-2
t <sub>35</sub>	CLK66 leads PCICLK	1.0	4.5		3	

**NOTES:**

1. The USBCLK is a 48 MHz that expects a 40/60% duty cycle.
2. The maximum high time (t<sub>18</sub> Max) provide a simple guaranteed method for devices to detect bus idle conditions.
3. This specification includes pin-to-pin skew from the clock generator as well as board skew.
4. BITCLK Rise and Fall times are measured from 10%VDD and 90%VDD.
5. SUSCLK duty cycle can range from 30% minimum to 70% maximum.

**Table 16-8. PCI Interface Timing**

Sym	Parameter	Min	Max	Units	Notes	Figure
t40	AD[31:0] Valid Delay	2	11	ns	Min: 0pF Max: 50pF	16-3
t41	AD[31:0] Setup Time to PCICLK Rising	7		ns		16-4
t42	AD[31:0] Hold Time from PCICLK Rising	0		ns		16-4
t43	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PAR, PERR#, PLOCK#, DEVSEL# Valid Delay from PCICLK Rising	2	11	ns	Min: 0pF Max: 50pF	16-3
t44	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PAR, PERR#, PLOCK#, IDSEL, DEVSEL# Output Enable Delay from PCICLK Rising	2		ns		16-7
t45	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, PERR#, PLOCK#, DEVSEL#, GNT[A:B]# Float Delay from PCICLK Rising	2	28	ns		16-5
t46	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, SERR#, PERR#, DEVSEL#, Setup Time to PCICLK Rising	7		ns		16-4
t47	C/BE[3:0]#, FRAME#, TRDY#, IRDY#, STOP#, SERR#, PERR#, DEVSEL#, REQ[A:B]# Hold Time from PCLKIN Rising	0		ns		16-4
t48	PCIRST# Low Pulse Width	1		ms		16-6
t49	GNT[A:B]#, GNT[5:0]# Valid Delay from PCICLK Rising	2	12	ns		
t50	REQ[A:B]#, REQ[5:0]# Setup Time to PCICLK Rising	12		ns		

Table 16-9. IDE PIO &amp; Multiword DMA Mode Timing

Sym	Parameter	Min	Max	Units	Notes	Figure
t60	PDIOR#/PDIOw#/SDIOR#/SDIOw# Active From CLK66 Rising	2	20	ns		16-8, 16-9
t61	PDIOR#/PDIOw#/SDIOR#/SDIOw# Inactive From CLK66 Rising	2	20	ns		16-8, 16-9
t62	PDA[2:0]/SDA[2:0] Valid Delay From CLK66 Rising	2	30	ns		16-8
t63	PDCS1#/SDCS1#, PDCS3#/SDCS3# Active From CLK66 Rising	2	30	ns		16-8
t64	PDCS1#/SDCS1#, PDCS3#/SDCS3# Inactive From CLK66 Rising	2	30	ns		16-8
t65	PDDACK#/SDDACK# Active From CLK66 Rising	2	20	ns		16-9
t66	PDDACK#/SDDACK# Inactive From CLK66 Rising	2	20	ns		
t67	PDDREQ/SDDREQ Setup Time to CLK66 Rising	7		ns		16-9
t68	PDDREQ/SDDREQ Hold From CLK66 Rising	7		ns		16-9
t69	PDD[15:0]/SDD[15:0] Valid Delay From CLK66 Rising	2	30	ns		16-8, 16-9
t70	PDD[15:0]/SDD[15:0] Setup Time to CLK66 Rising	10		ns		16-8, 16-9
t71	PDD[15:0]/SDD[15:0] Hold From CLK66 Rising	7		ns		16-8, 16-9
t72	PIORDY/SIORDY Setup Time to CLK66 Rising	7		ns	1	16-8
t73	PIORDY/SIORDY Hold From CLK66 Rising	7		ns	1	16-8
t74	PIORDY/SIORDY Inactive Pulse Width	48		ns		16-8
t75	PDIOR#/PDIOw#/SDIOR#/SDIOw# Pulse Width Low				2,3	16-8, 16-9
t76	PDIOR#/PDIOw#/SDIOR#/SDIOw# Pulse Width High				3,4	16-8, 16-9

**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 16-10. Ultra ATA Timing (Mode 0, Mode 1, Mode 2)**

Sym	Parameter (1)	Mode 0 (ns)		Mode 1 (ns)		Mode 2 (ns)		Figure
		Min	Max	Min	Max	Min	Max	
t80	Sustained Cycle Time (T2cyc <sub>typ</sub> )	240		160		120		
t81	Cycle Time (T <sub>cyc</sub> )	112		73		54		16-11
t82	Two Cycle Time (T2cyc)	230		154		115		16-11
t83	Data Setup Time (T <sub>ds</sub> )	15		10		7		16-11
t84	Data Hold Time (T <sub>dh</sub> )	5		5		5		16-11
t85	Data Valid Setup Time (T <sub>dvs</sub> )	70		48		30		16-11
t86	Data Valid Hold Time (T <sub>dvh</sub> )	6		6		6		16-11
t87	Limited Interlock Time (T <sub>li</sub> )	0	150	0	150	0	150	16-13
t88	Interlock Time w/ Minimum (T <sub>mli</sub> )	20		20		20		16-13
t89	Envelope Time (T <sub>env</sub> )	20	70	20	70	20	70	16-10
t90	Ready to Pause Time (T <sub>rp</sub> )	160		125		100		16-12
t91	DMACK setup/hold Time (T <sub>ack</sub> )	20		20		20		16-10, 16-13

**NOTE:**

1. The specification symbols in parentheses correspond to the Ultra ATA specification name.

**Table 16-11. Ultra ATA Timing (Mode 3, Mode 4, Mode 5)**

Sym	Parameter (1)	Mode 3 (ns)		Mode 4 (ns)		Mode 5 (ns)		Figure
		Min	Max	Min	Max	Min	Max	
t80	Sustained Cycle Time (T2cyc <sub>typ</sub> )	90		60		40		
t81	Cycle Time (T <sub>cyc</sub> ) <sup>(2)</sup>	39		25		16.8		16-11
t82	Two Cycle Time (T2cyc)	86		57		38		16-11
t83	Data Setup Time (T <sub>ds</sub> )	7		5		4.0		16-11
t84	Data Hold Time (T <sub>dh</sub> )	5		5		4.6		16-11
t85	Data Valid Setup Time (T <sub>dvs</sub> )	20		6		3.3		16-11
t86	Data Valid Hold Time (T <sub>dvh</sub> )	6		6		3.3		16-11
t87	Limited Interlock Time (T <sub>li</sub> )	0	100	0	100	0	75	16-13
t88	Interlock Time w/ Minimum (T <sub>mli</sub> )	20		20			20	16-13
t89	Envelope Time (T <sub>env</sub> )	20	55	20	55	20	50	16-10
t90	Ready to Pause Time (T <sub>rp</sub> )	100		100		85		16-12
t91	DMACK setup/hold Time (T <sub>ack</sub> )	20		20		20		16-10, 16-13

Table 16-12. Universal Serial Bus Timing

Sym	Parameter	Min	Max	Units	Notes	Fig
<b>Full Speed Source (Note 7)</b>						
t100	USBPx+, USBPx- Driver Rise Time	4	20	ns	1, C <sub>L</sub> = 50 pF	16-14
t101	USBPx+, USBPx- Driver Fall Time	4	20	ns	1, C <sub>L</sub> = 50 pF	16-14
t102	Source Differential Driver Jitter To Next Transition For Paired Transitions	-2 -1	2 1	ns ns	2, 3	16-15
t103	Source EOP Width	160	175	ns	4	16-16
t104	Differential to SE0 Transition Skew	-2	5	ns	5	
t105	Receiver Data Jitter Tolerance To Next Transition For Paired Transitions	-20 -10	20 10	ns ns	3	16-15
t106	EOP Width: Must reject as EOP EOP Width: Must accept as EOP	40 85		ns ns	4	16-16
t107	Differential to SE0 Transition Skew	-2	5	ns	5	
<b>Low Speed Source (Note 8)</b>						
t108	USBPx+, USBPx- Driver Rise Time	75	300	ns ns	1, 6 C <sub>L</sub> = 50 pF C <sub>L</sub> = 350 pF	16-14
t109	USBPx+, USBPx- Driver Fall Time	75	300	ns ns	1, 6 C <sub>L</sub> = 50 pF C <sub>L</sub> = 350 pF	16-14
t110	Source Differential Driver Jitter To Next Transition For Paired Transitions	-2 -1	2 1	ns ns	2, 3	16-15
t111	Source EOP Width	160	175	ns	4	16-16
t112	Differential to SE0 Transition Skew	-2	5	ns	5	
t113	Receiver Data Jitter Tolerance To Next Transition For Paired Transitions	-20 -10	20 10	ns ns	3	16-15
t114	EOP Width: Must reject as EOP EOP Width: Must accept as EOP	40 85		ns ns	4	16-16
t115	Differential to SE0 Transition Skew	-2	5	ns	5	

**NOTES:**

1. Driver output resistance under steady state drive is specified 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 16-13. IOAPIC Bus Timing**

Sym	Parameter	Min	Max	Units	Notes	Fig
t120	APICCD[1:0]# Valid Delay from APICCLK Rising	3.0	12.0	ns		16-3
t121	APICCD[1:0]# Setup Time to APICCLK Rising	8.5		ns		16-4
t122	APICCD[1:0]# Hold Time from APICCLK Rising	3.0		ns		16-4

**Table 16-14. SMBus Timing**

Sym	Parameter	Min	Max	Units	Notes	Fig
t130	Bus Tree Time Between Stop and Start Condition	4.7		us		16-17
t131	Hold Time after (repeated) Start Condition. After this period, the first clock is generated.	4.0		us		16-17
t132	Repeated Start Condition Setup Time	4.7		us		16-17
t133	Stop Condition Setup Time	4.0		us		16-17
t134	Data Hold Time	300		ns		16-17
t135	Data Setup Time	250		ns		16-17
t136	Device Time Out	25	35	ms	1	
t137	Cumulative Clock Low Extend Time (slave device)		25	ms	2	16-18
t138	Cumulative Clock Low Extend Time (master device)		10	ms	3	16-18

**NOTES:**

1. A device will time out 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.

**Table 16-15. AC'97 Timing**

Sym	Parameter	Min	Max	Units	Notes	Fig
t140	ACSDIN[0:1] Setup to Falling Edge of BITCLK	15		ns		
t141	ACSDIN[0:1] Hold from Falling Edge of BITCLK	5		ns		
t142	ACSYNC, ACSDOUT valid delay from rising edge of BITCLK		15	ns		16-3

**Table 16-16. LPC Timing**

Sym	Parameter	Min	Max	Units	Notes	Fig
t150	LAD[3:0] Valid Delay from PCICLK Rising	2	11	ns		16-3
t151	LAD[3:0] Output Enable Delay from PCICLK Rising	2		ns		16-7
t152	LAD[3:0] Float Delay from PCICLK Rising		28	ns		16-5
t153	LAD[3:0] Setup Time to PCICLK Rising	7		ns		16-4
t154	LAD[3:0] Hold Time from PCICLK Rising	0		ns		16-4
t155	LDRQ[1:0]# Setup Time to PCICLK Rising	12		ns		16-4
t156	LDRQ[1:0]# Hold Time from PCICLK Rising	0		ns		16-4
t157	LFRAME# Valid Delay from PCICLK Rising	2	12	ns		16-3

Table 16-17. Miscellaneous Timings

Sym	Parameter	Min	Max	Units	Notes	Fig
t160	SERIRQ Setup Time to PCICLK Rising	7		ns		16-4
t161	SERIRQ Hold Time from PCICLK Rising	0		ns		16-4
t162	RI#, EXTSMI#, GPI, USB Resume Pulse Width	2		RTCCLK		16-6
t163	SPKR Valid Delay from OSC Rising		200	ns		16-3
t164	SERR# Active to NMI Active		200	ns		
t165	IGNNE# Inactive from FERR# Inactive		230	ns		

Table 16-18. Power Sequencing and Reset Signal Timings

Sym	Parameter	Min	Max	Units	Notes	Fig
t170	VccRTC active to RTCRST# inactive	5	-	ms		16-18
t171	V5RefSus active to VccSus3_3, VccSus1_8 active	0	-	ms	1, 2	16-18
t172	VccRTC supply active to VccSus supplies active	0	-	ms	3	16-18
t173	VccSus supplies active to RSM_PWROK active, RSMRST# inactive	10	-	ms		16-18, 16-20
t174	V5Ref active to Vcc3_3, Vcc1_8 active	0	-	ms	1, 2	16-18
t175	VccSus supplies active to Vcc supplies active	0	-	ms	3	16-18
t176	Vcc supplies active to PWROK, VRMPWRGD active	10	-	ms		16-18, 16-20, 16-22
t177	PWROK, VRMPWRGD active to SUS_STAT# inactive	32	34	RTCCLK		16-18, 16-20, 16-22
t178	SUS_STAT# inactive to PCIRST# inactive	1	3	RTCCLK		16-18, 16-20, 16-22
t179	AC_RST# active low pulse width	1		us		
t180	AC_RST# inactive to BIT_CLK startup delay	162.8		ns		

**NOTES:**

1. The V5Ref supply must power up before or simultaneous with its associated 3.3V supply, and must power down simultaneous with or after the 3.3V supply. See [Section 2.20.4](#) for details.
2. The associated 3.3V and 1.8V supplies are assumed to power up or down together. The difference between the levels of the 3.3V and 1.8V supplies must never be greater than 2.0V.
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.

**Table 16-19. Power Management Timings**

Sym	Parameter	Min	Max	Units	Notes	Fig
t181	VccSus active to SLP_S3#, SLP_S5#, SUS_STAT# and PCIRST# active		50	ns		16-20
t182 t183	RSMRST# inactive to SUSCLK running, SLP_S3#, SLP_S5# inactive		110	ms	7	16-20
t184	Vcc active to STPCLK#, CPUSLP#, inactive, and processor Frequency Strap signals high		50	ns		16-20, 16-22
t185	PWROK and VRMPWRGD active to SUS_STAT# inactive and processor Frequency Straps latched to Strap Values	32	34	RTCCLK	1	16-20
t186	Processor Reset Complete to Frequency Strap signals unlatched from Strap Values	7	9	CLK66	2	16-20
t187	STPCLK# active to Stop Grant cycle	N/A	N/A		3	16-21, 16-22
t188	Stop Grant cycle to CPUSLP# active	60	63	PCICLK	4	16-22, 16-22
t189	S1 Wake Event to CPUSLP# inactive	1	25	PCICLK	4	16-21
t190	CPUSLP# inactive to STPCLK# inactive	204	237	us		16-21, 16-22
t192	CPUSLP# active to SUS_STAT# active	2	4	RTCCLK	1	16-22
t193	SUS_STAT# active to PCIRST# active	9	15	RTCCLK	1	16-22
t194	PCIRST# active to SLP_S3# active	1	2	RTCCLK	1	16-22
t195	SLP_S3# active to SLP_S5# active	1	2	RTCCLK	1, 6	16-22
t196	SLP_S3# active to PWROK, VRMPWRGD inactive	0		ms	5	16-22
t197	PWROK, VRMPWRGD inactive to Vcc supplies inactive	20		ns		16-22
t198	Wake Event to SLP_S3#, SLP_S5# inactive	1	20	RTCCLK	1	16-22
t204	Processor I/F signals latched prior to STPCLK# active	0	4	CLK66	2	16-23
t205	Break Event to STPCLK# inactive	30	3120	ns		16-23
t206	STPCLK# inactive to processor I/F signals unlatched	240	1880	ns		16-23

**NOTES:**

1. These transitions are clocked off the internal RTC. One RTC clock is approximately 32 us.
2. This transition is clocked off the 66 MHz CLK66. One CLK66 is approximately 15 ns.
3. The ICH2 STPCLK# assertion will trigger the processor to send a stop grant acknowledge cycle. The timing for this cycle getting to the ICH2 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 ICH2 has no maximum timing requirement for this transition. It is up to the system designer to determine if the SLP\_S3# 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# and SLP\_S5# are asserted together following timing t194 (PCIRST# active to SLP\_S3# and SLP\_S5# 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 1000 ms.

