Intel® 82801E Communications I/O Controller Hub (C-ICH)

Developer's Manual

January 2002

Order Number: 273599-001

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel's Terms and Conditions of Sale for such products, Intel assumes no liability whatsoever, and Intel disclaims any express or implied warranty, relating to sale and/or use of Intel products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel products are not intended for use in medical, life saving, or life sustaining applications.

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

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

The Intel® 82801E Communications I/O Controller Hub (C-ICH) 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.

MPEG is an international standard for video compression/decompression promoted by ISO. Implementations of MPEG CODECs, or MPEG enabled platforms may require licenses from various entities, including Intel Corporation.

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

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature may be obtained by calling 1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

Copyright © Intel Corporation, 2002

AlertVIEW, i960, AnyPoint, AppChoice, BoardWatch, BunnyPeople, CablePort, Celeron, Chips, Commerce Cart, CT Connect, CT Media, Dialogic, DM3, EtherExpress, ETOX, FlashFile, GatherRound, i386, i486, iCat, iCOMP, Insight960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel ChatPad, Intel Create&Share, Intel Dot.Station, Intel GigaBlade, Intel InBusiness, Intel Inside, Intel Inside logo, Intel NetBurst, Intel NetStructure, Intel Play, Intel Play logo, Intel Pocket Concert, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel TeamStation, Intel WebOutfitter, Intel Xeon, Intel XScale, Itanium, JobAnalyst, LANDesk, LanRover, MCS, MMX, MMX logo, NetPort, NetportExpress, Optimizer logo, OverDrive, Paragon, PC Dads, PC Parents, Pentium, Pentium II Xeon, Pentium III Xeon, Performance at Your Command, ProShare, RemoteExpress, Screamline, Shiva, SmartDie, Solutions960, Sound Mark, StorageExpress, The Computer Inside, The Journey Inside, This Way In, TokenExpress, Trillium, Vivonic, and VTune are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

Contents

1 Guide to this Manual 21

1.1 Manual Contents 21
1.2 Text Conventions 23
1.3 Technical Support 24
1.3.1 Electronic Support Systems 24
1.3.2 Telephone Technical Support 24
1.4 Product Literature 24
1.4.1 Related Documents 25

2 Introduction 27

2.1 Feature Summary 27
2.2 Overview 29
2.3 Capability Summary 31

3 Functional Description 35

3.1 Hub Interface to PCI Bridge (D30:F0) 35
3.1.1 PCI Bus Interface 35
3.1.2 PCI-to-PCI Bridge Model 36
3.1.3 IDSEL to Device Number Mapping 36
3.1.4 SERR# Functionality 36
3.1.5 Parity Error Detection 38
3.1.6 Standard PCI Bus Configuration Mechanism 39
3.1.7 PCI Dual Address Cycle (DAC) Support 40
3.2 LAN Controllers (B1:D8/D9:F0) 40
3.2.1 LAN Controller Architectural Overview 41
3.2.1.1 Parallel Subsystem Overview 42
3.2.1.2 FIFO Subsystem Overview 42
3.2.1.3 Serial CSMA/CD Unit Overview 43
3.2.2 LAN Controller PCI Bus Interface 43
3.2.2.1 Bus Slave Operation 43
3.2.2.2 Bus Master Operation 45
3.2.2.3 PCI Reset Signal 48
3.2.2.4 Wake-up Events 48
3.2.3 Serial EEPROM Interface 48
3.2.4 CSMA/CD Unit 50
3.2.5 Media Management Interface 51
3.2.6 TCO Functionality 51
3.3 LPC Bridge (with System and Management Functions) (D31:F0) 52
3.3.1 LPC Interface 52
3.3.1.1 LPC Cycle Types 52
3.3.1.2 Start Field Definition 53
3.3.1.3 Cycle Type/Direction (CYCTYPE + DIR) 53
3.3.1.4 Size 54
3.3.1.5 SYNC 54
3.3.1.6 SYNC Time-out 55
3.3.1.7 SYNC Error Indication 55
3.3.1.8 LFRAME# Usage 56

