Intel® 82801AA (ICH) and Intel® 82801AB (ICH0) I/O Controller Hub

Specification Update

September 2003

Notice: The Intel® 82801AA ICH and Intel® 82801AB ICH0 may contain design defects or errors known as errata. Characterized errata that may cause the Intel® 82801AA ICH and Intel® 82801AB ICH0's behavior to deviate from published specifications are documented in this specification update.

Document Number: 290677-014

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, or life sustaining applications.

Intel may make changes to specifications and product descriptions at any time, without notice.

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

The Intel® 82801AA ICH and Intel® 82801AB ICH0 may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

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

Contents

Revision History 4

Preface.. 5

Summary Table of Changes.. 6

Identification Information 12

Errata 13

Specification Changes 33

Specification Clarifications 37

Documentation Changes 43

Revision History

Revision	Description	Date
-001	Initial Release	October 1999
-002	Added Errata #41-45. Added Specification Change #2-10. Added Specification Clarification #3-11. Added Documentation Change #4-9.	April 2000
-003	Added Errata #46-55, Specification Changes #11-15, Specification Clarifications #12-16, and Documentation Changes #12-16	January 2001
-004	Added Errata #56-58, Specification Clarification #17, Specification Change #16, and Documentation Changes #13-14. Also added new S-Spec for ICH0.	May 2001
-005	Added **Errata:** AC97 Reset Race Condition, **Specification Changes:** I/O APIC Arbitration ID Not Set When APIC CLK Not Running, **Specification Clarifications: Running Block Read With SECOND_TO_STS Set.**	August 2001
-006	Added Errata: I2C Read Command Issue.	September 2001
-007	Added Errata: SE0 During Resume Causes Disconnect, PERR# Detection Issue and PERR# Response Issue	November 2001
-008	Added Errata: Delayed Transaction Timeout Bit Issue, SMBus NACK and Proc_Call Issue, SMBus Arbitration, SMI Asserted after STPCLK# is Active and Stopgrant Reveived, Specification Clarifications: DMA Clarification, USB Run/Stop Bit Clarification	April 2002
-009	Added Errata: AC’97 Overrun FIFO Error Bit Not Set Specification Clarifications: RTC SET Bit Clarification, End of SMI Bit, SMBus Wake	May 2002
-010	Added: Specification Clarification: 32 Clock Retry Enable Clarification. Documentation Changes: SUSCLK During RSMRST# Assertion, TCO Corrections.	July 2002
-011	Added: Specification Clarification: LPC LPCPD# Protocol Clarification	August 2002
-012	Added: Errata: PCI Non-linear Addressing Erratum Specification Clarification: PCI Master Clarification	June 2003
-013	Added: Errata: MW DMA Mode-1 Tdh Erratum Documentation Changes: APM I/O Decode Correction, Memory Map Table Change	August 2003
-014	Added: Documentation Changes: SMBus Host_Busy Correction	September 2003

Preface

This document is an update to the specifications contained in the documents listed in the following Affected Documents/Related Documents table. It is a compilation of device and document errata and specification clarifications and changes, and is intended for hardware system manufacturers and for software developers of applications, operating system, and tools.

Information types defined in the Nomenclature section of this document are consolidated into this update document and are no longer published in other documents. This document may also contain information that has not been previously published.

Affected Documents/Related Documents

Title	Number
Intel® 82801AA (ICH) and Intel® 82801AB (ICH0) Datasheet	290655-003

Nomenclature

Errata are design defects or errors. Errata may cause the ICH's behavior to deviate from published specifications. Hardware and software designed to be used with any given stepping must assume that all errata documented for that stepping are present in all devices.

Specification Changes are modifications to the current published specifications. These changes will be incorporated in any new release of the specification.

Specification Clarifications describe a specification in greater detail or further highlight a specification's impact to a complex design situation. These clarifications will be incorporated in any new release of the specification.

Documentation Changes include typos, errors, or omissions from the current published specifications. These will be incorporated in any new release of the specification.

Summary Table of Changes

The following table indicates the Specification Changes, Errata, Specification Clarifications, or Documentation Changes which apply to the Intel® 82801AA (ICH) and Intel® 82801AB (ICH0) I/O Controller Hub. Intel intends to fix some of the errata in a future stepping of the component(s), and to account for the other outstanding issues through documentation or specification changes as noted. This table uses the following notations:

Codes Used in Summary Table

Stepping

X:Specification Change, Erratum, Specification Clarification or Documentation Change that applies to a stepping or to this product line.

(No mark) or (Blank Box):This erratum is fixed in listed stepping or specification change does not apply to listed stepping.

Status

Doc:Document change or update that will be implemented.

PlanFix:This erratum may be fixed in a future stepping of the product.

Fixed:This erratum has been previously fixed.

NoFix:There are no plans to fix this erratum.

|(bar) This item is either new or modified from the previous version of the document.

Errata (Sheet 1 of 3)

No.	Steppings				Status	ERRATA
	A0	A1	B0	B1		
1	X				Fixed	Resume Well
2	X	X			Fixed	USB Port Enable and Port Status Bits
3	X	X			Fixed	SMBus Time-Out
4	X	X			Fixed	Alert On LAN* Busy Bit
5	X	X			Fixed	AC’97 Cold Reset
6	X	X			Fixed	PCI Peer-to-Peer Prefetchable Memory
7	X	X			Fixed	SMI on BIOS Access
8	X				Fixed	SCI from SMBus
9	X	X			Fixed	Multi-Transaction Timer
10	X	X			Fixed	IDE Prefetch
11	X	X			Fixed	GPIO21 / GPIO23
12	X	X			Fixed	Alert On LAN* SEQ
13	X	X			Fixed	Alert On LAN* Power State

Errata (Sheet 2 of 3)

No.	Steppings				Status	ERRATA
	A0	A1	B0	B1		
14	X	X			Fixed	HLCOMP
15	X				Fixed	IDE PRD
16	X	X			Fixed	SMBus BYTE_DONE_STS
17	X	X			Fixed	Native Audio Bus Master Base Address
18	X	X			Fixed	Alert On LAN* SENDNOW
19	X	X			Fixed	Alert On LAN* Second Time-Out
20	X	X			Fixed	BOOT_STS
21	X	X			Fixed	Daylight Savings
22	X	X			Fixed	IDE Bus Master Concurrency
23	X	X			Fixed	USB_LEGKEY SMI Generation
24	X	X			Fixed	C2 State STPCLK# Deassertion
25	X	X			Fixed	AC97 Buffer
26	X	X			Fixed	PCI Bus Contention
27	X	X			Fixed	PAM Region Access Live-Lock
28	X	X			Fixed	Single DWord Loss
29	X	X			Fixed	RTC Oscillator
30			X		Fixed	IOAPIC_EN
31	X	X	X		Fixed	AC’97 PCM Out
32	X	X	X	X	NoFix	LPC Master Cycle Causes PCI Violation
33	X	X			Fixed	SUSCLK Noise
34	X	X	X	X	NoFix	ICH Does Not Generate a STOP on SMBus When a System Lockup Occurs
35			X		Fixed	RTC Update Logic
36	X	X	X	X	NoFix	TCO1_STS Bit 7
37	X	X	X	X	NoFix	USB Rise/Fall Matching
38			X	X	NoFix	SUSCLK Drop-Out
39	X	X	X	X	NoFix	STPCLK# Deassertion/WAK_STS
40	X	X	X	X	NoFix	Coin-Cell Detect Status Bit
41	X	X	X	X	NoFix	CPUSLP# Assertion
42	X	X	X	X	NoFix	PCI Bus Config Read
43	X	X	X	X	NoFix	SMBus Controller HOST_BUSY Bit
44	X	X			Fixed	BOISWR_STS in TCO I/O Space
45	X	X	X	X	NoFix	SMI# Glitch
46	X	X	X	X	NoFix	SMBus BYTE_DONE_STS
47	X	X	X	X	NoFix	CF9 Full Reset
48	X	X	X	X	NoFix	AC’97 “Missed Sample”
49	X	X	X	X	NoFix	PCI Parity Error Detection

Errata (Sheet 3 of 3)

No.	Steppings				Status	ERRATA
	A0	A1	B0	B1		
50	X	X	X	X	NoFix	Hub I/F Parity Error Response
51	X	X	X	X	NoFix	I/O APIC and C2/C3
52	X	X	X	X	NoFix	LPC Signaled Target Abort Generation
53	X	X	X	X	NoFix	USB Handshake
54	X	X	X	X	NoFix	Frequency Strap
55	X	X	X	X	NoFix	DMA Mode-0
56	X	X	X	X	NoFix	Parity Error
57	X	X	X	X	NoFix	Power-Button/CF9 Reset
58	X	X	X	X	NoFix	TRDY#
59	X	X	X	X	NoFix	AC97 Reset Race Condition
60	X	X	X	X	NoFix	I2C Read Command Issue
61	X	X	X	X	NoFix	SE0 During Resume Causes Disconnect
62	X	X	X	X	NoFix	PERR# Detection Issue
63	X	X	X	X	NoFix	PERR# Response Issue
64	X	X	X	X	NoFix	Delayed Transaction Timeout Bit Issue
65	X	X	X	X	NoFix	SMBus NACK and Proc_Call Issue
66	X	X	X	X	NoFix	SMBus Arbitration
67	X	X	X	X	NoFix	SMI Asserted after STPCLK# is Active and Stopgrant Received
68	X	X	X	X	NoFix	AC’97 Overrun FIFO Error Bit Not Set
69	X	X	X	X	NoFix	PCI Non-linear Addressing Erratum
70	X	X	X	X	NoFix	MW DMA Mode-1 Tdh Erratum

Specification Changes

No.	Steppings				SPECIFICATION CHANGES
	A0	A1	B0	B1	
1			X	X	VBIAS Capacitor Value
2			X	X	Internal Pull-Down Resistor Values
3			X	X	USB Status Register Default
4			X	X	AC’97 Wake From S5
5			X	X	APIC Data Register Access DW Only
6	X	X	X	X	INTRD_SEL Change
7	X	X	X	X	IRQPA and EOIR Register Removal
8	X	X	X	X	SERR_DTT_EN Change
9	X	X	X	X	AC’97 Register Access Requirements
10	X	X	X	X	I/O APIC Max Frequency Spec Change
11	X	X	X	X	Function Disable Requirements
12	X	X	X	X	Addition of Discard Timer Mode Bit
13	X	X	X	X	Addition of PWROK Min Deassertion Specification
14	X	X	X	X	DMA Mode-0 Not Supported
15	X	X	X	X	AC’97 Integrated Pulldown Control
16	X	X	X	X	USB Clock PPM
17	X	X	X	X	I/O APIC Arb ID Not Set When APIC CLK Not Running

Specification Clarifications

No.	SPECIFICATION CLARIFICATIONS
1	Intruder_Detect Bit Behavior
2	AC_SDOUT Internal Pull-Down Strength
3	AD[31:0] Value during PCI Special Cycles
4	FREQ_STRAP[3:0] Behavior
5	PCI Cycles to LPC or FWH Devices
6	INTRD_DET Bit Behavior
7	NEWCENTURY_STS Behavior
8	Parity Error Response Enables ICH Parity Generation
9	Delayed Transaction Discard Timer Clarification
10	NO_REBOOT Strap Clarification
11	HST_STA Register INTR Bit Clarification
12	SMBus BYTE_DONE_STS Clarification
13	PCI Underrun Behavior
14	Virtual Wire Mode B Usage on ICH
15	PCI Latency Clarification
16	GPE Event Handling under ACPI OS
17	USB Overcurrent Detection In Sleep States
18	Running Block Read With SECOND_TO_STS Set
19	DMA Clarification
20	USB Run/Stop Bit Clarification
21	RTC SET Bit Clarification
22	End Of SMI Bit
23	SMBus Wake
24	32 Clock Retry Enable Clarification
25	LPC LPCPD# Protocol Clarification
26	PCI Master Clarification