## 16.5 Timing Diagrams

Figure 16-1. Clock Timing

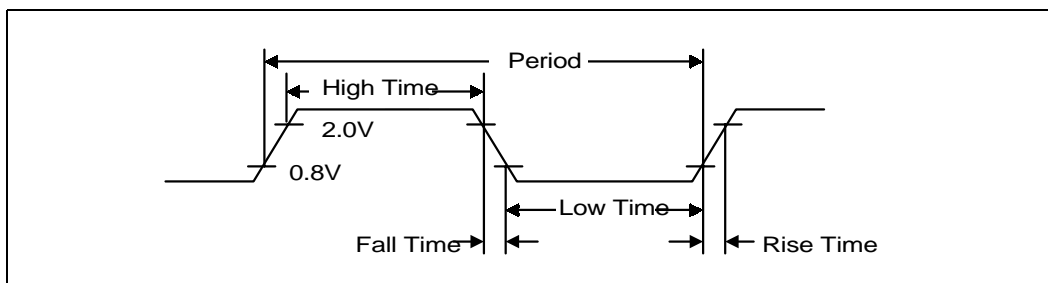


Figure 16-2. Valid Delay From Rising Clock Edge

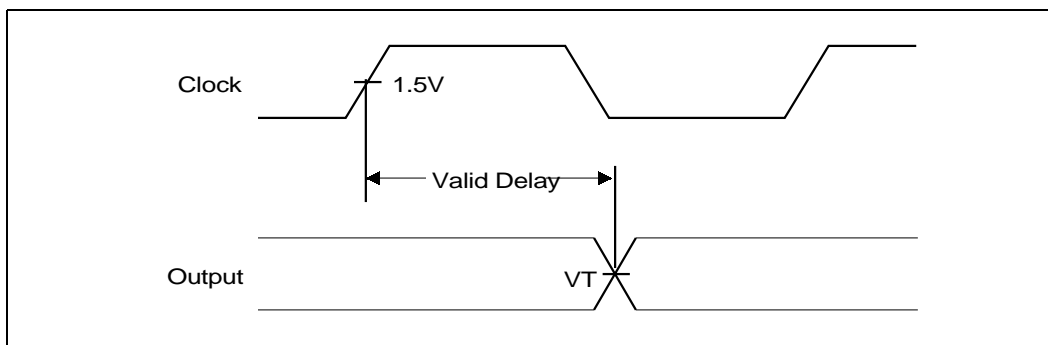


Figure 16-3. Setup And Hold Times

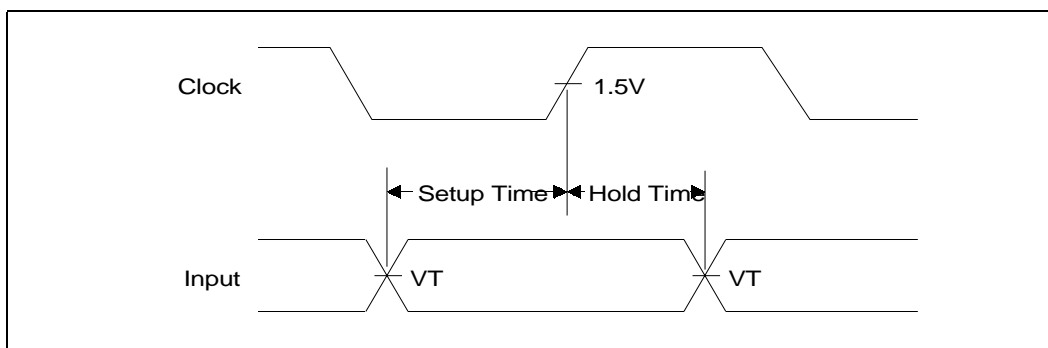


Figure 16-4. Float Delay

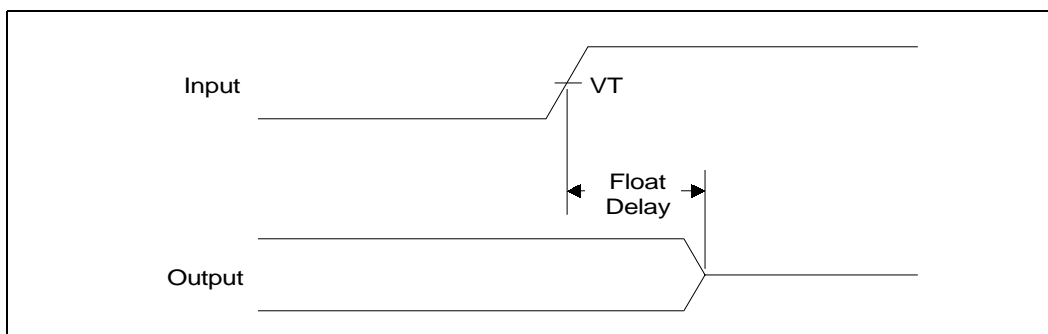


Figure 16-5. Pulse Width

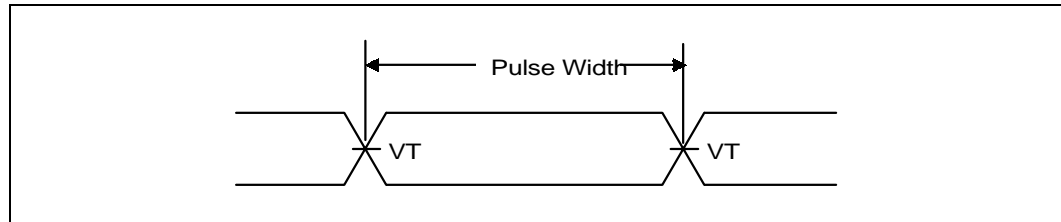


Figure 16-6. Output Enable Delay

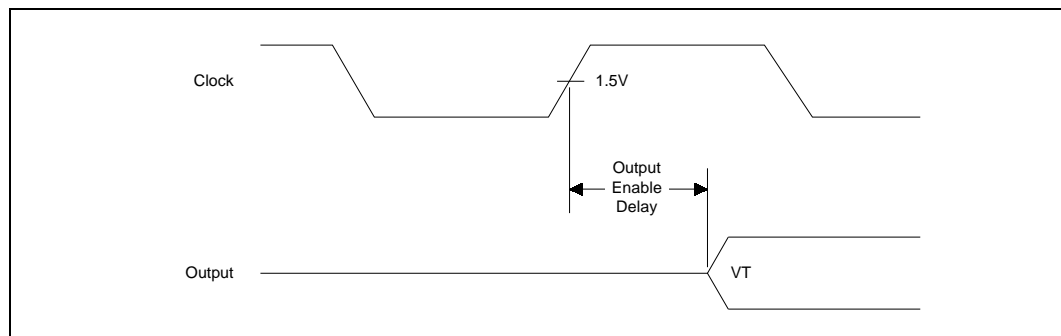


Figure 16-7. IDE PIO Mode

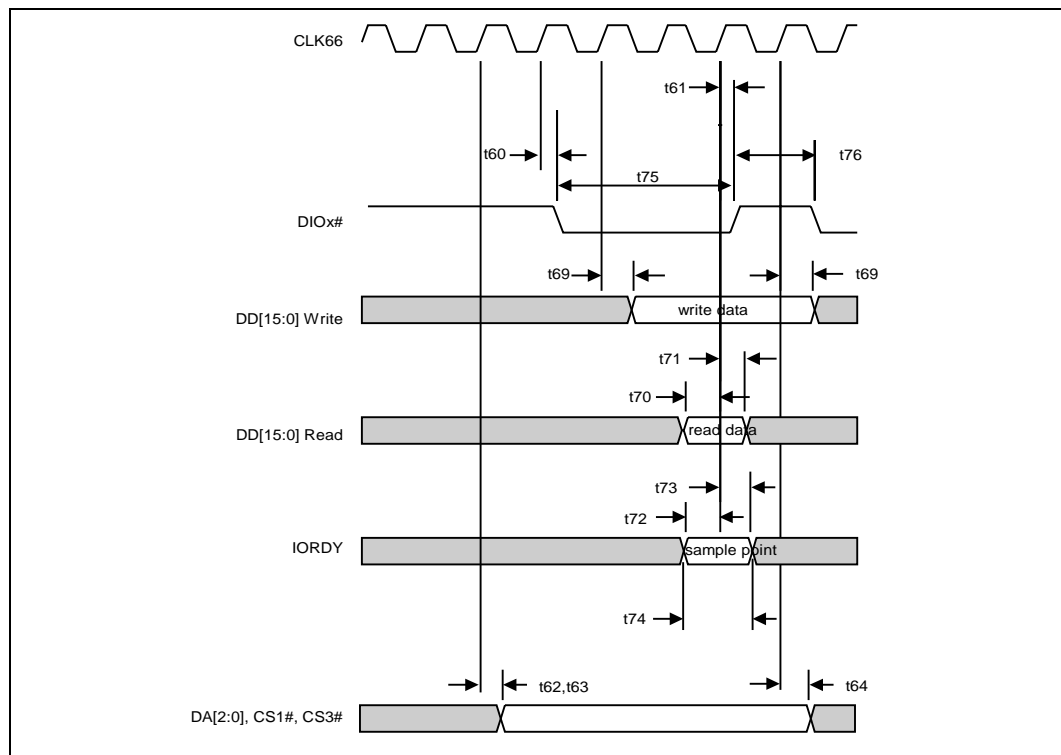


Figure 16-8. IDE Multiword DMA

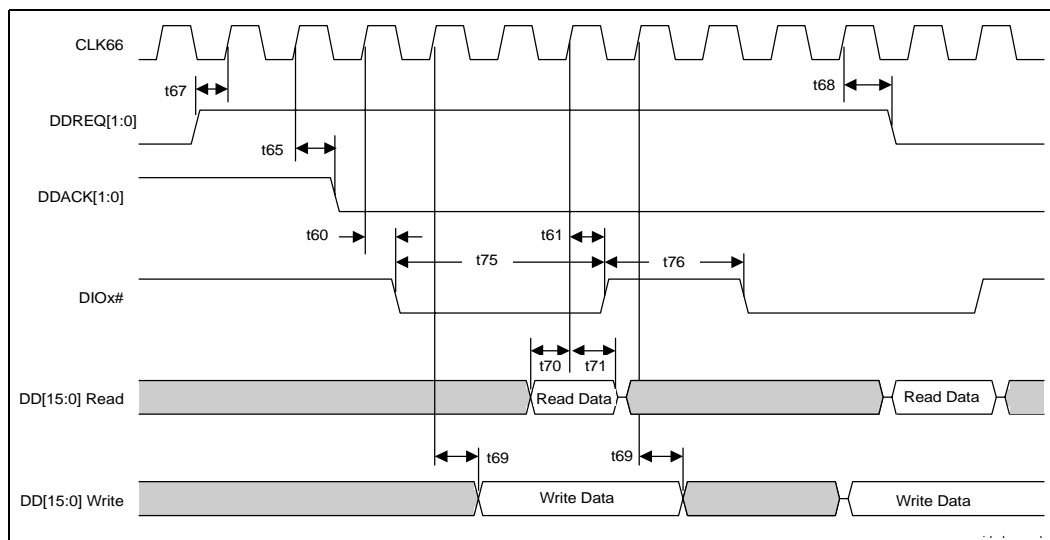


Figure 16-9. Ultra ATA Mode (Drive Initiating a Burst Read)

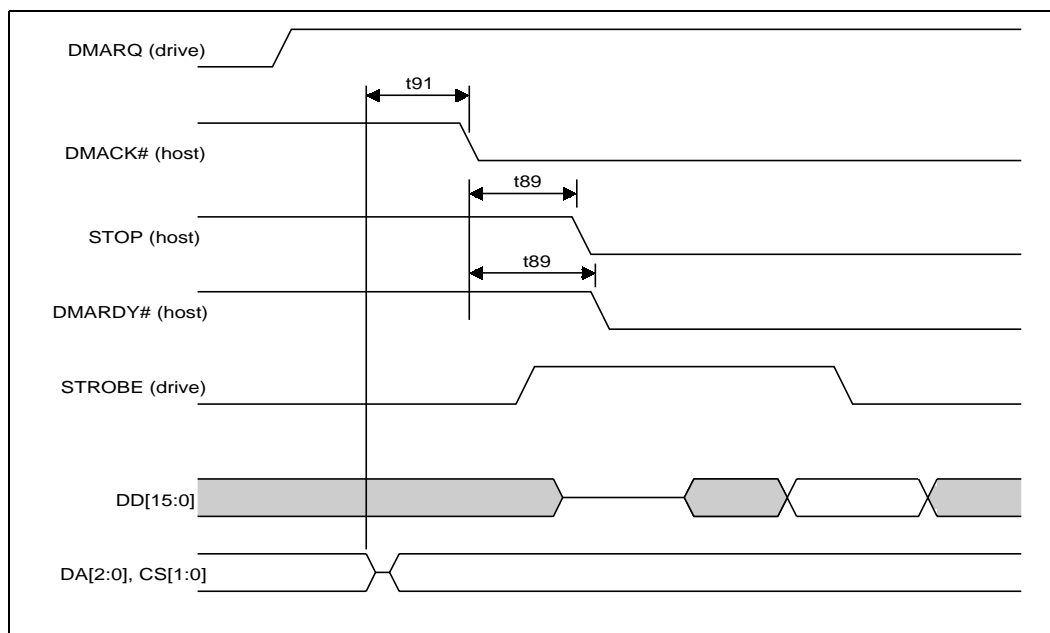


Figure 16-10. Ultra ATA Mode (Sustained Burst)

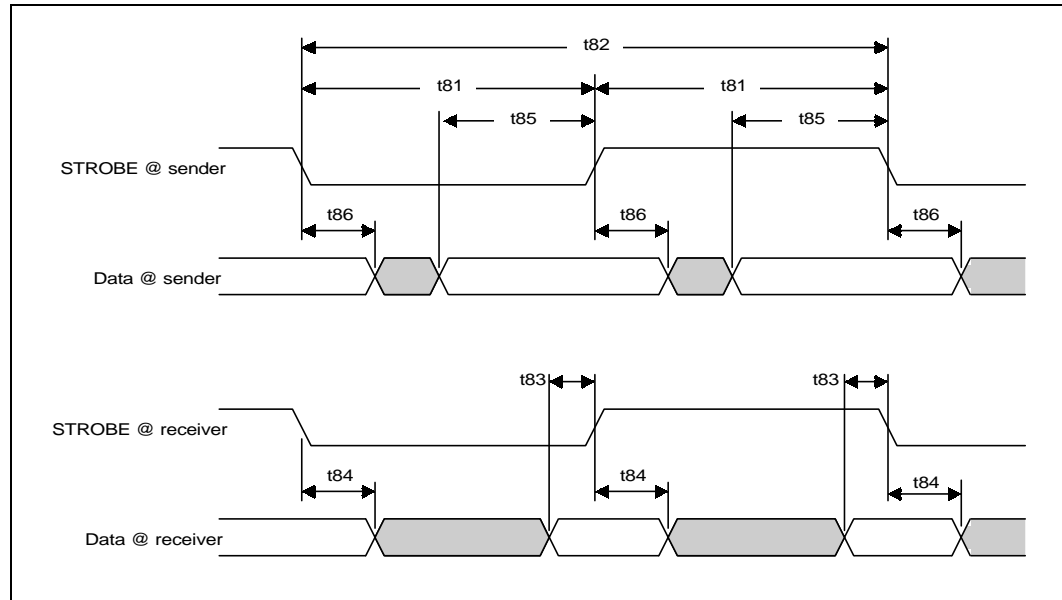


Figure 16-11. Ultra ATA Mode (Pausing a DMA Burst)