3.3.1.9 I/O Cycles 57
3.3.1.10 Bus Master Cycles 57
3.3.1.11 Configuration and 82801E C-ICH Implications 57
3.3.1.12 BIOS Mapping and START Fields 57
3.4 DMA Operation (D31:F0) 58
3.4.1 Channel Priority 58
3.4.2 Address Compatibility Mode 59
3.4.3 Summary of DMA Transfer Sizes 59
3.4.4 Autoinitialize 60
3.4.5 Software Commands 61
3.5 PCI DMA 62
3.5.1 PCI DMA Expansion Protocol 62
3.5.2 PCI DMA Expansion Cycles 64
3.5.3 DMA Addresses 64
3.5.4 DMA Data Generation 64
3.5.5 DMA Byte Enable Generation 65
3.5.6 DMA Cycle Termination 65
3.5.7 LPC DMA 65
3.5.8 Asserting DMA Requests 65
3.5.9 Abandoning DMA Requests 66
3.5.10 General Flow of DMA Transfers 66
3.5.11 Terminal Count 67
3.5.12 Verify Mode 67
3.5.13 DMA Request Deassertion 67
3.5.14 SYNC Field/LDRQ# Rules 68
3.6 8254 Timers (D31:F0) 69
3.6.1 Timer Programming 69
3.6.2 Reading from the Interval Timer 70
3.7 8259 Programmable Interrupt Controllers (PIC) (D31:F0) 72
3.7.1 Interrupt Handling 73
3.7.1.1 Generating Interrupts 73
3.7.1.2 Acknowledging Interrupts 73
3.7.1.3 Hardware/Software Interrupt Sequence 73
3.7.2 Initialization Command Words (ICWx) 74
3.7.3 Operation Command Words (OCW) 75
3.7.4 Modes of Operation 75
3.7.5 Masking Interrupts 77
3.7.6 Steering PCI Interrupts 78
3.7.7 Special Handling of IRQ1 and IRQ12 78
3.8 Advanced Interrupt Controller (APIC) (D31:F0) 79
3.8.1 Interrupt Handling 79
3.8.2 Interrupt Mapping 79
3.8.3 APIC Bus Functional Description 80
3.8.3.1 Physical Characteristics of APIC 80
3.8.3.2 APIC Bus Arbitration 80
3.8.3.3 Bus Message Formats 81
3.8.4 PCI Message-Based Interrupts 86
3.8.4.1 Theory of Operation 86
3.8.4.2 Registers and Bits Associated with PCI Interrupt Delivery 86

3.8.5 Processor-Side Bus Interrupt Delivery 87
3.8.5.1 Theory of Operation 87
3.8.5.2 Edge-Triggered Operation 87
3.8.5.3 Level-Triggered Operation 88
3.8.5.4 Registers Associated with Processor-Side Bus Interrupt Delivery 88
3.8.5.5 Interrupt Message Format 88
3.9 Serial Interrupt (D31:F0) 89
3.9.1 Start Frame 90
3.9.2 Data Frames 90
3.9.3 Stop Frame 90
3.9.4 Specific Interrupts Not Supported via SERIRQ 91
3.9.5 Data Frame Format 91
3.10 Real Time Clock (D31:F0) 92
3.10.1 Update Cycles 92
3.10.2 Interrupts 93
3.10.3 Lockable RAM Ranges 93
3.10.4 Century Rollover 93
3.10.5 Clearing Battery-Backed RTC RAM 93
3.11 Processor Interface (D31:F0) 95
3.11.1 Processor Interface Signals 95
3.11.1.1 A20M# 95
3.11.1.2 INIT# 95
3.11.1.3 FERR#/IGNNE# (Coprocessor Error) 96
3.11.1.4 NMI 96
3.11.1.5 STPCLK# and CPUSLP# Signals 97
3.11.1.6 CPUPWRGOOD Signal 97
3.11.2 Speed Strapping for Processor 97
3.12 Power Management (D31:F0) 98
3.12.1 82801E C-ICH and System Power States 98
3.12.2 82801E C-ICH Power Planes 99
3.12.3 SMI#/SCI Generation 99
3.12.4 Dynamic Processor Clock Control 101
3.12.4.1 Throttling Using STPCLK# 102
3.12.5 STPCLK# Implementation Notes 102
3.12.5.1 Transition Rules Among S0/Cx and Throttling States 103
3.12.6 Thermal Management 103
3.12.6.1 THRM# Signal for SMI# or SCI 103
3.12.6.2 THRM# Initiated Passive Cooling 103
3.12.6.3 THRM# Override Software Bit 104
3.12.6.4 Processor-Initiated Passive Cooling (Via Programmed Duty Cycle on STPCLK#) 104
3.12.6.5 Active Cooling 104
3.12.7 Event Input Signals and Their Usage 104
3.12.7.1 RI# — Ring Indicate 104
3.12.8 Alt Access Mode 104
3.12.8.1 Write Only Registers with Read Paths in Alternate Access Mode 105
3.12.8.2 PIC Reserved Bits 107
3.12.8.3 Read Only Registers with Write Paths in Alternate Access Mode 107
3.12.9 Clock Generators 108
3.12.10 Legacy Power Management Theory of Operation 108
3.12.10.1 Advanced Power Management (APM) 109