Documentation Changes

No.	DOCUMENTATION CHANGES
1	PCI Device Revision ID Table
2	PIO4 Maximum Performance
3	THRM# Signal Correction
4	SCI Level Correction
5	Ultra ATA33 Operation Correction
6	Figure Correction
7	PBUS_NUM Is Read Only
8	TCO_INT_SEL Description Correction
9	RTC Description Correction
10	PROC_CNT Bit Correction
11	Correction to Table 6-3
12	RTCRST# Jumper Clarification
13	Internal Pull-Up Resistor Value
14	IDE PRD Alignment
15	SUSCLK During RSMRST# Assertion
16	TCO Corrections
17	APM I/O Decode Correction
18	Memory Map Table Change
19	SMBus Host_Busy Correction

Identification Information

Markings

ICH Stepping	S-Spec/QDF	Top Marking	Notes
A0	Q725	Q725	Engineering Sample
A1	Q743	Q743	Engineering Sample, Ball G3=VSS
B0	Q781	Q781	Engineering Sample, Ball G3=VSS
B0	Q761	Q761	Engineering Sample, Ball G3=VSS
B0	Q762	Q762	Engineering Sample, Ball G3=VSS
B0	Q822	Q822	Engineering Sample, Ball G3=N/C
B0	SL38Q	SL38Q	Production, Ball G3=N/C
B1	Q813	Q813	Engineering Sample
B1	Q814	Q814	Engineering Sample
B1	Q815	Q815	Engineering Sample
B1	SL3MA	SL3MA	Production
B1	SL3Z2	SL3Z2	Production TSMC Fab
B1	Q942	Q942	Engineering Sample TSMC Fab
B1	Q943	Q943	Engineering Sample TSMC Fab

ICH0 Stepping	S-Spec/QDF	Top Marking	Notes
A0	Q678	Q678	Engineering Sample
A1	Q723	Q723	Engineering Sample
B0	Q778	Q778	Engineering Sample, Ball G3=VSS
B0	Q821	Q821	Engineering Sample, Ball G3=N/C
B0	SL38J	SL38QJ	Production, Ball G3= N/C
B1	Q810	Q810	Engineering Sample
B1	Q811	Q811	Engineering Sample
B1	Q812	Q812	Engineering Sample
B1	SL3MB	SL3MB	Production
B1	SL3N2	SL3N2	Production

Errata

1. Resume Well

Problem: Some internal ICH resume well logic is powered from the core well.

Implication:

1. SLP_S3# and SLP_S5#
   SLP_S3# and SLP_S5# will not be driven by the ICH when core power is not applied. In most system designs, SLP_S3# or SLP_S5# is used to turn on the power supply's core well. When the system is first plugged in, the resume well comes up first and the ICH is supposed to drive SLP_S3# and SLP_S5# high to turn on the power supply's core well. Since the ICH never drives SLP_S3# and SLP_S5# high, the power supply never turns on.

   When the system goes into suspend the ICH asserts SLP_S3# and SLP_S5# low. If these signals are used to turn off the power supply, when core power is removed the ICH stops driving SLP_S3# and SLP_S5#.
2. FREQ_STRAP bits
   In Section 8.1.22, the FREQ_STRAP bits in the General Status Register (D31:F0, offset D4h bits 11:8) are powered by the core well instead of the RTC well. When core power is removed, these bits return to their default values.
3. RTC_PWR_STS bit
   In Section 8.8.1.3, the RTC_PWR_STS bit in the GEN_PMCON3 Register (D31:F0, offset A4, bit 2) is powered by the core well instead of the RTC well. When core power is removed, these bits return to their default values.

Workaround: Attach a weak pull-up (4.7 kΩ to 10 kΩ) to 3.3 V Standby for SLP_S3# and SLP_S5#. **This workaround will allow a system to power-up.** However, this workaround will not allow the system to stay in the S3-S5 states. The ICH asserts SLP_S3# (upon entering S3-S5) and SLP_S5# (upon entering S4-S5). When core power is removed the ICH will stop driving SLP_S3# and SLP_S5# low and the pull-up resistor will force the power supply to turn back on. When the power comes back on, the ICH will assert SLP_S3# and SLP_S5# low. This causes the power supply to cycle on and off indefinitely. To power off the system with this workaround it is recommended to turn off or unplug the power supply.

Note that this workaround does not fix the FREQ_STRAP or RTC_PWR_STS bits.

Status: This erratum was fixed in the A-1 stepping.

2. USB Port Enable and Port Status Bits

Problem: A boundary condition can occur while the ICH is entering a S3-S4 suspend state and a USB wake event occurs. When the ICH resumes from S3 or S4 the port enable and port status bits are no longer set. These bits are in the Port Status and Control Register documented in Section 10.2.7 of the datasheet.

Implication: The OS or application may no longer be able to use USB.

Workaround: None

Status: This erratum was fixed in the B-0 stepping.

3. SMBus Time-Out

Problem: If software does not clear the BYTE_DONE_STS bit during an SMBus block read command, the ICH continues to hold the SMBus. Per the SMBus specification, the ICH should time-out after 25 ms and release the bus. However, the ICH will not time out and does not release the bus.

Implication: If an interrupt occurs in the middle of an SMBus block read command, S/W will not be able to clear the BYTE_DONE_STS bit and there could be a long interrupt latency (<25 ms) before the BYTE_DONE_STS bit can be set. There is no major side affect from this erratum unless there is a SMBus master that requires to use the SMBus within the 25 ms specification.

Workaround: None

Status: This erratum was fixed in the B-0 stepping.

4. Alert On LAN* Busy Bit

Problem: When sending Alert On LAN* messages, the ICH SMBus state machine clears the SMBus HOST_BUSY bit (Section 11.2.1, Host Status Register, offset 00h, bit 0) when there are still 9 bits of data left to transfer.

Implication: If S/W thinks that the SMBus is not busy and writes a new command to the SMBus controller, this new command may cause data corruption by over-writing the data that has not yet been transferred.

Workaround: After a wake event, ensure a minimum delay of 5 ms before using the SMBus.

Status: This erratum was fixed in the B-0 stepping.

5. AC’97 Cold Reset

Problem: An AC’97 Cold Reset does not clear the Codec Access Semaphore bit and Read Completion bit. The Cold Reset bit is located at bit 1 of the Global Control Register and is documented in Sections 12.2.8 and 13.2.8 of the datasheet. The Codec Access Semaphore bit is located at bit 0 of the Codec Access Semaphore Register and is documented in Sections 12.2.10 and 13.2.10 of the datasheet. The Read Completion bit is located at bit 15 of the Global Status Register and is documented in Sections 12.2.9 and 13.2.9 of the datasheet.

Implication: The AC’97 driver will not be able to use the AC’97 controller if it thinks the Codec is busy with the Semaphore bit set.

Workaround: Do not issue a Cold Reset command when the Codec Access Semaphore bit or the Read Completion bit is set. All I/O transactions should be completed before attempting to issue a cold reset. The AC’97 controller will clear the Semaphore bit upon completion of the I/O transaction. The AC’97 driver should also clear the read completion bit before issuing a cold reset.

Status: This erratum was fixed in the B-0 stepping.

6. PCI Peer-to-Peer Prefetchable Memory

Problem: If a PCI master is performing a peer-to-peer transaction with prefetchable memory enabled the ICH may claim the cycle and send it up the hub interface.

Implication: With both the ICH and a PCI target claiming the same cycle, there could be PCI bus contention.

Workaround: The BIOS should program all PCI devices that do peer-to-peer cycles to use non-prefetchable memory.

Status: This erratum was fixed in the B-0 stepping.

7. SMI on BIOS Access

Problem: As specified in the BIOS_CNTL Register (Section 8.1.12, D31:F0, offset 4Eh), the ICH will generate an SMI# if the BLE bit is a "1", the BIOSWE bit is "0", and a write is performed to the BIOS space. However, the ICH is also generating SMIs on reads to the BIOS space.

The BIOS space is enabled via the FWH Decode Enable Register (Section 8.1.30, D31:f0, offset E3h). For example, when the FWH_F0_EN bit is set, the ICH enables two ranges; FFF00000h–FFF7FFFFh and FFB00000h–FFB7FFFFh. The ICH should only generate SMIs upon reads to the higher range. However, the ICH is generating SMIs upon reads or writes to both ranges.

Implication: If BLE is set to "1" and BIOSWE is cleared to "0", the Intel Security Driver will report that the Random Number Generator is unavailable.

Workaround: BIOS must set the BIOSWE bit and clear the BLE bit.

Status: This erratum was fixed in the B-0 stepping

8. SCI from SMBus

Problem: When the SMBus controller is programmed to generate an SMI# by setting the SMB_SMI_EN bit in the Host Configuration Register (Section 11.1.13, D31:F3, offset 40h, bit 1) and the SCI_EN bit is set (Section 8.8.3.3, PMBASE + 04h, bit 0), the ICH generates an SCI instead of an SMI#. The SMBus controller should never generate an SCI.

Implication: If the SMBus controller generates an SCI, the OS will not be able to identify the source of the SCI and will not be able to clear the source. Since an SCI was generated instead of an SMI, the SMM handler will not be able to service the request.

Workaround: Software should not set the SCI_EN bit and the SMB_SMI_EN bit at the same time.

Status: This erratum was fixed in the A-1 stepping.

9. Multi-Transaction Timer

Problem: If the following three events occur in order:
1) The ICH Multi-Transaction Timer Register is set to 0 (Section 7.1.26, D30:F0, offset 70h)
2) A processor-to-PCI locked read cycle is master aborted
3) A second processor-to-PCI read cycle occurs
The ICH is unable to decode PCI master cycles during the second processor read cycle.

Implication: If a PCI master cycle to memory occurs during event 3, the cycle will be master aborted.

Workaround: Software should set the MTT Register to a non-zero value. A recommended setting is 40h.

Status: This erratum was fixed in the B-0 stepping.

10. IDE Prefetch

Problem: While executing a PIO IDE Read Sector(s) or Read Multiple command with PIO prefetching enabled, a read of a non-Data Register (such as ALT STATUS Register) may cause the ICH PIO pre-fetch counter to increment incorrectly, since it should only increment on data transfers.

Implication: The incorrect count causes ICH to confuse sector boundaries, resulting in invalid data being placed in memory. This erratum was observed during validation testing executing special test software.

Workaround: Do not perform non-data register reads while an IDE PIO transfer is taking place or disable IDE PIO prefetching.

Status: This erratum was fixed in the B-0 stepping.

11. GPIO21 / GPIO23

Problem: During PCIRST#, GPIO[21] should be high and GPIO[23] should be low. However, with A-1 silicon, during PCIRST# GPIO[21] is low and GPIO[23] is high-z.

Implication: External logic may not function correctly if it expects the specified values on these GPIOs during PCIRST#

Workaround: None

Status: This erratum was fixed in the B-0 stepping.

12. Alert On LAN* SEQ

Problem: If the TCO timer expires twice during boot an Alert On LAN event is generated. In the Alert On LAN message the SEQ number does not increment from 0 to 1.