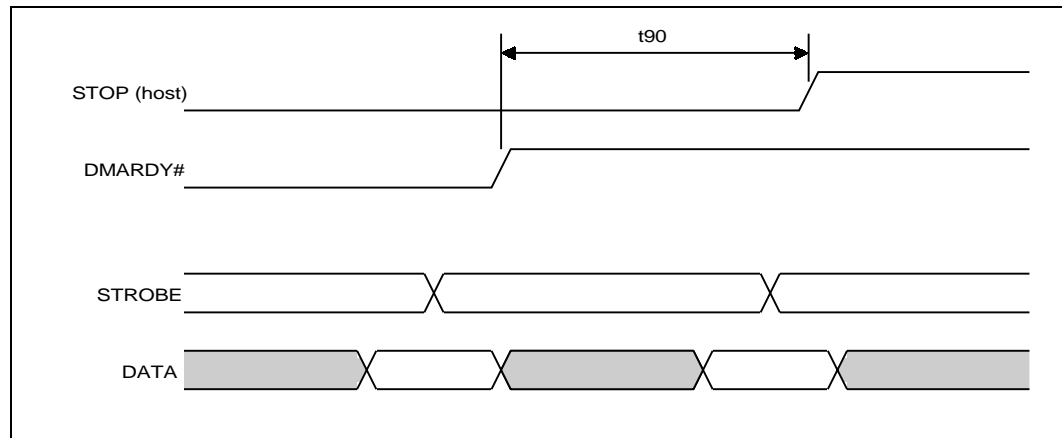


Figure 16-12. Ultra ATA Mode (Terminating a DMA Burst)