3.13 System Management (D31:F0) 109
3.13.1 Theory of Operation 109
3.13.2 Alert on LAN* 110
3.13.3 IRQ1 and IRQ12 for Legacy Elimination 112
3.14 General Purpose I/O 113
3.15 IDE Controller (D31:F1) 113
3.15.1 PIO Transfers 114
3.15.2 Bus Master Function 115
3.15.3 Ultra ATA/33 Protocol 119
3.15.4 Ultra ATA/66 Protocol 121
3.15.5 Ultra ATA/100 Protocol 121
3.15.6 Ultra ATA/33/66/100 Timing 121
3.16 USB Controller (Device 31:Function 2) 122
3.16.1 Data Structures in Main Memory 122
3.16.1.1 Frame List Pointer 122
3.16.1.2 Transfer Descriptor (TD) 123
3.16.1.3 Queue Head (QH) 127
3.16.2 Data Transfers To/From Main Memory 128
3.16.2.1 Executing the Schedule 128
3.16.2.2 Processing Transfer Descriptors 129
3.16.2.3 Command Register, Status Register, and TD Status Bit Interaction 130
3.16.2.4 Transfer Queuing 130
3.16.3 Data Encoding and Bit Stuffing 134
3.16.4 Bus Protocol 134
3.16.4.1 Bit Ordering 134
3.16.4.2 SYNC Field 134
3.16.4.3 Packet Field Formats 135
3.16.4.4 Address Fields 136
3.16.4.5 Frame Number Field 136
3.16.4.6 Data Field 136
3.16.4.7 Cyclic Redundancy Check (CRC) 137
3.16.5 Packet Formats 137
3.16.5.1 Token Packets 137
3.16.5.2 Start of Frame Packets 137
3.16.5.3 Data Packets 138
3.16.5.4 Handshake Packets 138
3.16.5.5 Handshake Responses 139
3.16.6 USB Interrupts 139
3.16.6.1 Transaction Based Interrupts 139
3.16.6.2 Non-Transaction Based Interrupts 141
3.16.7 USB Power Management 141
3.16.8 USB Legacy Keyboard Operation 142
3.17 SMBus Controller Functional Description (D31:F3) 144
3.17.1 Host Controller 145
3.17.1.1 Command Protocols 145
3.17.1.2 I^2C Read 150
3.17.1.3 I^2C Behavior 151
3.17.1.4 Heartbeat for Use With the External LAN Controller 151
3.17.2 Bus Arbitration 151
3.17.3 Interrupts/SMI# 152
3.17.4 SMBALERT# 152
3.17.5 SMBus Slave Interface 152

3.18 Firmware Hub Interface 156
3.18.1 Field Definitions 156
3.18.1.1 START 156
3.18.1.2 IDSEL (Device Select) 156
3.18.1.3 MSIZE (Memory Size) 156
3.18.1.4 MADDR (Memory Address) 156
3.18.1.5 SYNC 156
3.18.1.6 TAR 157
3.18.2 Protocol 157
3.18.2.1 Preamble 157

4 System Clock Domains 159

5 Register and Memory Mapping 161

5.1 PCI Devices and Functions 161
5.2 PCI Configuration Map 162
5.3 I/O Map 162
5.3.1 Fixed I/O Address Ranges 163
5.3.2 Variable I/O Decode Ranges 165
5.4 Memory Map 166
5.4.1 Boot-Block Update Scheme 167

6 LAN Controller Registers (B1:D8/D9:F0) 169

6.1 PCI Configuration Registers (B1:D8/D9:F0) 169
6.1.1 VID—Vendor ID Register (LAN Controller—B1:D8/D9:F0) 170
6.1.2 DID—Device ID Register (LAN Controller—B1:D8/D9:F0) 171
6.1.3 PCICMD—PCI Command Register (LAN Controller—B1:D8/D9:F0) 171
6.1.4 PCISTS—PCI Status Register (LAN Controller—B1:D8/D9:F0) 172
6.1.5 REVID—Revision ID Register (LAN Controller—B1:D8/D9:F0) 173
6.1.6 SCC—Sub-Class Code Register (LAN Controller—B1:D8/D9:F0) 173
6.1.7 BCC—Base-Class Code Register (LAN Controller—B1:D8/D9:F0) 173
6.1.8 CLS—Cache Line Size Register (LAN Controller—B1:D8/D9:F0) 173
6.1.9 PMLT—PCI Master Latency Timer Register (LAN Controller—B1:D8/D9:F0) 174
6.1.10 HEADTYP—Header Type Register (LAN Controller—B1:D8/D9:F0) 174
6.1.11 CSR_MEM_BASE CSR—Memory-Mapped Base Address Register (LAN Controller—B1:D8/D9:F0) 174
6.1.12 CSR_IO_BASE—CSR I/O-Mapped Base Address Register (LAN Controller—B1:D8/D9:F0) 175
6.1.13 SVID—Subsystem Vendor ID (LAN Controller—B1:D8/D9:F0) 175
6.1.14 SID—Subsystem ID (LAN Controller—B1:D8/D9:F0) 175
6.1.15 CAP_PTR—Capabilities Pointer (LAN Controller—B1:D8/D9:F0) 176