Implication: The LAN controller may report misinformation.

Workaround: None

Status: This erratum was fixed in the B-0 stepping.

13. Alert On LAN* Power State

Problem: If the system stays in a pre-boot state (CPURST# asserted) for more than 4 ms an Alert On LAN event is generated. In the Alert On LAN message the System Power State information reports 00b which indicates the system is in a G0 state. In this condition the System Power State should report 11b which indicates the system is in a pre-boot state.

Implication: The LAN controller may report misinformation.

Workaround: None

Status: This erratum was fixed in the B-0 stepping.

14. HLCOMP

Problem: Compensation circuitry in the ICH only samples the HLCOMP pin once at power-up. This circuitry samples the HLCOMP pin before most designs power the 1.8 V rail. Designs that implement a 40 Ω pull-up to 1.8 V on the HLCOMP pin will not be correctly sampled by the ICH.

Implication: The system may not boot without deasserting PWROK via the RESET button.

Workaround: Replace the 40 Ω ± 1% pull-up on HLCOMP with a 107 Ω ± 4% pull-up or a 110 Ω ± 1% pull-up. Change the pull-up source from 1.8 V to the 3.3 V core supply.

Status: This erratum was fixed in the B-0 stepping.

15. IDE PRD

Problem: Under certain circumstances, writes from disk to memory may result in same data word corruption. The corruption occurs when the following four events occur in order:

1. A burst is interrupted leaving 1 or 2 words remaining in the current PRD.
2. The burst starts up again and the disk drive sends 4 or 5 words more than the PRD needs.
3. A PIO cycle occurs on the primary or secondary IDE channel between the time when the burst initiates and the next descriptor is fetched from the PRD table, interrupting the transfer.
4. The next PRD has a count greater than 1 DWord.

Implication: The four events occur in order, data corruption can occur.

Workaround: None

Status: This erratum was fixed in the A-1 Stepping

16. SMBus BYTE_DONE_STS

Problem: When performing a Block Write command on the SMBus the BYTE_DONE_STS bit (Host Status Register, bit 7, Section 11.2.1) is set to 1 before the data byte has been sent. This condition happens on every byte of the block write transfer.

Implication: As specified, the ICH will generate an interrupt whenever the BYTE_DONE_STS bit gets set. The interrupt service routine (ISR) for a block write command will write the next byte of data in the block. If the ISR is fast enough to write the next byte of data before the previous byte has been sent, the ISR will overwrite and corrupt the SMBus block write transfer.

Workaround: BIOS could add a 5 ms delay in the SMBus ISR to allow enough time for the byte write transaction to complete before attempting the next byte write transfer.

Status: This erratum was fixed in the B-0 stepping.

17. Native Audio Bus Master Base Address

Problem: When bit 6 of the Native Audio Bus Mastering Base Address Register (Section 12.1.11) is set to 1, accesses to the Native Audio Bus Master Control Registers (Table 12-3) followed by accesses to the primary codec will only go to the secondary codec.

Implication: Since the primary codec cannot be accessed it can appear non-functional and the system audio could stop.

Workaround: System BIOS must ensure that bit 6 of the Native Audio Bus Mastering Base Address Register is set to 0.

Status: This erratum was fixed in the B-0 stepping.

18. Alert On LAN* SENDNOW

Problem: After completing a software initiated Alert On LAN message through writing a 1 to the SENDNOW bit (TCO1 Control Register, Section 8.9.8, bit 10), the ICH does not always clear the SENDNOW bit back to 0.

Implication: If the ICH does not clear the SENDNOW bit, software will no longer be able to initiate any further Alert On LAN messages to the LAN controller.

Workaround: None

Status: This erratum was fixed in the B-0 stepping.

19. Alert On LAN* Second Time-Out

Problem: If the system is in a G2 state (S5) and the SECOND_TO_STS bit (TCO2 Status Register, Section 8.9.7, bit 1) is set to 1, the ICH will send an Alert On LAN event message with the CPU Missing Event status bit set to 1.

Implication: The ICH is not supposed to send out event messages under these circumstances. These unexpected messages may cause the LAN controller to report misinformation.

Workaround: Add system BIOS code to continuously reset the TCO timer so that the SECOND_TO_STS bit does not get set.

Status: This erratum was fixed in the B-0 stepping.

20. BOOT_STS

Problem: When the SECOND_TO_STS bit is set and the system resets due to PCIRST#, the ICH is not setting the BOOT_STS bit (TCO2 Status Register, Section 8.9.7, bit 2). However, the ICH is properly booting into safe mode.

Implication: The BIOS may use the BOOT_STS bit to determine if a boot successfully completed or went into safe mode due to an illegal multiplier. If so, the BIOS will not be able recognize the instances when the ICH forced a safe mode multiplier. This could cause the system to run much slower than its optimal speed.

Workaround: BIOS could check processor registers to determine if the system booted in safe mode or not.

Status: This erratum was fixed in the B-0 stepping.

21. Daylight Savings

Problem: If the last Sunday in October is 30th or 31st, and the Daylight Savings Enable bit (RTC Register B, Section 8.6.2.2, bit 0) is set to 1, the ICH will not correctly adjust the time back one hour from 1:59:59am to 1:00:00am.

Implication: The time reported from the ICH RTC will be off one hour after daylight savings time change. The first manifestation of the erratum occured on October 31, 1999.

Workaround: There are three possible work-arounds for this erratum.

1. If the system is left on during the failing condition, Microsoft* Operating Systems will automatically correct the time.
2. Any user can update the time manually either using the OS date and time function or through the BIOS setup.
3. The BIOS can be designed to workaround the erratum by re-writing the time in the CMOS.

Status: This erratum was fixed in the B-0 stepping.

22. IDE Bus Master Concurrency

Problem: During periods of high bandwidth, concurrent I/O traffic to memory, the ICH may corrupt data being transferred from IDE to memory. This problem only occurs with Ultra ATA and BMIDE transfers.

Implication: This erratum could lead to a system hang or data corruption.

Workaround: Use IDE PIO Mode instead of Ultra ATA or BMIDE.

Status: This erratum was fixed in the B-0 stepping.

23. USB_LEGKEY SMI Generation

Problem: If system BIOS is using the USB_LEGKEY register (D31:F2, C0h) to generate SMIs on reads or writes to ports 60h or 64h, reads from ports 60 or 64h will be blocked from the keyboard by the ICH A-1 (and A-0) and 00h will be the data returned.

Implication: This will result in the BIOS Legacy Keyboard operation not functioning properly, if system BIOS is using the port 60/64h trap SMIs for its implementation. Also, this issue may result in a system hang during POST, if waiting for particular data when reading from ports 60 or 64h.

Workaround: It is recommended to use a non-trap method of generating an SMI. There are several methods that can be used to accomplish this. For example, by using SMIBYUSB (SMI caused BY USB interrupt) bit (D31:f2, C0h [12]) in conjunction with the IOC (Interrupt On Complete) bit (D31:F2, Base+04h[2]), System BIOS can determine when port 60 or 64h reads and writes occur.

Status: This erratum was fixed in the B-0 stepping.

24. C-2 State STPCLK# Deassertion

Problem: When software initiated throttling or THRM# throttling is enabled as described in the datasheet Section 5.12.4.3, and system software performs a level 2 read (D31:F0, PMBase+14h) to place the system into C2 state, the ICH A-1 may not deassert STPCLK#.

Implication: System hang may result.

Workaround: None.

Status: This was fixed in the B-0 stepping.

25. AC’97 Buffer

Problem: The ICH A-1 AC’97 controller may send out data from the lower word of the first data fetch in the new buffer instead of the upper word because the buffer is odd word aligned. This problem is only seen on the Modem Out channel if the following conditions occur in order:

1. The ICH A-1 AC’97 Controller reaches the Last Valid Buffer.
2. The Last Valid Index is updated some time after all data for the current buffer has been fetched, but before it has all been transmitted out.
3. New buffer address starts at an odd word location.
4. The ICH A-1 AC’97 Controller fetches the new Buffer Descriptor and then fetches the data for the new buffer before the data for the previous buffer (which was the last valid buffer) is sent out on the link.

Implication: This will result in one extra sample being sent out on the link before the new buffer’s samples are sent out. Modem drivers that incorporate an error-checking algorithm will see this issue.

Workaround: None.

Status: This was fixed in the B-0 stepping.

26. PCI Bus Contention

Problem: The ICH A-1 can cause contention with a PCI bus master under a specific circumstance, listed below:

1. The processor attempts a 2-DWord locked cycle sequence to PCI where the read is master or target aborted.
2. The ICH grants the PCI bus to a PCI master, which is supposed to perform a read operation to main memory.
3. The ICH completes the write cycle from step 1 above.

During step 2, the ICH can drive the bus at the same time it has granted the bus to the PCI master.

Implication: This contention could cause the address or cycle type to be incorrect, possibly resulting in data corruption.

Workaround: None.

Status: This erratum was fixed in the B-0 stepping.

27. PAM Region Access Live-Lock

Problem: If there is a lock sequence to a PAM region in memory that is read-only, the ICH incorrectly decides when to grant a retried write request and when to accept a burst write cycle. As a result, it is possible for burst writes to receive a grant and then be retried by the ICH. This also happens with unlocked cycles to PAM regions of memory. In this condition, the ICH is waiting for an up-bound cycle to be completed on the hub interface.

Implication: This can cause live-lock, where the grant alternates between the two PCI masters, shutting out all others.

Workaround: None.

Status: This erratum was fixed in the B-0 stepping.

28. Single DWord Loss

Problem: Any single DWord cycle from the hub interface may be lost by the ICH if this cycle aligns with the completion of the first DWord of a double DWord I/O read cycle.

Implication: This could cause the system to hang.

Workaround: None.

Status: This erratum was fixed in the B-0 stepping.

29. RTC Oscillator

Problem: The required external RTC circuitry specified in Section 2.18.3, previously required a 2.2 nF capacitor in series with a 1 kΩ resistor between VBIAS and the battery terminal. Random charging of this capacitor prevents current from flowing to VBIAS when the battery is first installed. The ICH RTC oscillator is a self biasing circuit, so without current flowing to VBIAS the oscillator will not start oscillating.

Implication: If no current flows to VBIAS the RTC oscillator will not start oscillating. Without the RTC oscillating, the ICH will not be able to deassert PCIRST#. With PCIRST# asserted, the MCH, processor, and all PCI devices are held in reset which prevent the system from booting.

Workaround: There are three possible workarounds for this erratum. Any one will start the RTC oscillator.

1. After installing the VccRTC battery, momentarily connect the VBIAS signal to VccRTC. This will start the oscillator.
2. Add a 60–100 MΩ resistor between VBIAS and VccRTC. This workaround allows a small leakage current that will prevent VBIAS from remaining at 0 V.
3. Replace the 2.2 nF capacitor with a 0.047 µF capacitor. Note the B-0 (and newer) steppings of the ICH **requires** a 2.2 nF capacitor. The 2.2 nF capacitor, for B-0 (and newer) steppings provide improved dT/dt performance while maintaining the same filter performance.

Status: This erratum was fixed in the B-0 stepping.

30. IOAPIC_EN

Problem: When the ICH IOAPIC receives Interrupt Request and sends Flush Request to the north-hub, the north-hub flushes its buffers and sends a Flush Acknowledge to the ICH. Under heavy PCI traffic, the ICH may request the northbridge to retry the Flush Ack cycle. In this case, the ICH erroneously sends the Flush Ack to the IOAPIC.