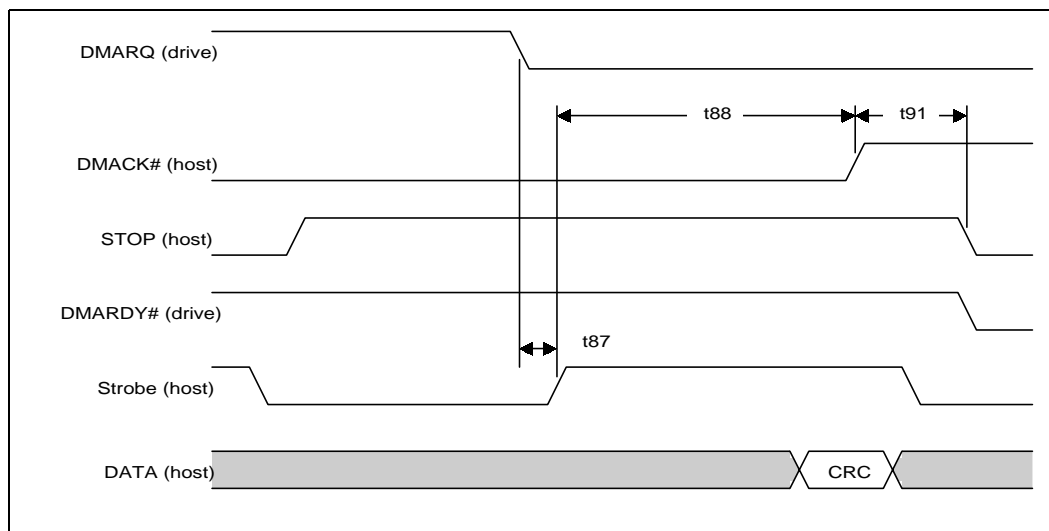


Figure 16-13. USB Rise and Fall Times

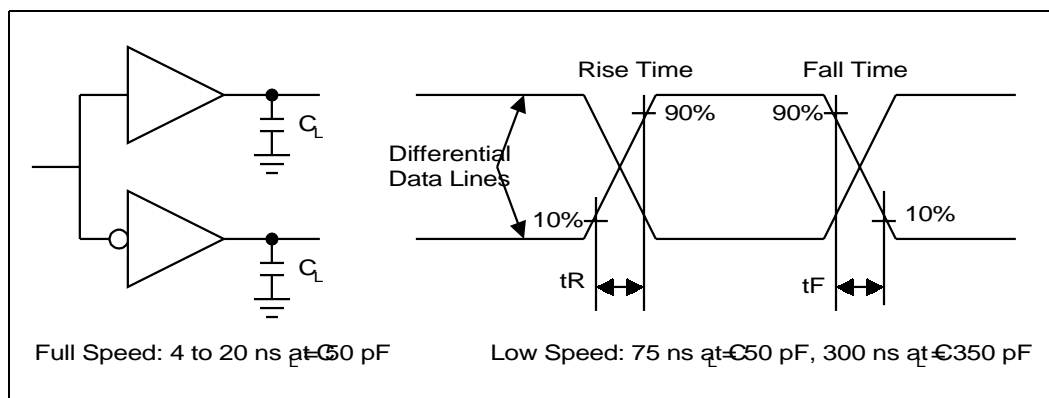


Figure 16-14. USB Jitter

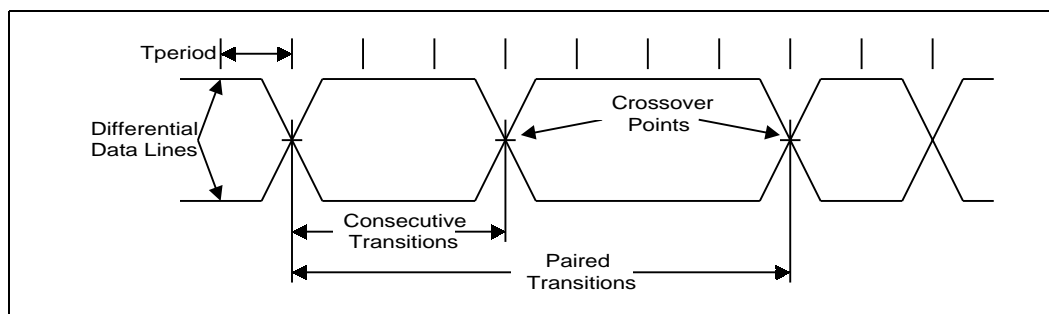


Figure 16-15. USB EOP Width

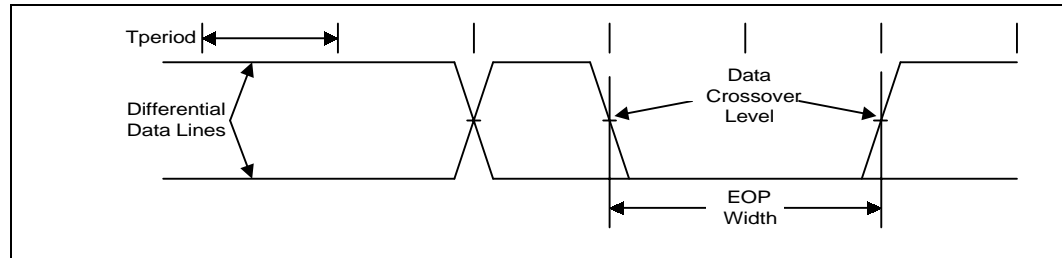


Figure 16-16. SMBus Transaction

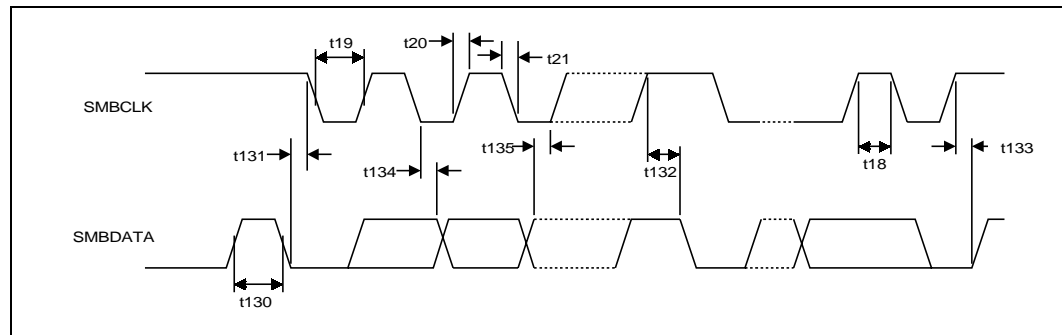


Figure 16-17. SMBus Time-out

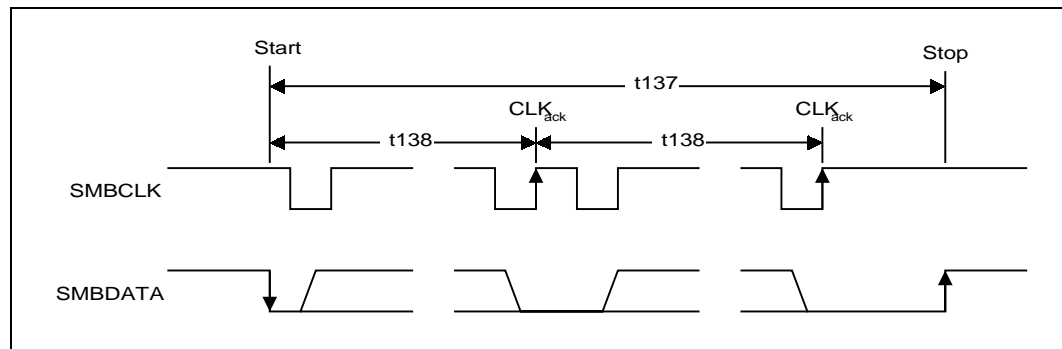


Figure 16-18. Power Sequencing and Reset Signal Timings

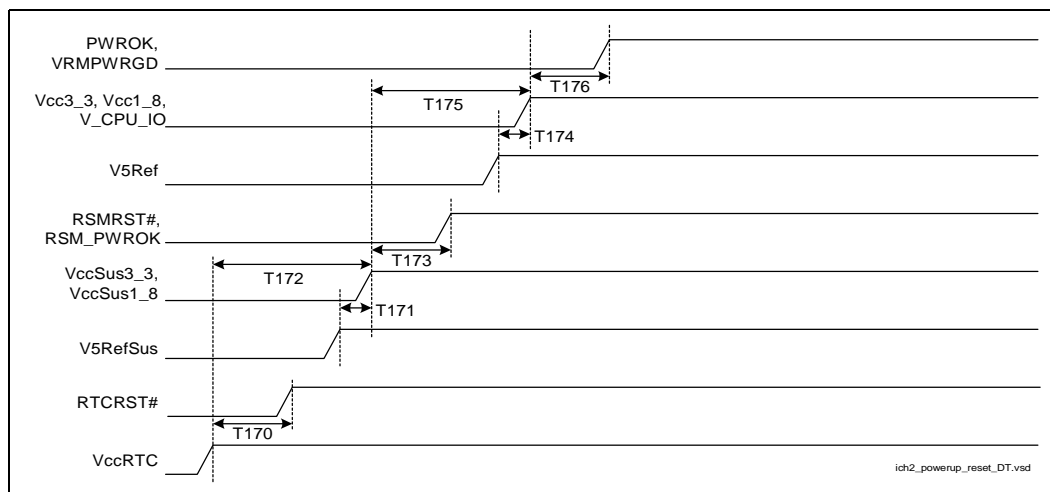


Figure 16-19. 1.8V/3.3V Power Sequencing

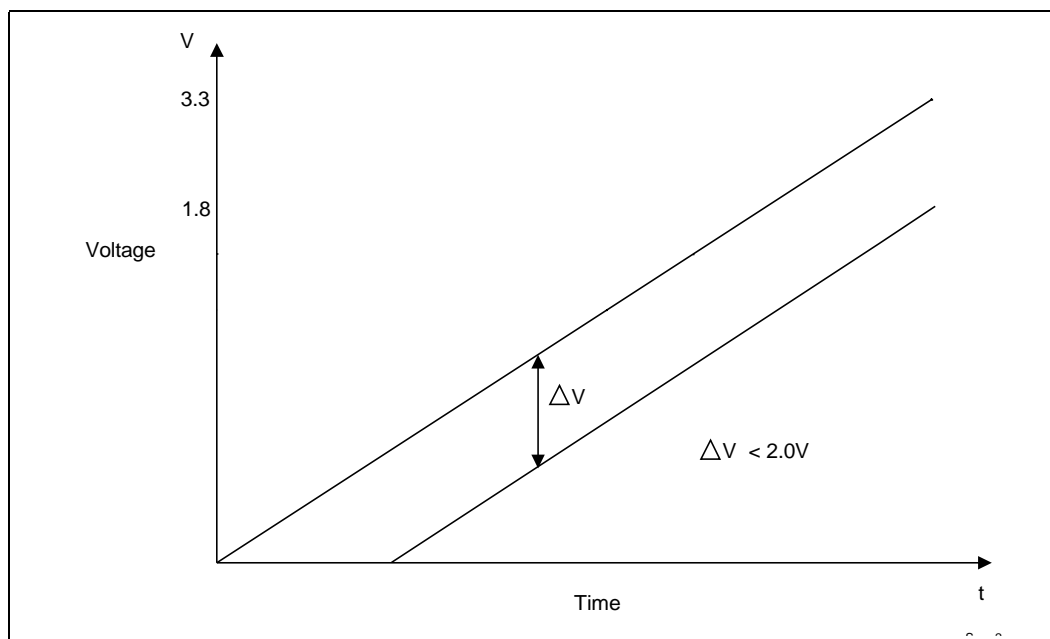


Figure 16-20. G3 (Mechanical Off) to S0 Timings

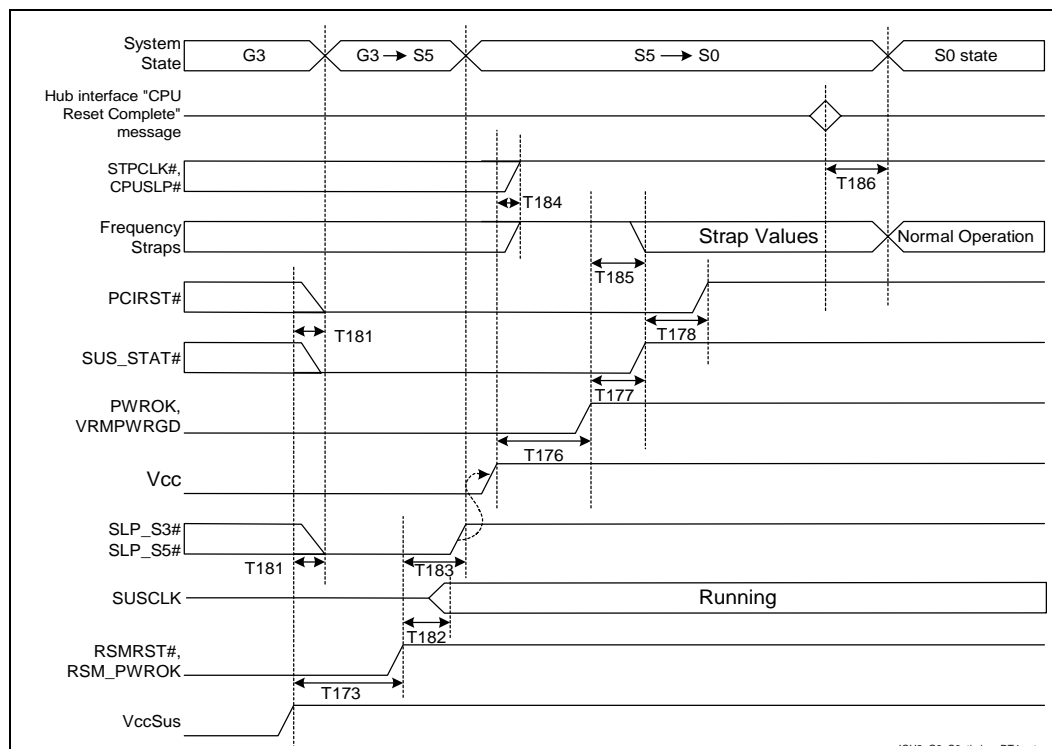


Figure 16-21. S0 to S1 to S0 Timings

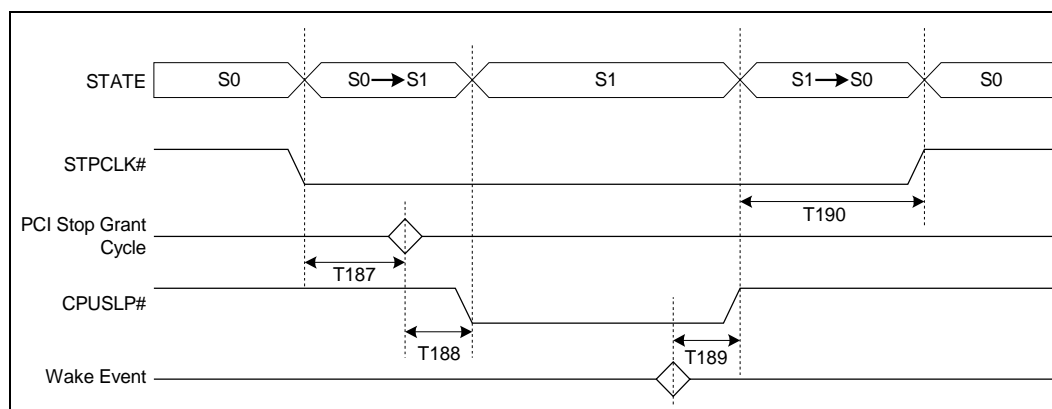


Figure 16-22. S0 to S5 to S0 Timings

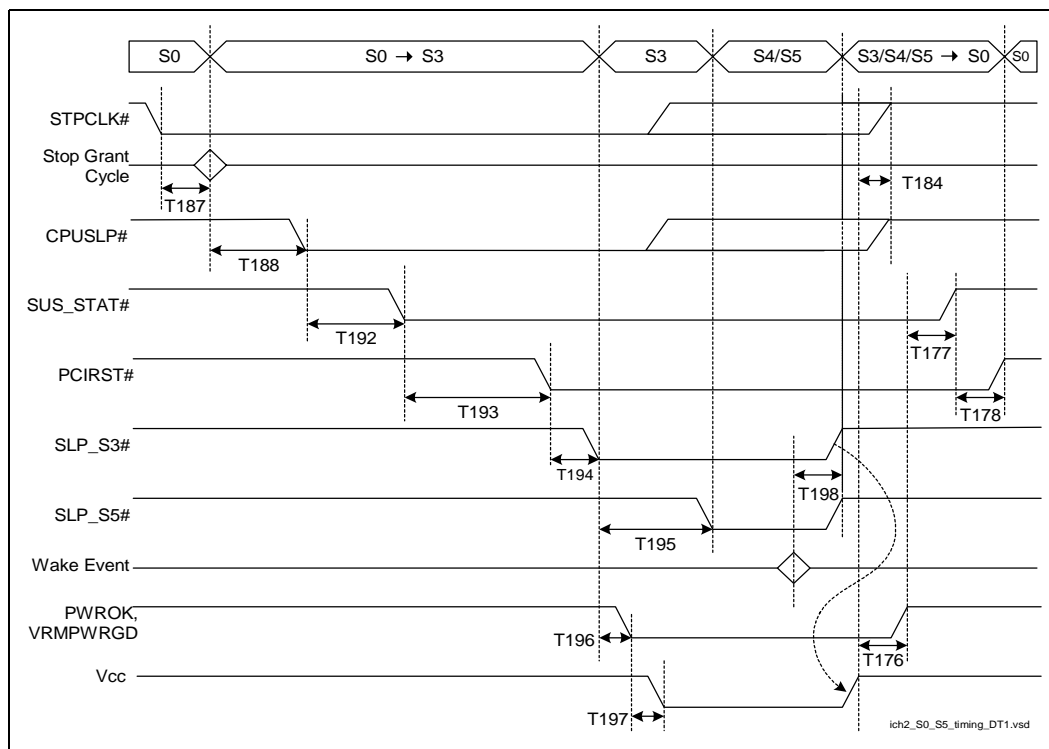
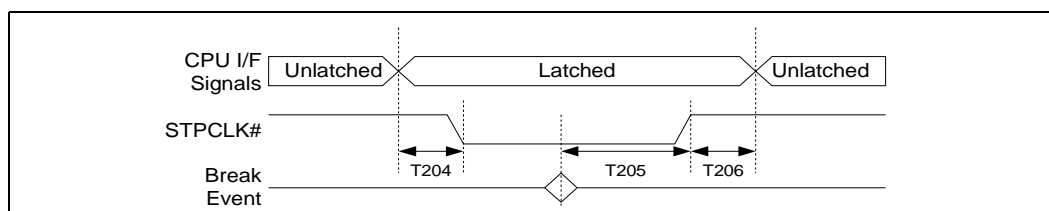


Figure 16-23. C0 to C2 to C0 Timings



# Testability

# 17

## 17.1 Test Mode Description

The ICH2 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 activates a particular test mode as described in [Table 17-1](#).

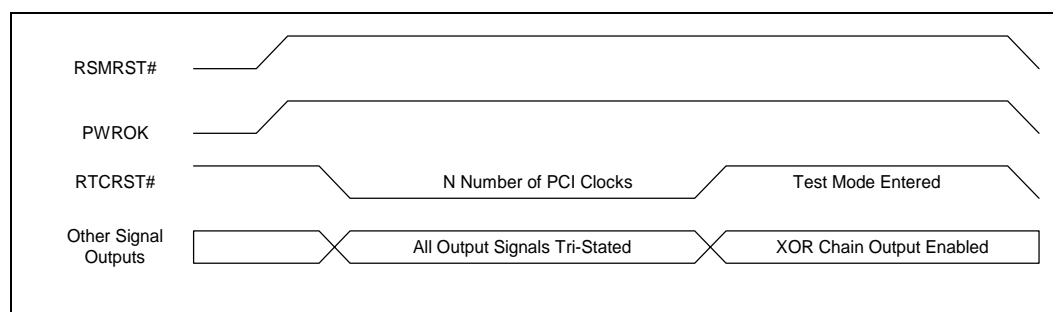
**Note:** RTCRST# can be driven low any time after PCIRST# is inactive.

**Table 17-1. 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 - 24	Reserved. DO NOT ATTEMPT
>24	No Test Mode Selected

[Figure 17-1](#) 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 ICH2 to normal operation, execute the sequence with RTCRST# being asserted so that no test mode is selected as specified in [Table 17-1](#).

**Figure 17-1. Test Mode Entry (XOR Chain Example)**



## 17.2 Tri-state Mode

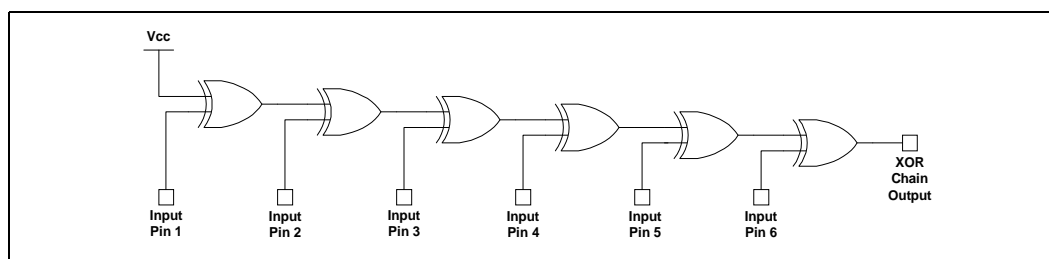
When in the tri-state mode, all outputs and bi-directional pin are tri-stated, including the XOR Chain outputs.

## 17.3 XOR Chain Mode

In the ICH2, provisions for Automated Test Equipment (ATE) board level testing are implemented with XOR Chains. The ICH2 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 17-2 is a schematic example of XOR chain circuitry.

Table 17-3 - Table 17-6 list each XOR chain pin ordering, with the first value being the first input and the last value being the XOR chain output. Table 17-7 lists the signal pins not included in any XOR chain.

Figure 17-2. Example XOR Chain Circuitry



### 17.3.1 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 17-2.

Table 17-2. 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 17-2](#), 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.

#### 17.3.1.1 Test Pattern Consideration for XOR Chain 4

When the ICH2 is operated with the Hub Interface in "Normal" mode (See [Section 2.20.1](#)), the HL\_STB and HL\_STB# signals must always be driven to complementary logic levels. For example, if a "1" is driven on HL\_STB, then a "0" must be driven on HL\_STB# and vice versa. This will need to be considered in applying test patterns to this chain.

When the ICH2 is operated with the Hub Interface in "Enhanced" mode there are no restrictions on the values that may be driven onto the HL\_STB and HL\_STB# signals.

Table 17-3. XOR Chain #1 (RTCRST# Asserted for 4 PCI Clocks while PWROK Active)

Pin Name	Ball #	Notes	Pin Name	Ball #	Notes
LAN_TXD0	F3	Top of XOR Chain	REQ2#	T1	
LAN_TXD1	F2	2nd signal in XOR	GNT2#	R4	
LAN_TXD2	F1		GNT3#	T2	
LAN_RXD0	G2		AD26	U1	
LAN_RXD1	G1		AD30	T3	
LAN_RXD2	H1		AD28	T4	
EE_DOUT	J4		AD18	V1	
EE_SHCLK	J3		AD22	U3	
EE_CS	K4		AD16	V2	
EE_DIN	K3		STOP#	W1	
GPIO21	L1		PAR	W2	
GPIO16 / GNTA#	L2		FRAME#	V3	
GPIO1 / REQB# / REQ5#	L3		AD20	U4	
GPIO17 / GNTB# / GNT5#	L4		AD15	Y1	
GNT1#	M1		TRDY#	V4	
GNT0#	M2		AD11	W3	
GPIO0 / REQA#	M3		AD13	Y2	
PIRQH#	M4		AD4	W4	
GPIO4 / PIRQG#	N1		AD9	Y3	
GPIO3 / PIRQF#	N2		C/BE0#	AA3	
PIRQE#	N3		AD2	Y4	
PIRQD#	N4		AD6	AB3	
PIRQA#	P1		AD3	W5	
PIRQB#	P2		AD0	AA4	
PIRQC#	P3		AD5	Y5	
REQ4#	P4		AD10	W6	
GNT4#	R1		AD7	AA5	Last in XOR Chain
REQ0#	R2				
REQ1#	R3				
			AC_SDIN1	W22	XOR Chain #1 OUTPUT

**Table 17-4. XOR Chain #2 (RTCRST# Asserted for 5 PCI clocks while PWROK Active)**

Pin Name	Ball #	Notes
AD1	AB4	Top of XOR Chain
AD12	Y6	2nd signal in XOR
AD8	AB5	
SERR#	W7	
AD14	AA6	
PERR#	Y7	
C/BE1#	AB6	
DEVSEL#	AB7	
PLOCK#	AA7	
C/BE2#	Y8	
IRDY#	W8	
AD17	AA8	
AD19	AB8	
AD23	Y9	
AD21	W9	
C/BE3#	AA9	
AD25	AB9	
AD27	W10	
AD29	Y10	
AD31	AA10	
REQ3#	AB10	
GPIO6	Y11	
GPIO7	AA11	
LFRAME# / FWH4	AB11	
LAD3 / FWH3	AB12	
FS0	AA12	
LAD0 / FWH0	Y12	
LAD1 / FWH1	W12	
LAD2 / FWH2	AB13	
THRM#	AA13	
LDRQ0#	Y13	

Pin Name	Ball #	Notes
LDRQ1#	W13	
GPIO27	AB14	
GPIO28	AA14	
GPIO8	Y14	
GPIO12	W14	
GPIO13	AB15	
PCIRST#	AA15	
PME#	Y15	
GPIO25	W15	
SMBCLK	AB16	
SMBDATA	AA16	
SMBALERT# / GPIO11	AB17	
RI#	AA17	
SLP_S5#	AB18	
SUSSTAT#	Y17	
SLP_S3#	W16	
SUSCLK	AA18	
USBP0P	W17	
USBP0N	Y18	
USBP1P	AB19	
USBP1N	AA19	
USBP2P	W18	
USBP2N	Y19	
USBP3P	AB20	
USBP3N	AA20	
OC0#	W19	
OC1#	Y20	
OC2#	Y21	
OC3#	W20	
<b>TP0</b>	U20	<b>XOR Chain #2 OUTPUT</b>

Table 17-5. XOR Chain #3 (RTCRST# Asserted for 6 PCI Clocks while PWROK Active)

Pin Name	Ball #	Notes	Pin name	Ball #	Notes
AC_SDIN0	Y22	Top of XOR Chain	PDD14	H21	
PWRBTN#	W21	2nd signal in XOR	PDD0	H19	
SMLINK0	U19		PDDREQ	G22	
SMLINK1	V20		PDIO#	G21	
AC_SDIN1	W22		PDD15	H20	
TP0	U20		PDDACK#	F22	
AC_RST#	V22		PDA2	E22	
GPIO24	V21		IRQ14	F21	
AC_SDOUT	P21		SDD6	D22	
AC_SYNC	P19		PIORDY	G20	
FERR#	R22		PDCS1#	E21	
APICD0	P22		PDIOR#	G19	
APICD1	N19		PDA0	F20	
SERIRQ	N21		SDD8	D21	
SPKR	N22		SDD9	C22	
PDD6	M21		PDA1	F19	
PDD7	M22		SDD7	E20	
PDD8	L22		SDD5	C21	
PDD9	L21		SDD10	D20	
PDD5	L20		SDD4	C20	
PDD10	K22		PDCS3#	E19	
PDD4	K21		SDD11	B20	
PDD11	K20		SDD2	D19	
PDD13	J20		SDD12	C19	
PDD3	J22		SDD3	A20	Last in XOR Chain
PDD12	J21		<b>XOR Chain #3</b>		
PDD1	H22		<b>R#</b>	AA17	<b>OUTPUT</b>
PDD2	J19				

**Table 17-6. XOR Chain #4 (RTCRST# Asserted for 7 PCI Clocks while PWROK Active)**

Pin Name	Ball #	Notes
SDD13	A19	Top of XOR Chain
SDD1	B19	2nd signal in XOR
SDD14	C18	
SDD0	D18	
SDIOR#	D17	
SDDREQ	B18	
SDIOW#	C17	
SDD15	A18	
SDA1	D16	
SDDACK#	B17	
IRQ15	C16	
SIORDY	A17	
SDA2	B16	
SDCS3#	D15	
SDA0	A16	
SDCS1#	C15	
VRMPWRGD	B15	
GPIO18	A15	
GPIO19	D14	
GPIO20	C14	
GPIO22	B14	
GPIO23	A14	
A20GATE	C13	
RCIN#	B13	
CPUPWRGD	A13	