Implication: This leads to two scenarios that cause the IOAPIC to hang which eventually leads to system hang:

1. The IOAPIC could receive multiple Flush Acks causing the IOAPIC to hang, and
2. The northbridge could lose a Flush request causing the IOAPIC to hang because it never received a flush.

Workaround: Disable the IOAPIC (D31:F0; D1h[0]) and do not build a MP table.

Status: This will be fixed in the B-1 stepping.

31. AC’97 PCM Out

Problem: The AC’97 controller in the ICH will incorrectly insert one extra sample pair on the PCM Out channel under a specific condition. The sequence and conditions are described below.

1. The last buffer in the current buffer list has been completely fetched and an odd number of buffers in this list has a length that is not a multiple of four.
2. After the last buffer has been completely fetched, software updates the LVI.
3. The first buffer in the next current buffer list has a length of exactly six samples.

When this occurs, the ICH will insert one extra sample pair, sent six frames (120 µs) earlier, between the last buffer of the old list, and the first buffer of the new list.

Implication: The PCM Out channel will distort the intended sound. Since the time between these two buffers represents approximately 20 µs, the added sample should not be noticeable to the human ear.

Workaround: None.

Status: This erratum is intended to be fixed in the B-1 stepping of the ICH/ICH0.

32. LPC Master Cycle Causes PCI Violation

Problem: If there are LPC master devices in a system, and the DMA Collection Buffer (DCB) is enabled (D31:F0; D3h[2]), unused byte lanes of the AD[31:0] bus may change on the second PCI clock of the data phase (IRDY# asserted), for LPC master initiated memory write cycles.

Implication: The ICH will violate the PCI specification under this specific condition. No failures are expected or have been reported to date as a result of this since this occurs only on unused byte lanes. PCI parity generation is still performed correctly - based on actual AD values, hence the target will still see the correct PCI parity. An implication of disabling the DCB (the workaround) is reduced performance when LPC or DMA traffic occurs. This could cause issues for AC’97 and USB since they may suffer from higher latencies.

Workaround: Disable the DCB.

Status: There are currently no plans to fix this erratum.

33. SUSCLK Noise

Problem: Noise from the motherboard may be coupled to internal clock signals on the ICH RTC under heavy PCI loading.

Implication: There are several possible implications. A violation of SUSCLK min high and/or min low time specification may occur. The RTC date and time may be inaccurately reported, and SMBus protocol violations may be encountered as a result of this erratum. Any combination of these may be observed.

Workaround: OEMs should test for SUSCLK (ICH only) min high and low time violations under heavy PCI load with the production B-0 ICH. If min high time or min low time specification violations are detected, contact an Intel Field Representative. Intel has published an application note describing how to determine if the problem exists and how to correct the behavior. This document is *AP-668, ICH Real Time Clock (RTC) Signal Quality Analysis* (document number 292226). AP-668 is available from Intel Literature or your Intel Field Representative.

Status: The ICH/ICH0 B0 production steppings will have a package change to improve noise immunity. This change is identified in Specification Change #1. Further improvements will be made in the ICH/ICH0 B-1 stepping.

34. ICH Does Not Generate a STOP on SMBus When a System Lockup Occurs

Problem: If the BYTE_DONE_STS (D31:F3; SMB I/O 0h [7]) has not been cleared by software, due to a system lock-up, for a SMBus Block Read transaction, the ICH will not discard the cycle after the 25 ms timer expires. The ICH does not drive a STOP, but instead may still pull the SMBus clock low for up to 30 ms. This may also cause the slave device to lock-up. This condition does not happen when Reboots are disabled.

Implication: The SMBus will lock-up because either the Host controller or the slave device is locked up, or both. When the system is rebooted, the SMBus may still be locked-up and may prevent proper booting of the system.

Workaround: If the system lock-up is due to the operating system locking, that is, where it is not processing interrupts and the SMBus handler is using the interrupt (rather than an SMI), then the SMI handler could attempt to recover by doing an SMBus KILL command, or possibly even attempting to complete the block read. If the system lock-up is due to a hardware lock-up, such as a PCI livelock condition, then the SMI handler can’t execute and the system will be rebooted as evidenced by SECOND_TO_STS (D31:F0; TCOBase+6h[1]) being set. In this case, the SMBus host controller in the ICH has been reset, and a KILL command may not unlock the SMBus slave device that is still waiting to complete the block read. In this case an added method could be used to force a stuck slave to release. The GPO[28:27] may be connected to the respective SMBus data and clock signals. BIOS can manually send out a “STOP sequence” on these lines which should unlock the slave device. A full power cord removal and re-installation is a last resort which will reset all SMBus devices and allow a reboot.

Status: There are currently no plans to fix this erratum.

35. RTC Update Logic

Problem: The Update In Progress bit (UIP) of the ICH/ICH B-0 stepping may be driven low for 400–700 ps during a 1 second update cycle. This has been observed running synthetic RTC specific stress tests.

Implication: This may result in reading incorrect data from the RTC. This can be manifested in several ways, such as year data getting copied into all data fields of the date/time registers.

Workaround: BIOS workaround utilizing an SMI patch is available. Contact your BIOS vendor or system OEM for more information.

Status: This erratum is intended to be fixed in the B-1 stepping of the ICH/ICH0.

36. TCO1_STS Bit 7

Problem: The ICH inadvertently sets the TCO1_STS bit 7 status bit and correspondingly generates an SMI, if RTC BCD mode is used, on all 199x to 199x+1 transitions. This status bit operates correctly for the 1999 to 2000 rollover.

Implication: The year may not be correctly reported by BIOS, or some Year 2000 test software may fail, for year rollovers prior to 1999 (1990-1991 ... 1998-1999). In this case, the symptom may show up only if the system is re-booted.

Workaround: Add an additional code check in the SMI handler for the TCO1_STS[7] as follows. If the year is not equal to "00", then clear the TCO1_STS[7] and exit SMI, otherwise, the SMI handler should proceed accordingly to increment its internal storage area.

Status: There are currently no plans to fix this erratum.

37. USB Rise/Fall Matching

Problem: The USB Specification defines a rise/fall time matching (Trfm) which is calculated by dividing rise time by fall time (Tr/Tf). The specification for a full speed device is 90% minimum and 110% maximum. The ICH/ICH0 does not meet this specification.

Implication: No failures have been reported in system validation testing to date as a result of this erratum. the ICH/ICH0 does meet the required output signal crossover voltage specification (Vcrs).

Workaround: None.

Status: This will not be fixed in the ICH/ICH0.

38. SUSCLK Drop-Out

Problem: An internal oscillator "kicker" circuit in the ICH B-step devices, employed when AC power is cycled (off-to-on), can change the on-board crystal circuit's DC offset. As a result, SUSCLK can incur initial startup delay and drop-outs within the first 1.2 seconds after power cycling. The occurrence and width of the SUSCLK startup delay and drop-outs varies from board to board.

Implication: RTC time loss corresponding to SUSCLK startup delay or drop-out, or 2) ICH t201 and SUSCLK AC timings may be violated.

Workaround: Increase the VBIAS cap C1, datasheet figure 2-1, from 2200 pF to 0.047 µF, or use an external oscillator (refer to AP-668 mentioned in Errata 33).

Status: This will not be fixed in the ICH/ICH0.

39. STPCLK# Deassertion/WAK_STS

Problem: The WAK_STS bit (D31:F0;PM1_STS, bit-15) will still be clear during a brief (4–5 PCI clocks) deassertion of STPCLK# that occurs while transitioning from throttling to sleep state (SLP_EN=1), if the ICH receives the cycle to set the SLP_EN bit (D31:F0;PM1_CNT[13]) at approximately the same time STPCLK# asserts due to THRM# or software throttling.

Implication: Since STPCLK# deasserts briefly, the processor may have some time to execute cycles. If it (code) finds the WAK_STS bit clear, when code expects it to be set, it could assume that there is a hardware problem and react accordingly. Operating systems that do not rely on the first read of this WAK_STS being set, for example, that keep checking until it does read set, will not be affected by this behavior.

Workaround: There is a pre-existing requirement (in the ICH/ICH0 datasheet) to DISABLE software throttling prior to setting SLP_EN. For THRM# based throttling, code should not expect the WAK_STS to be set and should poll until WAK_STS=1.

Status: There are currently no plans to fix this erratum.

40. Coin-Cell Detect Status Bit

Problem: The ICH will not set the RTC_PWR_STS bit (D31:F0;GEN_PMCON_3[bit-2]), on insertion of a new coin-cell battery or on boot-up with a low voltage battery. The RTCRST# is synchronized to RTC clock (internal version of SUSCLK), which may not start before RTCRST# ramp up since the delay from RTC power ramp to RTCRST# ramp is set by an external RC circuit which is in the 10 ms range. The oscillator may take longer than this 10 ms, resulting in failure of this bit to set.

Implication: On boot-up after a coin-cell battery installation or with a low battery voltage, the RTC_PWR_STS bit may not be read as set, and BIOS may not know to perform CMOS initialization functions. In this case, chipset default configuration values will be used, which may result in reduced system performance or configuration issues until the CMOS values are properly restored by the end user using the BIOS Setup Menu.

Workaround: BIOS should perform a non-zero based checksum calculation on the CMOS memory to check for loss of CMOS data. Zero-based checksum calculations will result in a false OK check and BIOS will, therefore, not know that CMOS power was lost, so non-zero based checksum should be used. This workaround should be much like the standard implementation used for PIIX4E based systems.

Status: There are currently no plans to fix this erratum.

41. CPUSLP# Assertion

Problem: The ICH will assert CPUSLP# in response to the Stop Grant Acknowledge special cycle from the processor, for S1 states, sooner than the 100 BCLK processor requirement.

Implication: While to date, no reports of system failures have been reported, the implication is that there could be possible problems in the processor as a result of clocks being stopped too early. This could manifest itself in many ways. Failure modes could be reset of the processor, data corruption (due to invalid clocks), lockups (state machines getting invalid clock), etc.

Workaround: Either completely disable CPUSLP# assertion for S1 state by clearing D31:F0;A0h, bit-5, the CPUSLP_EN bit in the General PM Configuration 1 Register (GEN_PMCON_1) of the ICH. Note that this will increase AC power draw slightly.

Status: There are currently no plans to fix this erratum.

42. PCI Bus Config Read

Problem: On a PCI Bus 1 Config Read to Device 31, Function 7, Register 0, the AD[15:11] are not driven to ‘0’ on PCI during the address phase. Instead, AD[15:11] have random value. Note that this device address is special in that on a Config write to the same device/function/register, ICH would convert this type-1 config cycle to a PCI Special Cycle. This was found during SLS Simulations, not in a real system.

Implication: PCI Spec violation of AD[15:11] should be ‘0’. Unsure if any software would ever do a config read to this address (usually would be a config write to generate the PCI Special Cycle).

Workaround: None.

Status: There are currently no plans to fix this erratum.

43. SMBus Controller HOST_BUSY Bit

Problem: The Host Busy bit located at SMBase+0, bit-0 in the SMBus can clear before the correct data is loaded in the data register.

Implication: The wrong data may be read from the SMBus controller’s data register.

Workaround: The BIOS should use the INTR bit and the other error indicator bits in the SMBus Host Status register of SMBus I/O space to indicate the end of the operation before reading the SMBus data register.

Status: There are currently no plans to fix this erratum.

44. BIOSWR_STS in TCO I/O Space

Problem: The ICH is supposed to set the BIOSWR_STS bit (TCOBase+Offset 4h) to a 1 and generate an SMI# to indicate an illegal attempt to write to the BIOS. This generation of an SMI# is supposed to occur when an illegal attempt occurs in one of these scenarios:

- The BIOSWE bit (Device# 31, Function 0, Register 4E-4F, bit-0) is changed from a 0 to a 1 when the BLE bit is set (Device#31, Function 0, Register 4E-4F, bit-1).
- Or any write is attempted to the BIOS and the BIOSWE bit (Device#31, Function 0, Register 4E–4Fh, bit-0) is not set.
- But the ICH is generating and SMI# on writes as well as reads to both the upper and lower BIOS space instead of generating the SMI#s only to indicate an illegal write to BIOS space.

Implication: Enabling and testing BIOS security algorithms will cause this bit to get set and continuous SMI#s may occur even on reads to BIOS space.

Workaround: The BIOS should reset the BIOSWR_STS bit (TCOBase + Offset 4h) after the BIOS data has been fully shadowed and before enabling SMIs so that the SMIs on reads can be prevented.

Status: This erratum was fixed in the ICH B-0 stepping.

45. SMI# Glitch

Problem: The ICH will briefly deassert the SMI# signal if the SMI is a result of an interrupt from the USB Host Controller. The ICH will briefly assert SMI# for 1 PCI clock, followed by a 4 PCI clock deassertion, followed again by a complete assertion. This redirection of USB HC interrupt to an SMI is done during USB Legacy support.

Implication: Uni-processor systems are immune to this behavior and this erratum does not cause any adverse system action.

Depending on the SMM code structure, this behavior may cause DP/MP systems to lock-up. If one of the processors is not sufficiently through a resume, due to some unrelated previous SMI, then it won’t be able to latch the 2nd SMI, however, if the other processor had latched it, then this results in a synchronization problem where a processor is never able to get to the semaphore which keeps the other processor constantly checking for semaphore coherency (locks-up).

Workaround: Uni-processor Systems: No workarounds required.

Multi-processor Systems: Disable USB Legacy Support in MP/DP systems using the ICH.

Status: This erratum will not be fixed in the ICH0/ICH1. It is intended to be fixed in future implementations of IO Controller Hub components.

46. SMBus BYTE_DONE_STS

Problem: The ICH may inadvertently set the BYTE_DONE_STS bit, in the SMBus Host Status register, when the INTR bit for the n+1 interrupt is set at the end of a block write transfer of two or more bytes.

Implication: The ICH may hang the next attempted transfer causing an SMBus lockup if this BYTE_DONE_STS bit is not cleared before the next attempted transfer. This can result in a system lock-up requiring an AC power-cycle to recover from.

Workaround: Ensure that a “1” is written to the BYTE_DONE_STS bit at the same time the INTR bit (both in the SMBus Host Status register) is being cleared for the “n+1” interrupt, regardless of if interrupts are used or not.

Status: This erratum will not be fixed in the ICH0/ICH1.

47. CF9 Full Reset

Problem: There is a boundary condition in which a Full Reset via the CF9h register will not assert SLP_S3# and SLP_S5# for the specified 3 to 5 seconds. When software attempts a Full Reset via a CF9h write, the ICH will assert SLP_S3# and SLP_S5#. If PWROK# does not de-assert within 120 us, and the ICH does not see at least five PCI clocks after SLP_S3# and SLP_S5# are asserted, these signals will be asserted for only about 300 us (instead of the specified 3 to 5 seconds).

Implication: Some systems will not perform the expected power cycling in response to a Full Reset via a CF9h write. This may result in some motherboard devices not being properly reset.

Only affects systems in which both of the following conditions are true:

- PWROK does not deassert within 120 us after SLP_S3# assertion
- At least 5 PCI clock edges cannot be guaranteed after SLP_S3# assertion - This would be true if SLP_S3# connects to the clock generator's PWRDOWN# pin.

Workaround: Option 1: For systems that connect SLP_S3# to the clock generator's PWRDOWN pin, use an RC delay on SLP_S3# to ensure > 5 PCI clocks after the ICH asserts SLP_S3#.

Option 2: Use SLP_S3# to gate PWROK assertion to the ICH.

Status: This erratum will not be fixed in the ICH0/ICH1.

48. AC'97 "Missed Sample"

Problem: If an incoming data overrun occurs on AC Link within a 2 clock time window of an AC'97 write completion on the hub interface, the AC'97 controller will not set the FIFOE status bit.

Implication: Possibility of missing a single sample of incoming (Modem In, Mic In or PCM in) AC'97 data. For modem data, software may retry a packet transmit, resulting in a negligible performance hit (modem data packet retries occur frequently due to telephone line noise). Neither audio nor modem drivers are affected, since they do not implement overrun error handlers.

The boundary conditions required for this to occur are extremely unlikely. This issue has only been observed in artificial, highly stressed system test environments.

Workaround: None.

Status: This erratum will not be fixed in the ICH0/ICH1.

49. PCI Parity Error Detection

Problem: The ICH, as a target of a PCI write transaction, calculates parity and may improperly signal a parity error during master-initiated wait states.

Implication: The ICH's PCI bus Parity Error Response can not be used.

Workaround: Disable Parity Error Response in the Bridge Control Register (D30:F0;3Eh) by setting bit-0 to a value of '0'.

Status: This erratum will not be fixed in the ICH0/ICH1.

50. Hub Interface Parity Error Response

Problem: ICH does not generate a valid Hub Interface Parity signal when Parity Error Response is disabled for the Hub/PCI Bridge function.

Implication: System hang if Hub Interface Parity Error Response is enabled in the MCH when it is disabled in the ICH. This issue only affects platforms in which the MCH supports Hub Interface Parity checking (840, 850, 860, 870). The issue has only been observed with the Linux OS.

Workaround: Software must not disable Hub Interface Parity Error Response in the ICH while it is enabled in the MCH.

Status: This erratum will not be fixed. A patch for Linux will be posted to the “Kernel Web Site”.

51. I/O APIC and C2/C3

Problem: A boundary condition exists in which completion of delivery of an interrupt by the I/O APIC coincides with completion of a LVL2 or LVL3 read by the processor (to enter C2 or C3).

Implication: Interrupts that hit this boundary condition will not break the system out of C2/C3. The interrupt will not be processed until another break event occurs. In the case of IRQ8 (RTC interrupt), the ISR will not have cleared the RTC Alarm Flag bit, and no further RTC interrupts will be generated until a subsequent break event allows the ISR to complete. The result of this is time loss from the perspective of SW clocks that depend on RTC interrupts (the HW clock does not lose time).

Workaround: Systems that support the I/O APIC cannot support C2/C3. C2/C3 support is not necessary for desktop systems. Systems that must support C2/C3 (mobile systems) cannot support the I/O APIC.

Status: This erratum will not be fixed in the ICH0/ICH1.

52. LPC Signaled Target Abort Generation

Problem: If there is a downstream IO cycle followed by posted memory writes, both targeted towards PCI with different BE#s and with Delayed Transaction enabled (D31:F0;GEN_CNTL(D0-D3h):[1]), the ICH can erroneously set the STA bit (bit-11) in D31:F0;PCISTS configuration space even though there is not Target Abort on PCI bus. This has only been observed on DP systems.

Implication: The STA bit in D31:F0;PCISTS is incorrectly set. No NMI or SERR# will be generated due to the STA bit being set. Software which polls this STA bit may incorrectly indicate a Target Abort has occurred.

Workaround: There are two workarounds available:

1. Ignore the STA bit (bit-11) in D31:F0;PCISTS(06-07h) when Delayed Transaction is enabled. The D30:F0;SECSTS(1E-1Fh):[RTA (bit-12)] bit remains an accurate reflection of downstream cycles towards PCI that get Target Aborted.
2. Disable Delayed Transaction (which may induce a performance penalty on PCI).

Status: This erratum will not be fixed in the ICH0/ICH1.

53. USB Handshake

Problem: The ICH UHCI will fail to provide a handshake if it receives an incoming data packet where the CRC has five consecutive 1s in the five least significant bits of CRC and is immediately followed by an EOP for Bulk, Interrupt, Control, and Isoc transfers **only if** a K-state (remote wake from port specific selective suspend) is being signaled on the other port at the time of this EOP. This behavior, to date, has only been observed during artificial testing procedures.

Implication: USB devices may stall. The OS will attempt to recover, bit if it fails to recover, an error message will be displayed. The user may have to unplug then re-install the USB device.

Workaround: One of these two may be employed.

1. Do not use the port specific selective suspend feature of the ICH when there can be activity on the other bus. Global Suspend must be employed.
2. Do not allow USB peripherals to use remote wake feature (from selective suspend).

Status: This erratum will not be fixed in the ICH.

54. Frequency Strap

Problem: The ICH will not drive the processor frequency strap signals (A20M#, INTR, NMI, IGNNE#) to 1s as V_{CC} comes up prior to PWROK assertion. The ICH drives these signals to 0 instead.

Implication: No implication to qualification and production processors as they drive their own “start” straps internally. Implication to pre-qualification processors is that the “start ratio” may be set to an illegal value and the system may not boot.

Workaround: (Not required for systems using qualification or production processors)

1. Re-implement Legacy mux that drives 1s prior to PWROK, or
2. Place a ~ 1.2 ms RC delay on CPUPWRGD so that it asserts when ICH has the run ratio Frequency Strap values present.

Status: This erratum will not be fixed in the ICH.

55. DMA Mode-0

Problem: If a device on one of the IDE interfaces, such as the secondary channel, is operating in Multi-Word DMA Mode with compatible timings where the cycle time is 600 ns, while a device on the other interface (primary channel), is running in PIO mode, the IDE PIO prefetch buffer will inadvertently provide an extra piece of secondary channel data to the primary device, resulting in data corruption. This happens when DMAREQ is deasserted and a DMA transaction is running while a PIO transaction is outstanding on the other channel. Note that DMA Mode-0 is an unsupported mode of the ICH.

Implication: Systems configured in this manner may experience a situation in which the DMA IDE controller transfers incorrect data from the PIO configured device. Exactly how this manifests itself in a system depends on the system activity at that time.

Workaround: When BIOS is determining which mode(s) an IDE device is capable of, it must not set the DMA capable bits in the ICH if that device only supports Mode-0 DMA or slower. That device should be configured for a PIO mode instead.

Status: This will not be fixed in the ICH.

56. Parity Error

Problem: When enabled, a parity error will not be signaled or detected, via either D30:F0;1Eh bit-8 or D31:F0;06h bit-8 (SECSTS:[DPD] or PCISTS:[DPED]), if 2 Double-Word up-bound writes are followed by a down-bound read. Neither PERR# will be asserted, nor the indicated status bits will be set, resulting in failure to generate NMI or SMI. Note that the Hub I/F Parity Error detect mechanism remains functional.

Implication: This results in loss of indication to PCI target and system software when bad PCI data is received. This could result in data corruption to either memory or hard drive data.

Workaround: None.

Status: This will not be fixed in the ICH.

57. Power-Button/CF9 Reset

Problem: If the power-button is pressed (PWRBTN# is asserted) during a CF9 hard reset event (an IO write of 06h to CF9h), the ICH will behave as if a Power-button Override event has occurred and transition the system to the S5 state (off).

Implication: If a CF9 hard reset sequence is initiated while the power-button is depressed, the system may unexpectedly transition to the S5 state (off). The user will have to awaken the system by pressing the power-button.

Workaround: Software testing of the PWRBTN# status bit before attempting a CF9 hard reset sequence can reduce the boundary of this failure.