Pin Name	Ball #	Notes
INIT#	C12	
SMI#	B12	
CPUSLP#	A12	
IGNNE#	A11	
NMI	B11	
INTR	C11	
A20M#	D11	
STPCLK#	C10	
HL7	A9	
HL5	A8	
HL6	B8	
HL4	B7	
HL8	C8	
HL10	C7	See <a href="#">Section 17.3.1.1</a>
HL_STB#	A7	See <a href="#">Section 17.3.1.1</a>
HL_STB	A6	
HL9	C6	
HL2	A5	
HL1	B5	
HL0	A4	
HL11	C5	
HLCOMP	A3	Last in XOR Chain
<b>XOR Chain #4</b>		
<b>OC0#</b>	<b>W19</b>	<b>OUTPUT</b>

Table 17-7. Signals Not in XOR Chain

Pin Name	Ball #	Notes
RSMRST#	R21	
PWROK	R20	
RTCX1	U22	
RTCX2	T22	
VBIAS	T21	
RTCRST#	T20	
LAN_CLK	G3	
AC_BITCLK	R19	

Pin Name	Ball #	Notes
CLK14	M19	
CLK48	P20	
CLK66	D4	
APICCLK	N20	
PCICLK	W11	
INTRUDER#	T19	
RSM_PWROK	Y16	

# I/O Register Index

# A

**Table A-1. ICH2 Fixed I/O Registers**

Register Name	Port	EDS Section and Location
Channel 0 DMA Base & Current Address Register	00h	Section 9.2.1, "DMABASE_CA—DMA Base and Current Address Registers" on page 9-24
Channel 0 DMA Base & Current Count Register	01h	Section 9.2.2, "DMABASE_CC—DMA Base and Current Count Registers" on page 9-25
Channel 1 DMA Base & Current Address Register	02h	Section 9.2.1, "DMABASE_CA—DMA Base and Current Address Registers" on page 9-24
Channel 1 DMA Base & Current Count Register	03h	Section 9.2.2, "DMABASE_CC—DMA Base and Current Count Registers" on page 9-25
Channel 2 DMA Base & Current Address Register	04h	Section 9.2.1, "DMABASE_CA—DMA Base and Current Address Registers" on page 9-24
Channel 2 DMA Base & Current Count Register	05h	Section 9.2.2, "DMABASE_CC—DMA Base and Current Count Registers" on page 9-25
Channel 3 DMA Base & Current Address Register	06h	Section 9.2.1, "DMABASE_CA—DMA Base and Current Address Registers" on page 9-24
Channel 3 DMA Base & Current Count Register	07h	Section 9.2.2, "DMABASE_CC—DMA Base and Current Count Registers" on page 9-25
Channel 0–3 DMA Command Register Channel 0–3 DMA Status Register	08h	Section 9.2.4, "DMACMD—DMA Command Register" on page 9-26 Section 9.2.5, "DMASTS—DMA Status Register" on page 9-26
Channel 0–3 DMA Write Single Mask Register	0Ah	Section 9.2.6, "DMA_WRSMSK—DMA Write Single Mask Register" on page 9-27
Channel 0–3 DMA Channel Mode Register	0Bh	Section 9.2.7, "DMACH_MODE—DMA Channel Mode Register" on page 9-27
Channel 0–3 DMA Clear Byte Pointer Register	0Ch	Section 9.2.8, "DMA Clear Byte Pointer Register" on page 9-28
Channel 0–3 DMA Master Clear Register	0Dh	Section 9.2.9, "DMA Master Clear Register" on page 9-28
Channel 0–3 DMA Clear Mask Register	0Eh	Section 9.2.10, "DMA_CLMSK—DMA Clear Mask Register" on page 9-28
Channel 0–3 DMA Write All Mask Register	0Fh	Section 9.2.11, "DMA_WRMSK—DMA Write All Mask Register" on page 9-29
<b>Aliased at 00h–0Fh</b>	10h–1Fh	
Master PIC ICW1 Init. Cmd Word 1 Register Master PIC OCW2 Op Ctrl Word 2 Register Master PIC OCW3 Op Ctrl Word 3 Register	20h	Section 9.4.2, "ICW1—Initialization Command Word 1 Register" on page 9-34 Section 9.4.8, "OCW2—Operational Control Word 2 Register" on page 9-37 Section 9.4.9, "OCW3—Operational Control Word 3 Register" on page 9-38

Table A-1. ICH2 Fixed I/O Registers (Continued)

Register Name	Port	EDS Section and Location
Master PIC ICW2 Init. Cmd Word 2 Register	21h	Section 9.4.3, "ICW2—Initialization Command Word 2 Register" on page 9-35
Master PIC ICW3 Init. Cmd Word 3 Register		Section 9.4.4, "ICW3—Master Controller Initialization Command Word 3 Register" on page 9-35
Master PIC ICW4 Init. Cmd Word 4 Register		Section 9.4.6, "ICW4—Initialization Command Word 4 Register" on page 9-36
Master PIC OCW1 Op Ctrl Word 1 Register		Section 9.4.7, "OCW1—Operational Control Word 1 (Interrupt Mask) Register" on page 9-36
<b>Aliased at 20h–21h</b>	24h–25h	
<b>Aliased at 20h–21h</b>	28h–29h	
<b>Aliased at 20h–21h</b>	24h–25h	
<b>Aliased at 20h–21h</b>	2Ch–2Dh	
<b>Aliased at 20h–21h</b>	30h–31h	
<b>Aliased at 20h–21h</b>	34h–35h	
<b>Aliased at 20h–21h</b>	38h–39h	
<b>Aliased at 20h–21h</b>	3Ch–3Dh	
Counter 0 Interval Time Status Byte Format	40h	Section 9.3.2, "SBYTE_FMT—Interval Timer Status Byte Format Register" on page 9-32
Counter 0 Counter Access Port Register		Section 9.3.3, "Counter Access Ports Register" on page 9-32
Counter 1 Interval Time Status Byte Format	41h	Section 9.3.2, "SBYTE_FMT—Interval Timer Status Byte Format Register" on page 9-32
Counter 1 Counter Access Port Register		Section 9.3.3, "Counter Access Ports Register" on page 9-32
Counter 2 Interval Time Status Byte Format	42h	Section 9.3.2, "SBYTE_FMT—Interval Timer Status Byte Format Register" on page 9-32
Counter 2 Counter Access Port Register		Section 9.3.3, "Counter Access Ports Register" on page 9-32
Timer Control Word Register	43h	Section 9.3.1, "TCW—Timer Control Word Register" on page 9-30
Timer Control Word Register Read Back		Section 9.3.1.1, "RDBK_CMD—Read Back Command" on page 9-31
Counter Latch Command		Section 9.3.1.2, "LTCH_CMD—Counter Latch Command" on page 9-31
<b>Aliased at 40h–43h</b>	50h–53h	
NMI Status and Control Register	61h	Section 9.7.1, "NMI_SC—NMI Status and Control Register" on page 9-51
NMI Enable Register	70h	Section 9.7.2, "NMI_EN—NMI Enable (and Real Time Clock Index)" on page 9-52
Real-Time Clock (Standard RAM) Index Register	70h	Table 9-7 "RTC (Standard) RAM Bank" on page 9-47 Section 9.7.2, "NMI_EN—NMI Enable (and Real Time Clock Index)" on page 9-52
Real-Time Clock (Standard RAM) Target Register	71h	Table 9-7 "RTC (Standard) RAM Bank" on page 9-47
Extended RAM Index Register	72h	
Extended RAM Target Register	73h	

**Table A-1. ICH2 Fixed I/O Registers (Continued)**

Register Name	Port	EDS Section and Location
<b>Aliased at 70h–71h</b>	74h–75h	Aliased if U128E bit in RTC Configuration Register is enabled <a href="#">Section 9.1.24, “RTC_CONF—RTC Configuration Register (LPC I/F—D31:F0)” on page 9-14</a>
<b>Aliased at 72h–73h or 70h–71h</b>	76h–77h	Aliased to 70h–71h if U128E bit in RTC Configuration Register is enabled <a href="#">Section 9.1.24, “RTC_CONF—RTC Configuration Register (LPC I/F—D31:F0)” on page 9-14</a>
Channel 2 DMA Memory Low Page Register	81h	<a href="#">Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 9-25</a>
Channel 3 DMA Memory Low Page Register	82h	<a href="#">Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 9-25</a>
Channel 1 DMA Memory Low Page Register	83h	<a href="#">Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 9-25</a>
Reserved Page Registers	84h–86h	
Channel 0 DMA Memory Low Page Register	87h	<a href="#">Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 9-25</a>
Reserved Page Register	88h	
Channel 6 DMA Memory Low Page Register	89h	<a href="#">Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 9-25</a>
Channel 7 DMA Memory Low Page Register	8Ah	<a href="#">Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 9-25</a>
Channel 5 DMA Memory Low Page Register	8Bh	<a href="#">Section 9.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 9-25</a>
Reserved Page Registers	8Ch–8Eh	
Refresh Low Page Register	8Fh	
<b>Aliased at 81h–8Fh</b>	91h–9Fh (except 92h)	
Fast A20 and INIT Register	92h	<a href="#">Section 9.7.3, “PORT92—Fast A20 and Init Register” on page 9-52</a>
Slave PIC ICW1 Init. Cmd Word 1 Register Slave PIC OCW2 Op Ctrl Word 2 Register Slave PIC OCW3 Op Ctrl Word 3 Register	A0h	<a href="#">Section 9.4.2, “ICW1—Initialization Command Word 1 Register” on page 9-34</a> <a href="#">Section 9.4.8, “OCW2—Operational Control Word 2 Register” on page 9-37</a> <a href="#">Section 9.4.9, “OCW3—Operational Control Word 3 Register” on page 9-38</a>
Slave PIC ICW2 Init. Cmd Word 2 Register Slave PIC ICW3 Init. Cmd Word 3 Register Slave PIC ICW4 Init. Cmd Word 4 Register Slave PIC OCW1 Op Ctrl Word 1 Register	A1	<a href="#">Section 9.4.3, “ICW2—Initialization Command Word 2 Register” on page 9-35</a> <a href="#">Section 9.4.4, “ICW3—Master Controller Initialization Command Word 3 Register” on page 9-35</a> <a href="#">Section 9.4.6, “ICW4—Initialization Command Word 4 Register” on page 9-36</a> <a href="#">Section 9.4.7, “OCW1—Operational Control Word 1 (Interrupt Mask) Register” on page 9-36</a>
<b>Aliased at A0h–A1h</b>	A4h–A5h	
<b>Aliased at A0h–A1h</b>	A8h–A9h	
<b>Aliased at A0h–A1h</b>	ACh–ADh	
<b>Aliased at A0h–A1h</b>	B0h–B1h	

Table A-1. ICH2 Fixed I/O Registers (Continued)

Register Name	Port	EDS Section and Location
Advanced Power Management Control Port Register	B2h	Section 9.8.2.1, "APM_CNT—Advanced Power Management Control Port Register" on page 9-58
Advanced Power Management Status Port Register	B3h	Section 9.8.2.2, "APM_STS—Advanced Power Management Status Port Register" on page 9-58
<b>Aliased at A0h–A1h</b>	B4h–B5h	
<b>Aliased at A0h–A1h</b>	B8h–B9h	
<b>Aliased at A0h–A1h</b>	BCh–BDh	
Channel 4 DMA Base & Current Address Register	C0h	Section 9.2.1, "DMABASE_CA—DMA Base and Current Address Registers" on page 9-24
<b>Aliased at C0h</b>	C1h	
Channel 4 DMA Base & Current Count Register	C2h	Section 9.2.2, "DMABASE_CC—DMA Base and Current Count Registers" on page 9-25
<b>Aliased at C2h</b>	C3h	
Channel 5 DMA Base & Current Address Register	C4h	Section 9.2.1, "DMABASE_CA—DMA Base and Current Address Registers" on page 9-24
<b>Aliased at C4h</b>	C5h	
Channel 5 DMA Base & Current Count Register	C6h	Section 9.2.2, "DMABASE_CC—DMA Base and Current Count Registers" on page 9-25
<b>Aliased at C6h</b>	C7h	
Channel 6 DMA Base & Current Address Register	C8h	Section 9.2.1, "DMABASE_CA—DMA Base and Current Address Registers" on page 9-24
<b>Aliased at C8h</b>	C9h	
Channel 6 DMA Base & Current Count Register	CAh	Section 9.2.2, "DMABASE_CC—DMA Base and Current Count Registers" on page 9-25
<b>Aliased at CAh</b>	CBh	
Channel 7 DMA Base & Current Address Register	CCh	Section 9.2.1, "DMABASE_CA—DMA Base and Current Address Registers" on page 9-24
<b>Aliased at CCh</b>	CDh	
Channel 7 DMA Base & Current Count Register	CEh	Section 9.2.2, "DMABASE_CC—DMA Base and Current Count Registers" on page 9-25
<b>Aliased at CEh</b>	CFh	
Channel 4–7 DMA Command Register Channel 4–7 DMA Status Register	D0h	Section 9.2.4, "DMACMD—DMA Command Register" on page 9-26 Section 9.2.5, "DMASTS—DMA Status Register" on page 9-26
<b>Aliased at D0h</b>	D1h	
Channel 4–7 DMA Write Single Mask Register	D4h	Section 9.2.6, "DMA_WRSMSK—DMA Write Single Mask Register" on page 9-27
<b>Aliased at D4h</b>	D5h	
Channel 4–7 DMA Channel Mode Register	D6h	Section 9.2.7, "DMACH_MODE—DMA Channel Mode Register" on page 9-27
<b>Aliased at D6h</b>	D7h	
Channel 4–7 DMA Clear Byte Pointer Register	D8h	Section 9.2.8, "DMA Clear Byte Pointer Register" on page 9-28
<b>Aliased at D8h</b>	D9h	

**Table A-1. ICH2 Fixed I/O Registers (Continued)**

Register Name	Port	EDS Section and Location
Channel 4–7 DMA Master Clear Register	DAh	Section 9.2.9, “DMA Master Clear Register” on page 9-28
<b>Aliased at DAh</b>	DBh	
Channel 4–7 DMA Clear Mask Register	DCh	Section 9.2.10, “DMA_CLMSK—DMA Clear Mask Register” on page 9-28
<b>Aliased at DCh</b>	DEh	
Channel 4–7 DMA Write All Mask Register	DEh	Section 9.2.11, “DMA_WRMSK—DMA Write All Mask Register” on page 9-29
<b>Aliased at DEh</b>	DFh	
Coprocessor Error Register	F0h	Section 9.7.4, “COPROC_ERR—Coprocessor Error Register” on page 9-52
PIO Mode Command Block Offset for Secondary Drive	170h–177h	See ATA Specification for detailed register description
PIO Mode Command Block Offset for Primary Drive	1F0h–1F7h	See ATA Specification for detailed register description
PIO Mode Control Block Offset for Secondary Drive	376h	See ATA Specification for detailed register description
PIO Mode Control Block Offset for Primary Drive	3F6h	See ATA Specification for detailed register description
Master PIC Edge/Level Triggered Register	4D0h	Section 9.4.10, “ELCR1—Master Controller Edge/Level Triggered Register” on page 9-39
Slave PIC Edge/Level Triggered Register	4D1h	Section 9.4.11, “ELCR2—Slave Controller Edge/Level Triggered Register” on page 9-40
Reset Control Register	CF9h	Section 9.7.5, “RST_CNT—Reset Control Register” on page 9-53

**NOTE:** When the POS\_DEC\_EN bit is set, additional I/O ports get positively decoded by the ICH2. Refer to through for a listing of these ranges.