Status: This will not be fixed in the ICH.

58. TRDY#

Problem: The ICH/ICH0 may not assert TRDY# for more than 16 PCI clocks (up-to 32 clocks) after a bus masters assertion of FRAME# if there are no other PCI masters requesting the bus. This behavior is inconsistent with its Sub-class code of a PCI-to-PCI bridge device.

Implication: The ICH/ICH0 may not respond with a data phase within 16 PCI clocks, as required by the PCI specification. This has not been found to cause any functional problems. Since prior generations of chipsets were required to meet 32 clock requirements for a Host-to-PCI Bridge, PCI adapters that worked in these systems should also work in an ICH1/ICH0 based system.

Workaround: None.

Status: This will not be fixed in the ICH/ICH0.

59. AC97 Reset Race Condition

Problem: If an AC97 reset is initiated (via GLB_CNT:[1]) just as a new frame is starting, a race condition between AC_RST# asserting and AC_SDIN transitioning from “Ready=1” to ground, due to the reset, may cause an unexpected wake event (SMI) if AC97 wake events are enabled (via GPE0_EN).

Implication: This will result in an unexpected wake event (SMI) that may not be comprehended by the SMI handler resulting in repeated SMI’s.

Workaround: Earyly in enumeration disable AC97 wake events and re-enable them after enumeration. BIOS must disable SMI & PME generation when doing a reset to or powering down the codecs. BIOS should clear the related status registers and then re-enable SMI & PME generation.

Status: This will not be fixed in the ICH/ICH0.

60. I2C Read Command Issue

Problem: ICH/ICH0 uses HST_D0 register (Dev 31, F3, Offset 05h) as the byte count register instead of depending on the LAST_BYTE bit in the Host Control register (Dev 31, F3, Offset 02h:[5]) to end the transaction.

Implication: The transaction will stop pre-maturely if HST_D0 contains a number smaller than the intended transaction.

Workaround: No workaround for 10-bit addressing of I2C devices. Can use SM Bus read command for 7-bit addressing of I2C devices.

Status: This will not be fixed in the ICH/ICH0. The I2C read command is de-featured.

61. SE0 During Resume Causes Disconnect

Problem: A transient SE0 during an upstream resume signal from the USB peripheral to the ICH/ICH0 while the system is in the S3/S4 sleep states will cause the ICH/ICH0 to register a disconnect. This violates the USB Rev 1.1 specification.

Implication: The implication is operating system dependent. It can range from additional latency on a resume before the USB device is functional (after the resume), to the USB device no longer working (after the resume) - in which case a system reboot must be done to obtain functionality on that USB device. In all cases the rest of the system does resume.

Workaround: None.

Status: This will not be fixed in the ICH/ICH0.

62. PERR# Detection Issue

Problem: ICH/ICH0's Detected Parity Error (DPE) bit in SECSTS register (D30:F0, offset 1Eh: bit-15) and PCISTA register (D31:F0, offset 06h: bit-15) will get set when PERR# is asserted by external PCI devices. This issue was found during ongoing internal validation using a synthetic test environment and there have been no failures reported by customers.

Implication: DPE bit of SECSTS and PCISTA may get erroneously set.

Workaround: BIOS needs to clear DPE of SECSTS and PCISTA when these bits are set.

Status: This will not be fixed in the ICH/ICH0.

63. PERR# Response Issue

Problem: The ICH/ICH0's Parity Error Response Enable (PER) bit in Bridge_CNT register (D30:F0, offset 3Eh: bit-0) is disabled (by default), it will block PERR# from being asserted when data parity error is detected on PCI bus during LPC or legacy DMA master read cycles, or when ICH/ICH0 is the target for write cycles to Device 31 Function 0 and Function 3. This bit should only block PERR# from being asserted when PCI data parity error is detected during PCI-to-memory writes or CPU-to-PCI read cycles. This issue was found during ongoing internal validation using a synthetic test environment and there have been no failures reported by customers.

Implication: PERR# will not be asserted when PCI parity error is detected during LPC or legacy DMA master read cycles, or when ICH/ICH0 is the target for write cycles to Device 31 Functions 0 and 3.

Workaround: BIOS needs to set PER of Bridge_CNT when the parity error detection is supported on LPC or legacy DMA.

Status: This will not be fixed in the ICH/ICH0.

64. Delayed Transaction Timeout Bit Issue

Problem: After a delayed transaction has been serviced the discard timer is not reset, which incorrectly times out. This timeout sets the SERR# Due to Delayed Transaction Timeout bit (D30:F0;92:bit-1). This may result in generation of SERR# based NMIs if the SERR# enable on Delayed Transaction Timeout bit (D30:F0;90h:bit-1) is set to '1' by software of system BIOS.

Implication: This erratum may cause excessive NMIs to occur which impacts system performance.

Workaround: System BIOS must clear D30:F0;90h:bit-1 to '0' if the delayed transaction based SERR#s should not be generated on the platform.

Status: This will not be fixed in the ICH/ICH0.

65. SMBus NACK and Proc_Call Issue

Problem: ICH/ICH0's SMBus controller fails to respond after being NACKed on the 4th data phase when using the SMBus command Process Call (with or without the I2C enabled).

Implication: This issue may lock-up the SMBus controller and cause the system to stop responding. This issue was found during ongoing internal validation using a synthetic test environment and there have been no failures reported by known commercial applications.

Workaround: None.

Status: This will not be fixed in the ICH/ICH0.

66. SMBus Arbitration

Problem: The ICH/ICH0 will not detect a bus collision when attempting to STOP at the end of a SM Bus transaction as a master. If there is another external bus master attempting to access the bus at the same time and wins the arbitration during STOP bit, the ICH/ICH0 may not set the Bus Error bit.

Implication: A master attempting a transfer that had actually "lost" may think that its transaction was completed.

Workaround: None.

Status: This will not be fixed in the ICH/ICH0.

67. SMI Asserted after STPCLK# is Active and Stopgrant Received

Problem: The ICH/ICH0 should latch SMI when STPCLK# is active. When USB UHCI is active and an SMI is generated by the UHCI, logic does not latch the interrupt with STPCLK#, thus SMI goes low when the processor is in sleep state.

Implication: SMI could be lost if the processor is in a sleep state.

Workaround: None.

Status: This will not be fixed.

68. AC'97 Overrun FIFO Error Bit Not Set

Problem: The ICH/ICH0 may not set the FIFO Error Bit in the Input Status register after an overrun error occurs on a highly stressed system in the vicinity of the end of an AC'97 input stream. The bit affected depends on what stream causes the error:

- PCM IN - PISR(D31:F5;NABMBAR+06h:bit-4)
- Mic IN - MCSR(D31:F5;NABMBAR+26h:bit-4)
- Modem IN - MISR(D31:F6;MBAR+06h:bit-4)

Implication: Driver vendors typically do not use this status bit in their production drivers.

Workaround: None.

Status: This erratum will not be fixed.

69. PCI Non-linear Addressing Erratum

Problem: If a PCI Memory Read Multiple or Memory Read Line transaction falls on the last DW of a 32 byte cache line boundary and non-linear addressing (cache-line wrap mode) is used, the ICH/ICH0 will prefetch data past the cache line boundary. All subsequent PCI bus master reads will get incorrect data. Subsequent CPU cycles to PCI/LPC will get blocked behind the surplus data resulting in a system hang.

Implication: None known.

- System hang only seen in synthetic test environment.

- No known commercial PCI devices support cache-line wrap mode using Memory Read Multiple or Memory Read Line.

Workaround: None.

Status: No Fix.

70. MW DMA Mode-1 Tdh Erratum

Problem: Data hold time of MW DMA Mode-1 writes may not meet ATA Specification.

Implication: None known.

Workaround: Program the controller to PIO Mode-4 instead.

Status: NoFix.

Specification Changes

1. VBIAS Capacitor Value

As defined in Errata #38, the VBIAS capacitor (Figure 2-1, C1) is changed from 2200 pF to 0.047 µF.

2. Internal Pull-Down Resistor Values

Table 3-4 of Section 3.4 lists integrated pull-up and pull-down resistors. The value of all ***22 kΩ pull-down*** resistors is changed to approximately 2.7 kΩ. The other resistors remain unchanged.

3. USB Status Register Default

The USB Status Register, D31:F2;USB_BASE+02h, should indicate that the HCHalted bit (bit-5) is default to “1”, not “0”. The default for this register in Table 10-2 and in Section 10.2.2 is changed to 0020 instead of 0000.

4. AC’97 Wake from S5

Table 5-38 indicates that AC97 can wake from states S1-S4. AC97 can actually wake from S5 as well, if the S5 state was entered by the software setting the SLP_EN bit. It can not wake if S5 was entered due to power failure, or power button override. So Note 2 applies to AC97 wake event.

5. APIC Data Register Access DW Only

Section 8.5.3 indicates that the DAT register of the APIC Controller, can be accessed by byte. It must be accessed by DW.

6. INTRD_SEL Change

Section 8.9.9 describes the TCO2 Control Register (TCO2_CNT). This specification change modifies the INTRD_SEL field, specifically the value of 00b for INTRD_SEL. The 00b value for this field is listed as RESERVED, and is changed to “No SMI# or Interrupt generated”.

7. IRQPA and EOIR Register Removal

Section 8.5.8 and 8.5.9 of the datasheet describes the IRQPA and EOIR registers of the APIC (D31:F0). These registers are used for MSI (PCI Message Signalled Interrupts), which is not supported. These two registers are not supported and will be removed from the datasheet.

8. SERR_DTT_EN Change

Certain PCI cards which may cause the ICH to discard pre-fetched delayed transaction data can cause the ICH to generate an SERR and an NMI if SERR_DTT_EN is enabled. Since these cards are behaving legally, although a little strangely, and the data is prefetchable, this is not an error case. To prevent generation of SERR or NMI with these legally behaving cards, the SERR_DTT_EN bit (D30:F0;Offset 90h bit-1) should be cleared to 0 at all times. Section 7.1.28 will be modified accordingly to indicate that this bit should be cleared to 0 unless NMI handlers are capable of handling this condition.

9. AC’97 Register Access Requirements

Certain AC’97 IO registers require access using specific alignment. Section 12.2 (AC’97 Audio IO Space) and Section 13.2 (AC’97 Modem IO Space) identify these various IO registers.

12.2.1 x_BDBAR-Buffer Descriptor Base Address Register

(PIBDBAR, POBDBAR, MCBDBAR) must be accessed with 32-bit cycles only.

12.2.4 x_SR-Status Register

(PISR, POSR, MCSR) must be accessed with 16-bit cycles only.

12.2.5 x_PICB-Position in Current Buffer Register

(PIPICB, POPICB, MCPIPB) must all be accessed with 16-bit cycles only.

12.2.8 GLOB_CNT-Global Control Register must be accesses with 32-bit cycles only.

12.2.9 GLOB_STA-Global Status Register must be accessed with 32-bit cycles only.

13.2.1 x_BDBAR-Buffer Descriptor List Base Address Register

(MIBDBAR, MOBDBAR) must all be accessed with 32-bit cycles only.

13.2.4 x_SR-Status Register

(MISR, MOSR) must all be accessed with 16-bit cycles only.

13.2.5 x_PICB-Position In Current Buffer Register

(MIPICB, MOPICB) must all be accessed with 16-bit cycles only.

13.2.8 GLOB_CNT-Global Control Register must be accessed with 32-bit cycles only.

13.2.9 GLOB_STA-Global Status Register must be accessed with 32-bit cycles only.

10. I/O APIC Max Frequency Spec Change

Table 14-7 of the datasheet shows the Operating Frequency of the I/O APIC as 16.67 MHz. This is being changed to 33.33 MHz.

11. Function Disable Requirements

Section 6.1 of the datasheet describes the disabling of ICH functions. The following will be added immediately after the first sentence of the second paragraph of Section 6.1: “BIOS must not enable I/O or memory decode in the PCI CMD register of any function that is going to be disabled.”

12. Addition of Discard Timer Mode Bit

Section 7.1.25 describes the functions of various bits in the ICH Configuration Register D30:F0;CNF. Bit-2, listed as RESERVED is now defined as Discard Timer Mode bit.

CNF—ICH Configuration Register (HUB-PCI-D30:F0;50-51h)

Bit	Description
7:3	Reserved
2	**Discard Timer Mode.** This bit shortens all of the Delayed Transaction discard timers to 64 PCI clocks. It controls how long the ICH will wait before flushing previously requested prefetched read data due to a Delayed Transaction, and then servicing a different request. 0 = 1024 PCI clocks (32 us) (Default). 1 = 64 PCI clocks (2 us).

13. Addition of PWROK Deassertion Minimum Specification

A new specification is being added for the PWROK signal. When PWROK is deasserted, it must be deasserted for at least 100 µs. This will be added to the datasheet under Table 14-19 (Power Management Timing) as symbol t204.

14. DMA Mode-0 Not Supported

DMA Mode-0 is not supported by the ICH.

15. AC’97 Integrated Pull-Down Control

The integrated pull-down resistors on AC_BITCLK, AC_SDIN:[1:0], and AC_SDOUT can be enabled when either the ACLINK Shut Off bit in the AC’97 Global Control Register is set or when both function 5 and function 6 are hidden (disabled). Note 2 below Table 3-4 will be changed accordingly.

16. USB Clock PPM

Table 14-7 lists the ppm of the USB Clock as 2500 ppm (max). This is changed to 500 ppm. The following sentence is added to the note associated with this item, “The source of this ppm is external to the ICH component.”

17. I/O APIC Arbitration ID Not Set When APIC CLK Not Running

In Section 8.5.6, ARBID - Arbitration ID Register, the second sentence of the first paragraph is being changed to “If APIC Clock is running, this register is loaded whenever the APIC ID register is loaded.”

This page is intentionally left blank.

Specification Clarifications

1. Intruder_Detect Bit Behavior

The following will be added to Section 5.13.2, "*Handling an Intruder.*"

Note: If the INTRUDER# signal is active when software attempts to clear the INTRD_DET bit, the bit will remain as a 1, and the SMI will be generated again immediately. The SMI handler can clear the INTRD_SEL bits to avoid further SMIs. However, if the INTRUDER# signal goes inactive and then active again, there will not be further SMIs (because the INTRD_SEL bits would select no SMI# be generated).

Additionally, if the INTRUDER# signal goes inactive some point after the INTRD_DET bit is written as a 1, then the INTRD_DET signal will go to a 0 when INTRUDER# input signal goes inactive. Note that this is slightly different than a classic sticky bit.

2. AC_SDOUT Internal Pull-Down Strength

Table 2-18 in Section 2.18.1 indicates that the AC_SDOUT pin contains a weak internal pull-down resistor. The strength of this resistor is approximately 2.2 kΩ. To properly detect a SAFE_MODE condition, a strong pull-up resistor will be required on the motherboard to over-ride this internal pull-down.

3. AD[31:0] Value during PCI Special Cycles

As per the PCI Specification, during the address phase of a special cycle, there is no provision regarding the pattern that is driven on AD[31:0]. The specification does state that these lines be driven with stable values. The behavior of the ICH is such that a stable pattern will be driven during the address phase, but the particular pattern of bits could be anything. The ICH will drive this bus with whatever was the last value in the internal buffer. This means that it is possible that the pattern driven could be in the address range of a PCI device residing in the system. Therefore, it is absolutely required that PCI devices not violate the PCI specification by claiming special cycles just because the address during the address phase happens to be within its range.

4. FREQ_STRAP[3:0] Behavior

Section 8.1.22 describes the FREQ_STRAP[3:0] field of the General Status Register (LPC I/F D31:F0). This field is only writable when SAFE_MODE (of the same register) is cleared to 0. Also note that SAFE_MODE is only cleared by PWROK rising edge.

5. PCI Cycles to LPC or FWH Devices

The following Note will be added to Section 5.1.1.

Note: PCI cycles to LPC or FWH devices or internal ICH registers can result in deadlocks if the PCI device is itself a target of LPC-to-PCI or FWH-to-PCI cycles. While this is highly unusual behavior, there may be some PCI adapters that do access LPC or FWH resources, however these are used only where the board is being debugged/tested. These cycles should be prohibited in these environments. ICH cycles initiated by 8237 (or LPC) should only target main memory, with the only exception being that LPC masters can access devices on PCI in diagnostics and emergency cases. It should be noted that there is a risk of livelock if the PCI devices are still active.

6. INTRD_DET Bit Behavior

The INTRD_DET bit (bit-0) of TCO2_STS, is in the RTC well and clocked by the RTC clock. After software initiates the clear (by writing a 1 to that bit), there is a 2 RTC-clock delay before it is actually cleared internally. If software reads the bit before this delay, it will still read back as a “1”, rather than a 0. The recovery time is 65 µs before attempting to read back this bit.

7. NEWCENTURY_STS Behavior

Section 8.9.6, TCO1_STS, describes the NEWCENTURY_STS bit. The following is an additional clarification. This bit may take up to 3 RTC-clocks for the bit to be cleared when a 1 is written to clear it. After writing a 1 to the NEWCENTURY_STS bit, software should not exit the SMI handler until after the bit has been cleared. This will ensure the SMI is not re-entered.

8. Parity Error Response Enables Intel® ICH Parity Generation

Section 7.1.3, *Command Register (CMD), D30:F0;04-05h*, describes the Parity Error Response bit (bit 6). The datasheet indicates that when this bit is set to 1 that “the ICH is allowed to report parity errors on the hub interface.” When this bit is set, it also enables generation of parity on hub interface.

9. Delayed Transaction Discard Timer Clarification

The ICH has a feature which keeps prefetched data in a FIFO when it, as a target to main memory, causes a disconnect due to a temporary under-run and the master wanted to continue a burst. This feature can cause latency to other devices with poorly behaving PCI masters.

As a master attempts to burst across a cache line, the ICH will disconnect the master if additional prefetched data has not arrived from the memory subsystem yet. This gives other devices access to the bus. After the disconnect, the ICH will eventually receive the remainder of the prefetched data from memory. If the master had indicated that it wanted to continue the burst at the time of the disconnect (i.e., FRAME# asserted), then the ICH will keep the prefetched data in its delayed transaction buffer. Therefore, if the master comes back after the disconnect to access a location other than the subsequent location prefetched by the ICH, then the ICH relies on an internal “discard timer” to prevent system deadlocks. This discard timer waits 32 µs before it flushes the FIFO and re-fetches the non-subsequent location. This overall system behavior may then starve out other accesses. PCI masters behaving in this way are considered poorly behaving masters, although they do behave within PCI specifications.

10. NO_REBOOT Strap Clarification

Section 8.1.22 describes the GEN_STA-General Status register which includes the NO_REBOOT bit (bit-1). This NO_REBOOT cannot be cleared while an external jumper is in place on the SPKR signal.

11. HST_STA Register INTR Bit Clarification

The datasheet currently indicates that the INTR (bit-1) of the SMBus Controller I/O HST_STA register (offset 0h) has the following functionality, “This bit can only be set by termination of a command and the INTREN bit of the Host Controller Register (offset 02h) is set.” The INTR bit is set by the successful completion of a command regardless of the setting of the INTREN bit.

12. SMBus BYTE_DONE_STS Clarification

The datasheet indicates that the BYTE_DONE_STS (bit-7) of the SMBus Controller I/O HST_STA register is set by the ICH after a byte has been transferred or sent. The ICH actually will set the BYTE_DONE_STS as soon as the data has been loaded into an internal transmit buffer, which actually occurs just before the byte is sent on the SMBus. This will typically result in software detecting this bit as set, and then software clearing it, probably before the byte is actually sent over the SMBus. This software/ICH interaction or behavior will continue to result in normal SMBus operation.

13. PCI Underrun Behavior

If the processor issues a locked cycle to a resource that is too slow (e.g., PCI), the ICH will not allow upstream requests to be performed until the cycle completion. This may be critical for isochronous buses which assume certain timing for their data flow, such as 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 ICH is not able to issue a request for the next data. Snoop cycles are not permitted while the front side bus is locked.

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

This will result in the following "note" being added to Section 5.1.1, "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).

14. Virtual Wire Mode B Usage on Intel® ICH

When an IOAPIC based system is configured in Virtual Wire Mode B with edge triggered interrupt delivery on the IOAPIC input pin INITIN_0 (INTR output from the 8259), a high priority interrupt occurring just as the ICH receives an INTACK for a preceeding low-priority interrupt can cause an unusually small interrupt de-assertion on INTR signal which can be missed at either the IOAPIC or the processor, depending on the configuration, and most likely cause a system hang.

The unusually small interrupt de-assertion time does not meet the input min specifications for the device receiving this signal. This can result in system lockup.

If Virtual Wire Mode B is used, where INTR output of the 8259 is routed to IOAPIC INTIN_0, software must program the IOAPIC redirection table entry 0, Trigger Mode (bit-15 of redirection table) to Level Sensitive (rather than edge sensitive). This may result in spurious interrupts which should have minimal adverse impact on performance.

Virtual Wire Mode A can also be used instead with the local APICs set to Level Trigger Mode.

15. PCI Latency Clarification

Section 5.1.1 describes the PCI Bus Interface. This section does not describe standard PCI bus behavior in any detail. The following note will be added to this section.

The ICH was designed to provide high performance support to PCI peripherals using its data prefetch capabilities. If a PCI master is burst reading and is disconnected by the ICH to pre-fetch the requested cache line, the ICH will Delay Transaction the cycle while it prefetches more data, and give the bus to another agent. Once the bus is given back to this bus master, if it does not return with the successive previously requested read address, which was prefetched by the ICH, the ICH will keep retrying the bus master until either it comes back for the prefetched data, or the Delayed Transaction Discard Timer expires (1024 PCI clocks) before discarding this prefetched data and servicing the request. This induces long latencies to PCI bus masters that behave this way. To reduce this latency, the Discard Timer Mode bit (D30:F0;CNF(50-51h):[bit-2]) can be set to 1. This will reduce the discard timer from 1024 PCI clocks (32 µs) to 64 clocks (2 µs) and improve latency for masters with this behavior.

16. GPE Event Handling under ACPI OS

ICH uses the same GPE1_EN register (I/O address: PMBase+2Eh) to enable/disable both SMI and ACPI SCI general-purpose input events. ACPI OS assumes that it owns the entire GPE1_EN register per ACPI spec. Problems arise when some of the general-purpose inputs are enabled as SMI by BIOS, and some of the general-purpose inputs are enabled for SCI. In this case, ACPI OS turns off the enable bit for any GPIx input signal that are not indicated as SCI general-purpose events at boot, and at entry and exit from sleeping states.