Table A-2. ICH2 Variable I/O Registers

Register Name	Offset	EDS Section and Location
<b>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 7-5 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 7-5</b>		
SCB Status Word	01h–00h	Section 7.2.1, "System Control Block Status Word Register" on page 7-11
SCB Command Word	03h–02h	Section 7.2.2, "System Control Block Command Word Register" on page 7-12
SCB General Pointer	07h–04h	Section 7.2.3, "System Control Block General Pointer Register" on page 7-14
PORT	0Bh–08h	Section 7.2.4, "PORT Register" on page 7-14
EEPROM Control Register	0Fh–0Eh	Section 7.2.5, "EEPROM Control Register" on page 7-15
MDI Control Register	13h–10h	Section 7.2.6, "Management Data Interface (MDI) Control Register" on page 7-16
Receive DMA Byte Count	17h–14h	Section 7.2.7, "Receive DMA Byte Count Register" on page 7-16
Early Receive Interrupt	18h	Section 7.2.8, "Early Receive Interrupt Register" on page 7-17
Flow Control Register	1Ah–19h	Section 7.2.9, "Flow Control Register" on page 7-18
PMDR	1Bh	Section 7.2.10, "Power Management Driver (PMDR) Register" on page 7-19
General Control	1Ch	Section 7.2.11, "General Control Register" on page 7-19
General Status	1Dh	Section 7.2.12, "General Status Register" on page 7-20
<b>Power Management I/O Registers at PMBASE+Offset  PMBASE set in Section 9.1.10, "PMBASE—ACPI Base Address (LPC I/F—D31:F0)" on page 9-6</b>		
PM1 Status	00–01h	Section 9.8.3.1, "PM1_STS—Power Management 1 Status Register" on page 9-60
PM1 Enable	02–03h	Section 9.8.3.2, "PM1_EN—Power Management 1 Enable Register" on page 9-61
PM1 Control	04–07h	Section 9.8.3.3, "PM1_CNT—Power Management 1 Control Register" on page 9-62
PM1 Timer	08–0Bh	Section 9.8.3.4, "PM1_TMR—Power Management 1 Timer Register" on page 9-62
Processor Control	10h–13h	Section 9.8.3.5, "PROC_CNT—Processor Control Register" on page 9-63
Level 2 Register	14h	Section 9.8.3.6, "LV2—Level 2 Register" on page 9-64
General Purpose Event 0 Status	28–29h	Section 9.8.3.7, "GPE0_STS—General Purpose Event 0 Status Register" on page 9-64
General Purpose Event 0 Enables	2A–2Bh	Section 9.8.3.8, "GPE0_EN—General Purpose Event 0 Enables Register" on page 9-66
General Purpose Event 1 Status	2C–2D	Section 9.8.3.9, "GPE1_STS—General Purpose Event 1 Status Register" on page 9-67
General Purpose Event 1 Enables	2E–2F	Section 9.8.3.10, "GPE1_EN—General Purpose Event 1 Enable Register" on page 9-68

**Table A-2. ICH2 Variable I/O Registers (Continued)**

Register Name	Offset	EDS Section and Location
SMI# Control and Enable	30–31h	Section 9.8.3.11, “SMI_EN—SMI Control and Enable Register” on page 9-68
SMI Status Register	34–35h	Section 9.8.3.12, “SMI_STS—SMI Status Register” on page 9-70
Monitor SMI Status	40h	Section 9.8.3.13, “MON_SMI—Device Monitor SMI Status and Enable Register” on page 9-71
Device Activity Status	44h	Section 9.8.3.14, “DEVACT_STS—Device Activity Status Register” on page 9-72
Device Trap Enable	48h	Section 9.8.3.15, “DEVTRAP_EN—Device Trap Enable Register” on page 9-73
Bus Address Tracker	4Ch	Section 9.8.3.16, “BUS_ADDR_TRACK—Bus Address Tracker Register” on page 9-74
Bus Cycle Tracker	4Eh	Section 9.8.3.17, “BUS_CYC_TRACK—Bus Cycle Tracker Register” on page 9-74
<b>TCO I/O Registers at TCOBASE + Offset</b> TCOBASE = PMBASE + 40h PMBASE is set in Section 9.1.10, “PMBASE—ACPI Base Address (LPC I/F—D31:F0)” on page 9-6		
TCO_RLD: TCO Timer Reload and Current Value	00h	Section 9.9.2, “TCO1_RLD—TCO Timer Reload and Current Value Register” on page 9-75
TCO_TMR: TCO Timer Initial Value	01h	Section 9.9.3, “TCO1_TMR—TCO Timer Initial Value Register” on page 9-76
TCO_DAT_IN: TCO Data In	02h	Section 9.9.4, “TCO1_DAT_IN—TCO Data In Register” on page 9-76
TCO_DAT_OUT: TCO Data Out	03h	Section 9.9.5, “TCO1_DAT_OUT—TCO Data Out Register” on page 9-76
TCO1_STS: TCO Status	04h–05h	Section 9.9.6, “TCO1_STS—TCO1 Status Register” on page 9-76
TCO2_STS: TCO Status	06h–07h	Section 9.9.7, “TCO2_STS—TCO2 Status Register” on page 9-78
TCO1_CNT: TCO Control	08h–09h	Section 9.9.8, “TCO1_CNT—TCO1 Control Register” on page 9-79
TCO2_CNT: TCO Control	0Ah–0Bh	Section 9.9.9, “TCO2_CNT—TCO2 Control Register” on page 9-80
<b>GPIO I/O Registers at GPIOBASE + Offset</b> GPIOBASE is set in Section 9.1.14, “GPIOBASE—GPIO Base Address (LPC I/F—D31:F0)” on page 9-8		
GPIO Use Select	00–03h	Section 9.10.2, “GPIO_USE_SEL—GPIO Use Select Register” on page 9-83
GPIO Input/Output Select	04–07h	Section 9.10.3, “GP_IO_SEL—GPIO Input/Output Select Register” on page 9-84
GPIO Level for Input or Output	0C–0Fh	Section 9.10.4, “GP_LVL—GPIO Level for Input or Output Register” on page 9-84
GPIO Blink Enable	18–1Bh	Section 9.10.5, “GPO_BLINK—GPO Blink Enable Register” on page 9-85
GPIO Signal Invert	2C–2Fh	Section 9.10.6, “GPI_INV—GPIO Signal Invert Register” on page 9-86

Table A-2. ICH2 Variable I/O Registers (Continued)

Register Name	Offset	EDS Section and Location
<b>BMIDE I/O Registers at BM_BASE + Offset</b> BM_BASE is set at Section 10.1.7, "BM_BASE—Bus Master Base Address Register (IDE—D31:F1)" on page 10-4		
Command Register Primary	00h	Section 10.2.1, "BMIC[P,S]—Bus Master IDE Command Register" on page 10-11
Status Register Primary	02h	Section 10.2.2, "BMIS[P,S]—Bus Master IDE Status Register" on page 10-12
Descriptor Table Pointer Primary	04h–07h	Section 10.2.3, "BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register" on page 10-12
Command Register Secondary	08h	Section 10.2.1, "BMIC[P,S]—Bus Master IDE Command Register" on page 10-11
Status Register Secondary	0Ah	Section 10.2.2, "BMIS[P,S]—Bus Master IDE Status Register" on page 10-12
Descriptor Table Pointer Secondary	0Ch–0Fh	Section 10.2.3, "BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register" on page 10-12
<b>USB I/O Registers at Base Address + Offset</b> USB Base Address is set at Section 11.1.9, "BASE—Base Address Register (USB—D31:F2/F4)" on page 11-4		
USB Command Register	00h–01h	Section 11.2.1, "USBCMD—USB Command Register" on page 11-8
USB Status Register	02h–03h	Section 11.2.2, "USBSTA—USB Status Register" on page 11-11
USB Interrupt Enable	04h–05h	Section 11.2.3, "USBINTR—Interrupt Enable Register" on page 11-12
USB Frame Number	06h–07h	Section 11.2.4, "FRNUM—Frame Number Register" on page 11-12
USB Frame List Base Address	08h–0Bh	Section 11.2.5, "FRBASEADD—Frame List Base Address" on page 11-13
USB Start of Frame Modify	0Ch	Section 11.2.6, "SOFMOD—Start of Frame Modify Register" on page 11-13
Port 0, 2 Status/Control	10h–11h	Section 11.2.7, "PORTSC[0,1]—Port Status and Control Register" on page 11-14
Port 1, 3 Status/Control	12h–13h	Section 11.2.7, "PORTSC[0,1]—Port Status and Control Register" on page 11-14
Loop Back Test Data	18h	
<b>SMBus I/O Registers at SMB_BASE + Offset</b> SMB_BASE is set at Section 12.1.7, "SMB_BASE—SMBus Base Address Register (SMBUS—D31:F3)" on page 12-3		
Host Status	00h	Section 12.2.1, "HST_STS—Host Status Register" on page 12-6
Host Control	02h	Section 12.2.2, "HST_CNT—Host Control Register" on page 12-7
Host Command	03h	Section 12.2.3, "HST_CMD—Host Command Register" on page 12-8
Transmit Slave Address	04h	Section 12.2.4, "XMIT_SLVA—Transmit Slave Address Register" on page 12-8
Host Data 0	05h	Section 12.2.5, "HST_D0—Data 0 Register" on page 12-8
Host Data 1	06h	Section 12.2.6, "HST_D1—Data 1 Register" on page 12-8

**Table A-2. ICH2 Variable I/O Registers (Continued)**

Register Name	Offset	EDS Section and Location
Block Data Byte	07h	Section 12.2.7, "BLOCK_DB—Block Data Byte Register" on page 12-9
Receive Slave Address	09h	Section 12.2.8, "RCV_SLVA—Receive Slave Address Register" on page 12-9
Receive Slave Data	0Ah	Section 12.2.9, "SLV_DATA—Receive Slave Data Register" on page 12-9
<b>AC'97 Audio I/O Registers at NAMBAR + Offset</b> <b>NAMBAR is set at Section 13.1.11, "NABMBAR—Native Audio Bus Mastering Base Address Register (Audio—D31:F5)" on page 13-5</b>		
PCM In Buffer Descriptor list Base Address Register	00h	Section 13.2.1, "x_BDBAR—Buffer Descriptor Base Address Register" on page 13-9
PCM In Current Index Value	04h	Section 13.2.2, "x_CIV—Current Index Value Register" on page 13-10
PCM In Last Valid Index	05h	Section 13.2.3, "x_LVI—Last Valid Index Register" on page 13-10
PCM In Status Register	06h	Section 13.2.4, "x_SR—Status Register" on page 13-11
PCM In Position In Current Buffer	08h	Section 13.2.5, "x_PICB—Position In Current Buffer Register" on page 13-12
PCM In Prefetched Index Value	0Ah	Section 13.2.6, "x_PIV—Prefetched Index Value Register" on page 13-12
PCM In Control Register	0Bh	Section 13.2.7, "x_CR—Control Register" on page 13-13
PCM Out Buffer Descriptor list Base Address Register	10h	Section 13.2.1, "x_BDBAR—Buffer Descriptor Base Address Register" on page 13-9
PCM Out Current Index Value	14h	Section 13.2.2, "x_CIV—Current Index Value Register" on page 13-10
PCM Out Last Valid Index	15h	Section 13.2.3, "x_LVI—Last Valid Index Register" on page 13-10
PCM Out Status Register	16h	Section 13.2.4, "x_SR—Status Register" on page 13-11
PCM Out Position In Current Buffer	18h	Section 13.2.5, "x_PICB—Position In Current Buffer Register" on page 13-12
PCM Out Prefetched Index Value	1Ah	Section 13.2.6, "x_PIV—Prefetched Index Value Register" on page 13-12
PCM Out Control Register	1Bh	Section 13.2.7, "x_CR—Control Register" on page 13-13
Mic. In Buffer Descriptor list Base Address Register	20h	Section 13.2.1, "x_BDBAR—Buffer Descriptor Base Address Register" on page 13-9
Mic. In Current Index Value	24h	Section 13.2.2, "x_CIV—Current Index Value Register" on page 13-10
Mic. In Last Valid Index	25h	Section 13.2.3, "x_LVI—Last Valid Index Register" on page 13-10
Mic. In Status Register	26h	Section 13.2.4, "x_SR—Status Register" on page 13-11
Mic In Position In Current Buffer	28h	Section 13.2.5, "x_PICB—Position In Current Buffer Register" on page 13-12
Mic. In Prefetched Index Value	2Ah	Section 13.2.6, "x_PIV—Prefetched Index Value Register" on page 13-12
Mic. In Control Register	2Bh	Section 13.2.7, "x_CR—Control Register" on page 13-13

Table A-2. ICH2 Variable I/O Registers (Continued)

Register Name	Offset	EDS Section and Location
Global Control	2Ch	Section 13.2.8, "GLOB_CNT—Global Control Register" on page 13-14
Global Status	30h	Section 13.2.9, "GLOB_STA—Global Status Register" on page 13-15
Codec Access Semaphore Register	34h	Section 13.2.10, "CAS—Codec Access Semaphore Register" on page 13-16
<b>AC'97 Modem I/O Registers at MBAR + Offset</b> <b>MBAR is set in Section 14.1.11, "MBAR—Modem Base Address Register (Modem—D31:F6)" on page 14-5</b>		
Modem In Buffer Descriptor List Base Address Register	00h	Section 14.2.1, "x_BDBAR—Buffer Descriptor List Base Address Register" on page 14-8
Modem In Current Index Value Register	04h	Section 14.2.2, "x_CIV—Current Index Value Register" on page 14-9
Modem In Last Valid Index Register	05h	Section 14.2.3, "x_LVI—Last Valid Index Register" on page 14-9
Modem In Status Register	06h	Section 14.2.4, "x_SR—Status Register" on page 14-10
Modem In Position In Current Buffer Register	08h	Section 14.2.5, "x_PICB—Position In Current Buffer Register" on page 14-11
Modem In Prefetch Index Value Register	0Ah	Section 14.2.6, "x_PIV—Prefetch Index Value Register" on page 14-11
Modem In Control Register	0Bh	Section 14.2.7, "x_CR—Control Register" on page 14-11
Modem Out Buffer Descriptor List Base Address Register	10h	Section 14.2.1, "x_BDBAR—Buffer Descriptor List Base Address Register" on page 14-8
Modem Out Current Index Value Register	14h	Section 14.2.2, "x_CIV—Current Index Value Register" on page 14-9
Modem Out Last Valid Register	15h	Section 14.2.3, "x_LVI—Last Valid Index Register" on page 14-9
Modem Out Status Register	16h	Section 14.2.4, "x_SR—Status Register" on page 14-10
Modem In Position In Current Buffer Register	18h	Section 14.2.5, "x_PICB—Position In Current Buffer Register" on page 14-11
Modem Out Prefetched Index Register	1Ah	Section 14.2.6, "x_PIV—Prefetch Index Value Register" on page 14-11
Modem Out Control Register	1Bh	Section 14.2.7, "x_CR—Control Register" on page 14-11
Global Control	3Ch	Section 14.2.8, "GLOB_CNT—Global Control Register" on page 14-12
Global Status	40h	Section 14.2.9, "GLOB_STA—Global Status Register" on page 14-13
Codec Access Semaphore Register	44h	Section 14.2.10, "CAS—Codec Access Semaphore Register" on page 14-14



## Register Bit Index

## B

### Numerics

4 Channel Capability 13-15, 14-13

6 Channel Capability 13-15, 14-13

### A

A20Gate Pass-Through Enable  
(A20PASSEN) 11-6

AC '97 Cold Reset# 13-14

AC '97 Interrupt Routing 14-6

AC '97 Interrupt Routing 13-7

AC'97 Cold Reset# 14-12

AC'97 Warm Reset 13-14, 14-12

AC97\_EN 9-66

AC97\_STS 9-65

ACLINK Shut Off 13-14, 14-12

ACPI\_EN 9-6

AD3 13-15, 14-13

ADDRESS 12-8

Address Increment/Decrement Select 9-27

ADLIB\_ACT\_STS 9-72

ADLIB\_LPC\_EN 9-18

ADLIB\_TRP\_EN 9-73

AF Alarm Flag 9-50

AFTERG3\_EN 9-55

AIE Alarm Interrupt Enable 9-49

ALT\_A20\_GATE 9-52

ALTACC\_EN Alternate Access Mode Enable  
9-12

APIC Data 9-42

APIC ID 9-43

APIC Index 9-41

APIC\_EN 9-12

APM\_STS 9-71

APMC\_EN 9-69

AUDIO\_ACT\_STS 9-72

AUDIO\_TRP\_EN 9-73

Autoinitialize Enable 9-27

Automatic End of Interrupt (AEOI) 9-36

Auxiliary Current 7-8

### B

Base Address 7-5, 9-6, 9-8, 10-4, 11-4, 11-13,  
12-3, 13-5, 14-5

Base address of Descriptor table (BADDR)  
10-12

Base and Current Address 9-24

Base and Current Count 9-25

Base Class Code 7-4, 8-5, 9-5, 10-4, 11-4,  
12-3, 13-4

Base Class Code Value 14-4

Binary/BCD Countdown Select 9-30

BIOS\_EN 9-69

BIOS\_RLS BIOS Release 9-69

BIOS\_STS 9-71

BIOSWE BIOS Write Enable 9-7

BIOSWR\_STS 9-77

Bit 1 of slot 12 13-15, 14-13

Bit 2 of slot 12 13-15, 14-13

Bit 3 of slot 12 13-15, 14-13

BLE BIOS Lock Enable 9-7

Block Data Byte 12-9

BOOT\_STS 9-78

Buffer Completion Interrupt Status (BCIS)  
13-11, 14-10

Buffer Descriptor Base Address 13-9

Buffer Descriptor List Base Address 14-8

Buffered Mode (BUF) 9-36

Bus Master Enable (BME) 7-2, 8-3, 10-2,  
11-2, 13-2, 14-2

Bus Master IDE Active (ACT) 10-12

BUS\_ERR 12-6

BYTE\_DONE\_STS 12-6

### C

CAP\_ID Capability ID 7-8

CAP\_LIST Capabilities List 7-3

CAP\_PTR Capabilities Pointer 7-6

Cascaded Interrupt Controller IRQ Con-  
nection 9-35

Channel 0 Select 9-11

- Channel 1 Select 9-11
- Channel 2 Select. 9-11
- Channel 3 Select 9-11
- Channel 5 Select 9-11
- Channel 6 Select 9-11
- Channel 7 Select 9-11
- Channel Mask Bits 9-29
- Channel Mask Select 9-27
- Channel Request Status 9-26
- Channel Terminal Count Status 9-26
- Clear Byte Pointer 9-28
- Clear Mask Register 9-28
- CLS Cache Line Size 7-4
- CNA CU Not Active 7-11
- CNA Mask 7-12
- CNF1\_LPC\_EN 9-17
- CNF2\_LPC\_EN 9-17
- Codec Write In Progress (CWIP) 13-16
- COMA Decode Range 9-14
- COMA\_LPC\_EN 9-18
- COMB Decode Range 9-14
- COMB\_LPC\_EN 9-18
- Configure Flag (CF) 11-9
- Connect Status Change 11-15
- COPR\_ERR\_EN Coprocessor Error Enable 9-11
- COPROC\_ERR 9-52
- Count Register Status 9-32
- Countdown Type Status 9-32
- Counter 0 Select 9-31
- Counter 1 Select 9-31
- Counter 2 Select 9-31
- Counter Latch Command 9-31
- Counter Mode Selection 9-30
- Counter OUT Pin State 9-32
- Counter Port 9-32
- Counter Select 9-30
- Counter Selection 9-31
- CPU\_BIST\_EN 9-13
- CPUPWR\_FLR CPU Power Failure 9-55
- CPUSLP\_EN 9-54
- CUC Command Unit Command 7-13
- Current Connect Status 11-15
- Current Equals Last Valid (CELV) 13-11, 14-10
- Current Index Value 13-10, 14-9
- CUS Command Unit Status 7-11
- CX Command Unit (CU) Executed 7-11
- CX Mask 7-12
- D**
- D1 Support 7-8
- D2 Support 7-8
- Data 7-16
- Data Parity Error Detected (DPD) 7-3, 8-4, 8-8
- Data Scale 7-9
- Data Select 7-9
- DATA\_MSG0 Data Message Byte 0 12-9
- DATA\_MSG1 Data Message Byte 1 12-9
- DATA0/COUNT 12-8
- DATA1 12-8
- Date Alarm 9-50
- DCB\_EN DMA Collection Buffer Enable 9-12
- Deep Power-Down on Link Down Enable 7-19
- Delivery Mode 9-46
- Delivery Status 9-45
- Destination 9-45
- Destination Mode 9-45
- Detected Parity Error (DPE) 7-3, 8-4, 8-8
- DEV\_ERR 12-6
- DEV\_STS DEVSEL# Timing Status 9-4
- DEV\_TRAP\_EN 9-71
- DEV\_TRAP\_STS 9-71
- Device ID Value 9-2, 11-2, 13-2, 14-2
- Device ID value 12-1
- Device Identification Number 7-2, 8-2
- Device Specific Initialization (DSI) 7-8
- DEVMON\_STS Device Monitor Status 9-70
- DEVSEL# Timing Status (DEVT) 10-3, 11-3, 12-2, 13-3
- DM Data Mode 9-49
- DMA Channel Group Enable 9-26
- DMA Channel Select 9-27
- DMA Controller Halted (DCH) 13-11, 14-10
- DMA Group Arbitration Priority 9-26
- DMA Low Page 9-25

DMA Transfer Mode 9-27  
 DMA Transfer Type 9-27  
 DPE Detected Parity Error 9-4  
 DPED Data Parity Error Detected 9-4  
 Drive 0 DMA Capable 10-12  
 Drive 0 DMA Timing Enable (DTE0) 10-6  
 Drive 0 Fast Timing Bank (TIME0) 10-7  
 Drive 0 IORDY Sample Point Enable (IE0) 10-7  
 Drive 0 Prefetch/Posting Enable (PPE0) 10-7  
 Drive 1 DMA Capable 10-12  
 Drive 1 DMA Timing Enable (DTE1) 10-6  
 Drive 1 Fast Timing Bank (TIME1) 10-6  
 Drive 1 IORDY Sample Point Enable (IE1) 10-6  
 Drive 1 Prefetch/Posting Enable (PPE1) 10-6  
 Drive 1 Timing Register Enable (SITRE) 10-6  
 DSE Daylight Savings Enable 9-49  
 DT Delivery Type 9-44  
 DTE Delayed Transaction Enable 9-12  
 Duplex Mode 7-20  
 DV Division Chain Select 9-48  
 Dynamic Data 7-9

## E

Early Receive Count 7-17  
 Edge/Level Bank Select (LTIM) 9-34  
 EECS EEPROM Chip Select 7-15  
 EEDI EEPROM Serial Data In 7-15  
 EEDO EEPROM Serial Data Out 7-15  
 EESK EEPROM Serial Clock 7-15  
 Enable Special Mask Mode (ESMM) 9-38  
 Enter Global Suspend Mode (EGSM) 11-9  
 EOS End of SMI 9-69  
 ER Early Receive 7-11  
 ER Mask 7-12  
 Error 10-12

## F

F1\_Disable 9-22  
 F2\_Disable 9-22  
 F3\_Disable 9-22  
 F4\_Disable 9-22  
 F5\_Disable 9-22  
 F6\_Disable 9-22  
 FAILED 12-6

FAST\_PCB0 Fast Primary Drive 0 Base Clock 10-10  
 FAST\_PCB1 Fast Primary Drive 1 Base Clock 10-9  
 FAST\_SCB0 Fast Secondary Drive 0 Base Clock 10-9  
 FAST\_SCB1 Fast Secondary Drive 1 Base Clock 10-9  
 FC Full 7-18  
 FC Paused 7-18  
 FC Paused Low 7-18  
 FCP Flow control Pause 7-11  
 FCP Mask 7-12  
 FDD Decode Range 9-15  
 FDD\_LPC\_EN 9-18  
 FIFO error (FIFOE) 13-11, 14-10  
 FIFO Error Interrupt Enable (FEIE) 13-13, 14-11  
 Flow Control Threshold 7-18  
 Force Global Resume (FGR) 11-9  
 FORCE\_THTL 9-63  
 FR Frame Received 7-11  
 FR Mask 7-12  
 Frame List Current Index/Frame Number 11-12  
 FREQ\_STRAP 9-13  
 FULL\_RST 9-53  
 FWH\_C0\_EN 9-16, 9-21  
 FWH\_C0\_IDSEL 9-19  
 FWH\_C8\_EN 9-16, 9-21  
 FWH\_C8\_IDSEL 9-19  
 FWH\_D0\_EN 9-16, 9-21  
 FWH\_D0\_IDSEL 9-19  
 FWH\_D8\_EN 9-16, 9-21  
 FWH\_D8\_IDSEL 9-19, 9-20  
 FWH\_E0\_EN 9-16  
 FWH\_E0\_IDSEL 9-19, 9-20  
 FWH\_E8\_EN 9-16  
 FWH\_E8\_IDSEL 9-19, 9-20  
 FWH\_F0\_EN 9-16  
 FWH\_F0\_IDSEL 9-19, 9-20  
 FWH\_F8\_EN 9-16  
 FWH\_F8\_IDSEL 9-19

**G**

GAMEH\_LPC\_EN 9-17  
 GAMEL\_LPC\_EN 9-17  
 GBL\_STS 9-61  
 GBL\_EN 9-61  
 GBL\_RLS Global Release 9-62  
 GBL\_SMI\_EN 9-69  
 GEN1\_BASE Generic I/O Decode Range 1 Base 9-17  
 GEN1\_EN Generic Decode Range 1 Enable 9-17  
 GEN2\_BASE Generic I/O Decode Range 2 Base 9-20  
 GEN2\_EN Generic I/O Decode Range 2 Enable 9-20  
 Global Reset (GRESET) 11-9  
 GPE0\_STS 9-70  
 GPE1\_STS 9-70  
 GPI Interrupt Enable (GIE) 13-14, 14-12  
 GPI Route 9-56  
 GPI Status Change Interrupt (GSCI) 13-16, 14-14  
 GPI\_EN 9-68  
 GPI\_STS 9-67  
 GPIO\_EN 9-8  
 GPIO\_SEL 9-84  
 GPIO\_USE\_SEL 9-83

**H**

HCHalted 11-11  
 Header Type 7-5, 9-5  
 Header Type Value 13-4  
 Header Value 14-4  
 HIDE\_ISA Hide ISA Bridge 9-11  
 Hole Enable (15MB-16MB). 8-12  
 Host Controller Process Error 11-11  
 Host Controller Reset (HCRESET) 11-9  
 Host System Error 11-11  
 HOST\_BUSY 12-6  
 HOURFORM Hour Format 9-49  
 HP\_PCI\_EN 8-12  
 HST\_EN SMBus Host Enable 12-4  
 HUBNMI\_STS 9-76  
 HUBSCI\_STS 9-77  
 HUBSERR\_STS 9-76

HUBSMI\_STS 9-77

**I**

I/O Address Base bits 8-7  
 I/O Address Limit bits 8-7  
 I/O Addressing Capability 8-7  
 I/O APIC Identification 9-44  
 I/O Space (IOS) 14-2  
 I/O Space Enable (IOE) 7-2, 8-3  
 I/O Space Enable (IOSE) 11-2, 12-2  
 I2C\_EN 12-4  
 iA64\_EN  
 iA64 Processor Mode Enable 9-54  
 ICW/OCW select 9-34  
 ICW4 Write Required (IC4) 9-34  
 IDE Decode Enable (IDE) 10-6  
 IDEP0\_ACT\_STS 9-73  
 IDEP0\_TRP\_EN 9-73  
 IDEP1\_ACT\_STS 9-73  
 IDEP1\_TRP\_EN 9-73  
 IDES0\_ACT\_STS 9-72  
 IDES0\_TRP\_EN 9-73  
 IDES1\_ACT\_STS 9-72  
 IDES1\_TRP\_EN 9-73  
 INIT\_NOW 9-52  
 INT\_LN Interrupt Line 7-7  
 INT\_PN Interrupt Pin 7-7  
 Interesting Packet 7-19  
 Internal LAN Master Request Status (LAN\_MREQ\_STS) 8-13  
 Internal PCI Master Request Status (INT\_MREQ\_STS) 8-13  
 Interrupt 10-12  
 Interrupt Enable 7-16  
 Interrupt Input Pin Polarity 9-45  
 Interrupt Level Select (L2, L1, L0) 9-37  
 Interrupt Line 11-5, 13-6, 14-6  
 Interrupt line 12-4  
 Interrupt On Complete (IOC) Enable 11-12  
 Interrupt On Completion Enable (IOCE) 13-13, 14-11  
 Interrupt PIN 12-4  
 Interrupt pin 11-5  
 Interrupt Request Level 9-35  
 Interrupt Request Mask 9-36

Interrupt Vector Base Address 9-35  
 INTR 12-6  
 INTRD\_DET Intruder Detect 9-78  
 INTRD\_SEL 9-80  
 INTREN 12-7  
 INUSE\_STS 12-6  
 IO Space Indicator 12-3  
 IOCHK\_NMI\_EN 9-51  
 IOCHK\_NMI\_STS IOCHK# NMI Source  
     Status 9-51  
 IORDY Sample Point (ISP) 10-6  
 IOS (I/O Space) 13-2  
 IOSE I/O Space Enable (IOSE) 10-2  
 IRQ Number 9-42  
 IRQ Routing 9-8, 9-9  
 IRQ10 ECL 9-40  
 IRQ11 ECL 9-40  
 IRQ12 ECL 9-40  
 IRQ12LEN Mouse IRQ12 Latch Enable 9-12  
 IRQ14 ECL 9-40  
 IRQ15 ECL 9-40  
 IRQ1LEN Keyboard IRQ1 Latch Enable 9-11  
 IRQ3 ECL 9-39  
 IRQ4 ECL 9-39  
 IRQ5 ECL 9-39  
 IRQ6 ECL 9-39  
 IRQ7 ECL 9-39  
 IRQ9 ECL 9-40  
 IRQEN Interrupt Routing Enable 9-8, 9-9  
 IRQF Interrupt Request Flag 9-50  
 ISA Enable 8-11  
**K**  
 KBC\_ACT\_STS 9-72  
 KBC\_LPC\_EN 9-17  
 KBC\_TRP\_EN 9-73  
 KILL 12-7  
**L**  
 L128LOCK Lower 128-byte Lock 9-14  
 LAN Connect Address 7-16  
 LAN Connect Register Address 7-16  
 LAN Connect Software Reset 7-19  
 Last Valid Buffer Completion Interrupt (LVB-  
     CI) 13-11, 14-10