Since SMI events from general-purpose inputs are blocked after OS intialization, or after entry into a system sleeping state in ACPI OS environments (both Microsoft* Windows* 98 and Windows* 2000), system BIOS should define a dummy control method which prevents the ACPI OS from clearing the SMI GPE1_EN bits.

17. USB Overcurrent Detection in Sleep States

The ICH/ICH0 will not wake from a USB overcurrent condition when the system is in S3, S4, or S5 sleep states. In these sleep states neither of the overcurrent status bits in the PORTSC:[1:0] register will set. If an overcurrent still exists when the system is in S0 or S1 - then the overcurrent can be detected. This will be added as a note to the PORTSC:[1:0] register.

18. Running Block Read with SECOND_TO_STS Set

In section 11.2.2, HST_CNT - Host Control Register, add the following note to the description for LAST_BYTE (bit-5).

Note: Once the SECOND_TO_STS bit in the TCO2_STS register (Dev31, Func 0; TCOBase+06h:[1]) is set, the LAST_BYTE bit also gets set. While the SECOND_TO_STS bit is set, the LAST_BYTE bit cannot be cleared. This prevents the ICH from running some of the SM Bus commands (Block Read/Write, I2C Read, Block I2C Write).

19. DMA Clarification

Section 8.2.1 (DMABASE_CA) is being replaced with the following:

Bit	Description
15:0	**Base and Current Address - R/W.:** This register determines the address for the transfers to be performed. The address specified points to two seperate registers. On writes, the value stored in the *Base Address* register and copied to the *Current Address* register. On reads, the value is returned from the *Current Address* register. 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. For transfers to/from a 16-bit slave (channels 5-7), the address is shifted left one bit location. Bit 15 will be shifted into Bit 16. The register is accessed in 8 bit quantities. The byte is pointed to by the current byte pointer flop/flop. Before accessing an address register, the byte pointer flip/flop should be cleared to ensure that the low byte is accessed first.

Section 8.2.3 (DMAMEM_LP) is being replaced with the following:

Bit	Description
7:0	**DMA Low Page (ISA Address bits [23:16]) - R/W.:** This register works in conjunction with the DMA controller's Current Base Address register to define the complete 24-bit address for the DMA channel. This register remains static throughout the DMA transfer. Bit 16 of this register is ignored when in 16 bit I/O count by words mode as it is replaced by the bit 15 shifted out from the current address register.

Section 5.3.2 (Address Compatibility Mode) is being replaced with the following:

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

20. USB Run/Stop Bit Clarification

The following note is added to the USBCMD - USB Command Register, bit-0, in Section 10.2.1:

Note: The USB run/stop bit should be cleared only under one of the following conditions:

1. There are no active Transaction Descriptors in the schedule.

2. A reset of the USB host controller is guaranteed prior to a subsequent run/stop bit assertion.

21. RTC SET Bit Clarification

The RTC SET bit (bit-7) in RTC_REGB should be set then cleared early in BIOS POST after each powerup directly after coin-cell battery insertion.

22. End Of SMI Bit

In section 8.8.3.11 (SMI_EN) the following note is added to bit-1:

Note: ICH/ICH0 is able to generate the first SMI after reset even though the EOS bit is not set. Subsequent SMI's require the EOS bit be set.

23. SMBus Wake

In section 8.8.3.7 (GPE0_STS) the following is added to bit-7 (SMB_WAK_STS):

Note: The SMBALERT_STS bit (D31:F3;Offset 00h:bit-5) should be cleared by software before this bit is cleared.

24. 32 Clock Retry Enable Clarification

The description for the 32-Clock Retry Enable bit (bit-1) in the Device 30 Function 0:CNF(50h) register is incorrect. There is no relationship to PCI locked cycles. The description is changed accordingly as indicated:

32 Clock Retry Enable - R/W, System BIOS must set this bit for PCI compliance.

1=The ICH/ICH0 will retry a PCI to memory cycle (reads or writes) if the ICH/ICH0 is not able to complete the transfer in 32 PCI clocks.

0=The ICH/ICH0 will insert as many wait states as needed to complete the PCI to memory cycle.

25. LPC LPCPD# Protocol Clarification

The LPC specification defines the LPCPD# protocol where there is at least 30uS from LPCPD# assertion to assertion of LRST#. This specification explicitly states that this protocol only applies to entry/exit of low power states which does not include asynchronous reset events. The ICH/ICH0 will assert both SUS_STAT# (connects to LPCPD#) and PCIRST# (connects to LRST#) at the same time when the core logic is reset (via CF9, PWROK, etc). This is not inconsistent with the LPC LPCPD protocol.

26. PCI Master Clarification

Add Note to section 5.1.1 (PCI Bus Interface). Note: PCI bus masters should not use memory area locations as a target if that area is programmed to be anything but Read/Write.

Documentation Changes

1. PCI Device Revision ID Table

PCI Revision ID Register Values (PCI Offset 08h) for all ICH/ICH0 functions are shown below.

This is not listed in the datasheet. This is the standard reference document.

Table 1. Revision ID Table

Function	ICH/ ICH0 A-0 Step	ICH/ ICH0 A-1 Step	ICH/ ICH0 B-0 Step	ICH/ ICH0 B-1 Step
D30:F0	0	0	1	2
D31:F0	0	0	1	2
D31:F1	0	0	1	2
D31:F2	0	0	1	2
D31:F3	0	0	1	2
D31:F5	0	0	1	2
D31:F6	0	0	1	2

2. PIO4 Maximum Performance

The IDE Interface section of the Overview (Section 1.2) indicates that the IDE interface supports PIO IDE transfers up to 14 Mbytes/sec. This is actually 16.6 Mbytes/sec which is achieved with ISP=3, RCT=1 with PCI at 33 MHz.

3. THRM# Signal Correction

Section 5.12.6.1 of the datasheet, the last paragraph, states that another SMI or SCI can optionally be generated when the THRM# signal goes back low. It should state, "... when the THRM# signal goes back high."

4. SCI Level Correction

Section 5.12.3 describes SCI generation. The third paragraph indicates that the SCI interrupt generated internally is active low level. This is actually active HIGH level.

5. Ultra ATA33 Operation Correction

The third paragraph of Section 5.14.3.2 is incorrect. The second sentence "When DMACK# is asserted, the host controller drives CS0# and CS1# inactive, DA0-DA2 low and the IDE device drives IOCS16# inactive." This is changed to the following; "When DMACK# is asserted, the host controller drives CS0# and CS1# inactive.

6. Figure Correction

Figures 14-7 and 14-8 (IDE Timing Figures) incorrectly refer to PCICLK at the top. The label for this signal should read CLK66.

7. PBUS_NUM Is Read Only

Section 7.1.10 indicates that D30:F0 PBUS_NUM - Primary Bus Number Register has R/W attribute. It is actually Read Only since this field is hardwired.

8. TCO_INT_SEL Description Correction

Section 8.1.13 of the datasheet describes the TCO_INT_SEL bit. The text incorrectly states that the TCOSCI_EN bit is located in bit-6 of the same register (TCO_CNTL), which is incorrect. The TCOSCI_EN bit is located at bit-6 of the GPE0_EN register, which is described in Section 8.8.3.8 of the datasheet. Additionally, the active level of the internal TCO Interrupt request is active high for any IRQ[9:11, 20:23].

9. RTC Description Correction

The description of the Real Time Clock (RTC), Section5.10, states "The SET bit of register B *should* be one while programming these locations (time and calendar CMOS locations) to avoid clashes with an update cycle. The SET bit *must* be one while programming these locations.

10. PROC_CNT Bit Correction

The datasheet, Section 8.8.3.5 describes the various bits of the PROC_CNT Register (D31:F0;PMBase+10h). Bit-4 is incorrectly listed as THTT_EN, and should be listed as THTL_EN. The description for this bit remains correct.

11. Correction to Table 6-3 (Fixed IO Ranges Decoded by Intel® ICH)

The notes 1 and 2 associated with table 6-3 are incorrectly applied. Note 1 should apply to IO Address 1F0–1F7h and 3F6h (primary IDE controller), and Note 2 should apply to IO Address 170–177h and 376h (secondary IDE controller).

12. RTCRST# Jumper Clarification

Section 5.10.5 describes useage of the RTCRST# jumper. The following will be added to the paragraph "Using RTCRST# to clear CMOS"

RTCRST# should be used to reset configuration bits (and signal BIOS to clear CMOS) **only** in a G3 state. RTCRST# assertion while power is on must **only** be done to invoke the test modes, and that it should only be asserted for the specific number of clocks to invoke the desired mode. Assertion for any other number of clocks may put the component into an indeterminate state, and is not supported.

13. Internal Pull-Up Resistor Value

The "Internal Pull-Up Resistor Values" specification change is being renamed to "Internal Pull-Down Resistor Values."

14. IDE PRD Alignment

Section 5.14.2.1, *Physical Region Descriptor Format*, second paragraph, describes requirements for the Physical Region Descriptor (PRD) Format. This section incorrectly indicates that the PRD must be aligned to 64-KB boundaries. This paragraph will be changed to indicate that the PRD must be DWord aligned and must not cross a 64-KB boundary.

15. SUSCLK During RSMRST# Assertion

Table 3-2 incorrectly indicates that SUSCLK is "Running" during PCIRST#. SUSCLK is low when RSMRST# is asserted, but will be "Running" when PCIRST# is deasserted.

16. TCO Corrections

Section 8.9.2 incorrectly lists TCO1_RLD as the register name for TCOBase+00h. This register is correctly named TCO_RLD - "TCO Timer Reload and Current Value". This is an 8-bit register, thus the correct default value of this register is 00h.

Section 8.9.3 incorrectly lists TCO1_TMR as the register name for TCOBase+01h. This register is correctly named TCO_TMR - "TCO Timer Initial Value". This is an 8-bit register, thus the correct default value of this register is 04h. The description of bits 5:0 incorrectly indicates that values of 0-3h are ignored and should not be attempted". Only values of 0-1 are actually ignored.

Section 8.9.4 incorrectly lists TCO1_DAT_IN as the register name for TCOBase+02h. This register is correctly named TCO_DAT_IN - "TCO Data In Register". This is an 8-bit register, thus the correct default value of this register is 00h. The last sentence of the description of this register is changed to: "Writes to this register cause an SMI and sets the SW_TCO_SMI bit in the TCO1_STS register.".

Section 8.9.5 incorrectly lists TCO1_DAT_OUT as the register name for TCOBase+03h. This register is correctly named TCO_DAT_OUT - "TCO Data Out Register". This is an 8-bit register, thus the correct default value of this register is 00h.

17. APM I/O Decode Correction

The second sentence of Section 8.8.2 is changed to read "This register space cannot be moved (fixed I/O location)."

18. Memory Map Table Change

The last Row of Table 6-5 in Section 6.3 (Memory Map) is changed as indicated:

Memory Range	Target
All other	Any memory range access that makes it to the ICH's PCI bus and is not specified in one of the D31:F0;0xE0-0xEF registers or below 16M will be master aborted.

19. SMBus Host_Busy Correction

Section 11.2.1 HST_STS - Host Status Register is being corrected. Specifically, bit-0 Host_Busy is corrected to read as follows:

0 = Cleared by the ICH when the current transaction is completed.

1 = Indicates that the ICH is running a command from the host interface. No SMB registers should be accessed while this bit is set, except the BLOCK DATA BYTE or LAST BYTE registers. The BLOCK DATA BYTE and LAST BYTE registers can be accessed when this bit is set only when the SMB_CMD bit in the Host Control register is programmed for Block command or I2C Read command. This is necessary in order to check the DONE_STS bit.