Last Valid Buffer Interrupt Enable (LVBIE)  
     13-13, 14-11  
 Last Valid Index 13-10, 14-9  
 Latch Count of Selected Counters 9-31  
 Latch Status of Selected Counters 9-31  
 LEG\_ACT\_STS 9-72  
 LEG\_IO\_TRP\_EN 9-73  
 LEGACY\_USB\_EN 9-69  
 LEGACY\_USB\_STS 9-71  
 Line Status 11-14  
 Link Status Change Indication 7-19  
 Link Status Indication 7-20  
 Loop Back Test Mode 11-8  
 Low Speed Device Attached (LS) 11-14  
 LPT Decode Range 9-15  
 LPT\_LPC\_EN 9-18  
**M**  
 M Interrupt Mask 7-12  
 Magic Packet 7-19  
 MAS (Master-Abort Status) 14-3  
 Mask 9-45  
 Master Abort Mode 8-11  
 Master Clear 9-28  
 Master Latency Count 8-6  
 Master/Slave in Buffered Mode 9-36  
 Master-Abort Status (MAS) 13-3  
 Max Packet (MAXP) 11-9  
 Maximum Redirection Entries 9-44  
 MC\_LPC\_EN 9-17  
 MCSMI\_EN Microcontroller SMI Enable  
     9-68  
 MCSMI\_STS Microcontroller SMI# Status  
     9-70  
 MD3 13-15, 14-13  
 MDI Management Data Interrupt 7-11  
 Memory Address Base 8-9  
 Memory Address Limit 8-9  
 Memory Space Enable (MSE) 7-2, 8-3  
 Mic In Interrupt (MINT) 13-15, 14-13  
 Microprocessor Mode 9-36  
 MIDI Decode Range 9-15  
 MIDI\_ACT\_STS 9-72  
 MIDI\_LPC\_EN 9-18  
 MIDI\_TRP\_EN 9-73

MLTC Master Latency Timer Count 7-4  
Mode Selection Status 9-32  
Modem In Interrupt (MIINT) 13-16, 14-14  
Modem Out Interrupt (MOINT) 13-16, 14-14  
MON\_TRAP\_BASE 9-57  
MON4\_FWD\_EN 9-56  
MON4\_MASK 9-57  
MON5\_FWD\_EN 9-56  
MON5\_MASK 9-57  
MON6\_FWD\_EN 9-56  
MON6\_MASK 9-57  
MON7\_FWD\_EN 9-56  
MON7\_MASK 9-57  
MSS Decode Range 9-15  
MSS\_LPC\_EN 9-18  
Multi-function Device 9-5  
Multi-function Device. 7-5  
Multi-Transaction Timer Count Value 8-12  
MWIE Memory Write and Invalidate Enable  
7-2  
**N**  
NEWCENTURY\_STS 9-77  
NMI\_EN 9-52  
NMI\_NOW 9-79  
NMI2SMI\_EN 9-77, 9-79  
NO\_REBOOT 9-13  
NXT\_PTR Next Item Pointer 7-8  
**O**  
OCW2 Select 9-37  
OCW3 Select 9-38  
Opcode 7-16  
Overcurrent Active 11-14  
Overcurrent Indicator 11-14  
**P**  
Parity Error Response 8-3  
Parity Error Response Enable 8-11  
Pass Through State (PSTATE) 11-6  
PCB0 10-10  
PCB1 10-10  
PCI Interrupt Enable (USBPIRQEN) 11-6  
PCI Master Request Status  
(PCI\_MREQ\_STS) 8-13  
PCI\_DAC\_EN 8-11  
PCI\_SERR\_EN 9-51  
PCM 4/6 Enable 13-14  
PCM In Interrupt (PIINT) 13-16, 14-14  
PCM Out Interrupt (POINT) 13-16, 14-13  
PER Parity Error Response 7-2, 9-3  
PER\_SMI\_SEL 9-54  
PERIODIC\_EN 9-68  
PERIODIC\_STS 9-70  
PF Periodic Interrupt Flag 9-50  
PIE Periodic Interrupt Enable 9-49  
PIRQAE\_ACT\_STS 9-72  
PIRQBF\_ACT\_STS 9-72  
PIRQCG\_ACT\_STS 9-72  
PIRQDH\_ACT\_STS 9-72  
PM1\_STS\_REG 9-70  
PME Clock 7-8  
PME Enable 7-9  
PME Status 7-9, 7-19  
PME Support 7-8  
PME\_EN 9-66  
PME\_STS 9-64  
Pointer Field 7-14  
Poll Mode Command 9-38  
Port Enable/Disable Change 11-15  
Port Enabled/Disabled (PORT\_EN) 11-15  
PORT Function Selection 7-14  
Port Reset 11-14  
PORT0EN 11-7  
PORT1EN 11-7  
POS\_DEC\_EN Positive Decode Enable 9-12  
Position In Current Buffer 13-12, 14-11  
Power State 7-9  
PRBTNOR\_STS Power Button Override Sta-  
tus 9-60  
Prefetchable Memory Address Base 8-9  
Prefetchable Memory Address Limit 8-10  
Prefetched Index Value 13-12  
Prefetched Index value 14-11  
PRIM\_SIG\_MODE 10-9  
Primary Resume Interrupt Enable 13-14,  
14-12  
Primary Bus Number 8-6  
Primary Codec Ready (PCR) 13-15, 14-13  
Primary Drive 0 Cycle Time (PCT0) 10-9

Primary Drive 0 Synchronous DMA Mode Enable (PSDE0) 10-8  
 Primary Drive 1 Cycle Time (PCT1) 10-9  
 Primary Drive 1 IORDY Sample Point (PISP1) 10-7  
 Primary Drive 1 Recovery Time (PRCT1) 10-7  
 Primary Drive 1 Synchronous DMA Mode Enable (PSDE1) 10-8  
 Primary Master Channel Cable Reporting 10-10  
 Primary Resume Interrupt 13-15, 14-13  
 Primary Slave Channel Cable Reporting 10-10  
 Programming Interface Value 10-3, 14-3  
 PRQ 9-44  
 PWR\_FLR Power Failure 9-55  
 PWRBTN\_\_STS 9-60  
 PWRBTN\_EN 9-61  
 PWRBTN\_LVL 9-54  
 PWROK\_FLR PWROK Failure 9-55  
**R**  
 Read / Write Control (RWC) 10-11  
 Read Back Command 9-31  
 Read Completion Status 13-15  
 Read/Write Select 9-30  
 Read/Write Selection Status 9-32  
 Ready 7-16  
 Receive DMA Byte Count 7-16  
 Received Master Abort (RMA) 8-4, 8-8  
 Received Master-Abort Status (RMA) 10-3, 11-3  
 Received System Error (SSE) 8-4, 8-8  
 Received Target Abort (RTA) 7-3, 8-4, 8-8  
 Recovery Time (RCT) 10-6  
 Redirection Entry Clear 9-43  
 REF\_TOGGLE Refresh Cycle Toggle 9-51  
 Register Read Command 9-38  
 Remote IRR 9-45  
 Reset Registers(RR) 14-11  
 Reset Registers(RR). 13-13  
 Resource Indicator 9-6, 9-8  
 Resource Type Indicator (RTE) 10-4, 11-4, 13-5, 14-5  
 Resume Detect (RSM\_DET) 11-11, 11-14

Resume Interrupt Enable 11-12  
 Revision ID Value 13-3, 14-3  
 Revision Identification Number 8-4, 9-4  
 Revision Identification Number. 7-3  
 RI\_EN 9-66  
 RI\_STS 9-64  
 RMA Master Abort Status 7-3, 9-4  
 RNR Mask 7-12  
 RNR Receive Not Ready 7-11  
 Rotate and EOI Codes (R, SL, EOI) 9-37  
 RS Rate Select 9-48  
 RST\_CPU 9-53  
 RTA Received Target Abort 9-4  
 RTC\_EN RTC Event Enable 9-61  
 RTC\_INDX Real Time Clock Index Address 9-52  
 RTC\_PWR\_STS 9-55  
 RTC\_STS 9-60  
 RUC Receive Unit Command 7-13  
 Run/Pause Bus master (RPM) 13-13, 14-11  
 Run/Stop (RS) 11-10  
 RUS Receive Unit Status 7-11  
 RW 12-8  
**S**  
 SAFE\_MODE 9-13  
 SB16 Decode Range 9-15  
 SB16\_LPC\_EN 9-18  
 SCB General Pointer 7-14  
 SCB1 10-10  
 SCBO 10-10  
 SCI\_EN 9-62  
 SCI\_IRQ\_SEL 9-6  
 SEC\_SIG\_MODE 10-9  
 SECOND\_TO\_STS 9-78  
 Secondary Bus Number 8-6  
 Secondary Codec Ready (SCR) 13-15, 14-13  
 Secondary Drive 0 Cycle Time (SCT0) 10-8  
 Secondary Drive 0 Synchronous DMA Mode Enable (SSDE0) 10-8  
 Secondary Drive 1 Cycle Time (SCT1) 10-8  
 Secondary Drive 1 IORDY Sample Point (SISP1) 10-7  
 Secondary Drive 1 Recovery Time (SRCT1) 10-7

Secondary Drive 1 Synchronous DMA Mode Enable (SSDE1) 10-8	SIRQMD Serial IRQ Mode Select 9-9
Secondary Master Channel Cable Reporting 10-10	SIRQSZ Serial IRQ Frame Size 9-9
Secondary Resume Interrupt 13-15, 14-13	Slave Identification Code 9-36
Secondary Resume Interrupt Enable 13-14, 14-12	SLAVE_ADDR 12-9
Secondary Slave Channel Cable Reporting 10-10	SLP_EN 9-62
SENDNOW 9-79	SLP_SMI_EN 9-69
Serial Bus Release Number 11-5	SLP_SMI_STS 9-71
SERIRQ_SMI_STS 9-70	SLP_TYP 9-62
SERR# Due to Delayed Transaction Timeout (SERR_DTT). 8-14	SMB_CMD 12-7
SERR# Due to Received Target Abort (SERR_RTA). 8-14	SMB_FOR_BIOS 9-22
SERR# Enable 8-11	SMB_SMI_EN 12-4
SERR# Enable (SERR_EN) 7-2, 8-3	SMB_WAK_STS SMBus Wake Status 9-64
SERR# enable on Delayed Transaction Timeout (SERR_DTT_EN) 8-13	SMBALERT_STS 12-6
SERR# enable on receiving target abort (SERR_RTA_EN) 8-13	SMBUS_SMI_STS 9-70
SERR#_NMI_STS SERR# NMI Source Status 9-51	SMI at End of Pass-through Enable (SMIA-TENDPS) 11-6
SERR_DTT SERR# Due to Delayed Transaction Timeout 9-10	SMI Caused by End of Pass-through (SMIBY-ENDPS) 11-6
SERR_DTT_EN SERR# on Delayed Transaction Timeout Enable 9-10	SMI Caused by Port 60 Read (TRAPBY60R) 11-6
SERR_EN 9-3	SMI Caused by Port 60 Write (TRAPBY60W) 11-6
SERR_RTA SERR# Due to Received Target Abort 9-10	SMI Caused by Port 64 Read (TRAPBY64R) 11-6
SERR_RTA_EN SERR# on Received Target Abort Enable 9-10	SMI Caused by Port 64 Write (TRAPBY64W) 11-6
SET Update Cycle Inhibit 9-49	SMI Caused by USB Interrupt (SMIBYUSB) 11-6
SFPW Start Frame Pulse Width 9-9	SMI on Port 60 Reads Enable (60REN) 11-7
Short Packet Interrupt Enable 11-12	SMI on Port 60 Writes Enable (60WEN) 11-7
SI Software Generated Interrupt 7-12	SMI on Port 64 Reads Enable (64REN) 11-7
Signaled System Error (SSE) 7-3	SMI on Port 64 Writes Enable (64WEN) 11-7
Signaled Target Abort (STA) 8-4	SMI on USB IRQ (USBSMIEN) 11-7
Signaled Target-Abort Status 12-2	SOF Timing Value 11-13
Signaled Target-Abort Status (STA) 10-3, 11-3	Software Debug (SWDBG) 11-9
Single or Cascade (SNGL) 9-34	Special Fully Nested Mode (SFNM) 9-36
SIRQEN Serial IRQ Enable 9-9	Special Mask Mode (SMM) 9-38
	Speed 7-20
	SPKR_DAT_EN 9-51
	SQWE Square Wave Enable 9-49
	SSE Signaled System Error 9-4
	STA Signaled Target Abort 9-4
	START 12-7
	Start/Stop Bus Master (START) 10-11

Sub Class Code 10-3, 11-4, 12-3, 13-4  
 Sub Class Code Value 14-4  
 Sub-Class Code 7-4, 8-5, 9-5  
 Subordinate Bus Number 8-6  
 Subsystem ID Value 13-6, 14-6  
 Subsystem Vendor ID Value 13-6, 14-5  
 Suspend 11-14  
 SW\_TCO\_SMI 9-77  
 SWI Software Interrupt 7-11  
 SWSMI\_RATE\_SEL 9-54  
 SWSMI\_TMR\_EN Software SMI# Timer Enable 9-69  
 SWSMI\_TMR\_STS 9-71  
 SYS\_RST 9-53  
**T**  
 TCO\_EN 9-68  
 TCO\_INT\_EN TCO Interrupt Enable 9-7  
 TCO\_INT\_SEL TCO Interrupt Select 9-7  
 TCO\_INT\_STS 9-77  
 TCO\_MESSAGE 9-80  
 TCO\_STS 9-70  
 TCO\_TMR\_HLT TCO Timer Halt 9-79  
 TCOSCI\_EN 9-66  
 TCOSCI\_STS 9-64  
 THRM#\_POL 9-66  
 THRM\_DTY 9-63  
 THRM\_EN 9-66  
 THRM\_STS Thermal Interrupt Status 9-65  
 THRMOR\_STS Thermal Interrupt Override Status 9-65  
 THT\_EN 9-63  
 THTL\_DTY 9-63  
 THTL\_STS Throttle Status 9-63  
 TIM\_CNT2\_EN Timer Counter 2 Enable 9-51  
 TIMEOUT 9-77

Timeout/CRC Interrupt Enable 11-12  
 TMR\_VAL 9-62  
 TMR2\_OUT\_STS Timer Counter 2 OUT Status 9-51  
 TMROF\_EN Timer Overflow Interrupt Enable 9-61  
 TMROF\_STS Timer Overflow Status 9-61  
 TOP\_SWAP 9-13  
 Trigger Mode 9-45  
**U**  
 U128E Upper 128-byte Enable 9-14  
 U128LOCK Upper 128-byte Lock 9-14  
 UF Update-ended Flag 9-50  
 UIE Update-ended Interrupt Enable 9-49  
 UIP Update In Progress 9-48  
 USB Error Interrupt 11-11  
 USB Interrupt (USBINT) 11-11  
 USB1\_EN 9-66  
 USB1\_STS 9-65  
 USB2\_EN 9-66  
 USB2\_STS 9-65  
**V**  
 Vendor ID Value 9-2, 11-1, 12-1, 13-1, 14-1  
 Vendor Identification Number 8-2  
 Version 7-8  
 VGA Enable 8-11  
 VRT Valid RAM and Time Bit 9-50  
**W**  
 WAK\_STS 9-60  
 WDSTATUS Watchdog Status 9-80  
 WR\_PingPong\_EN 10-10  
**X**  
 Xoff 7-18  
 Xon 7-18