6.1.16 INT_LN—Interrupt Line Register (LAN Controller—B1:D8/D9:F0) 176
6.1.17 INT_PN—Interrupt Pin Register (LAN Controller—B1:D8/D9:F0) 176
6.1.18 MIN_GNT—Minimum Grant Register (LAN Controller—B1:D8/D9:F0) 177
6.1.19 MAX_LAT—Maximum Latency Register (LAN Controller—B1:D8/D9:F0) 177
6.1.20 CAP_ID—Capability ID Register (LAN Controller—B1:D8/D9:F0) 177
6.1.21 NXT_PTR—Next Item Pointer (LAN Controller—B1:D8/D9:F0) 177
6.1.22 PM_CAP—Power Management Capabilities (LAN Controller—B1:D8/D9:F0) 178
6.1.23 PMCSR—Power Management Control/Status Register (LAN Controller—B1:D8/D9:F0) 178
6.1.24 DATA—Data Register (LAN Controller—B1:D8/D9:F0) 178
6.2 LAN Control/Status Registers (CSR) 179
6.2.1 System Control Block Status Word Register 180
6.2.2 System Control Block Command Word Register 181
6.2.3 System Control Block General Pointer Register 183
6.2.4 PORT Register 183
6.2.5 EEPROM Control Register 184
6.2.6 Management Data Interface (MDI) Control Register 185
6.2.7 Receive DMA Byte Count Register 185
6.2.8 Early Receive Interrupt Register 186
6.2.9 Flow Control Register 186
6.2.10 Power Management Driver (PMDR) Register 187
6.2.11 General Control Register 188
6.2.12 General Status Register 188
6.2.13 Statistical Counters 188

7 Hub Interface to PCI Bridge Registers (D30:F0) 191

7.1 PCI Configuration Registers (D30:F0) 191
7.1.1 VID—Vendor ID Register (HUB-PCI—D30:F0) 192
7.1.2 DID—Device ID Register (HUB-PCI—D30:F0) 192
7.1.3 CMD—Command Register (HUB-PCI—D30:F0) 193
7.1.4 PD_STS—Primary Device Status Register (HUB-PCI—D30:F0) 194
7.1.5 REVID—Revision ID Register (HUB-PCI—D30:F0) 195
7.1.6 SCC—Sub-Class Code Register (HUB-PCI—D30:F0) 195
7.1.7 BCC—Base-Class Code Register (HUB-PCI—D30:F0) 195
7.1.8 PMLT—Primary Master Latency Timer Register (HUB-PCI—D30:F0) 195
7.1.9 HEADTYP—Header Type Register (HUB-PCI—D30:F0) 196
7.1.10 PBUS_NUM—Primary Bus Number Register (HUB-PCI—D30:F0) 196
7.1.11 SBUS_NUM—Secondary Bus Number Register (HUB-PCI—D30:F0) 196
7.1.12 SUB_BUS_NUM—Subordinate Bus Number Register (HUB-PCI—D30:F0) 196

7.1.13 SMLT—Secondary Master Latency Timer Register (HUB-PCI—D30:F0) 197
7.1.14 IOBASE—I/O Base Register (HUB-PCI—D30:F0) 197
7.1.15 IOLIM—I/O Limit Register (HUB-PCI—D30:F0) 197
7.1.16 SECSTS—Secondary Status Register (HUB-PCI—D30:F0) 198
7.1.17 MEMBASE—Memory Base Register (HUB-PCI—D30:F0) 199
7.1.18 MEMLIM—Memory Limit Register (HUB-PCI—D30:F0) 199
7.1.19 PREF_MEM_BASE—Prefetchable Memory Base Register (HUB-PCI—D30:F0) 200
7.1.20 PREF_MEM_MLT—Prefetchable Memory Limit Register (HUB-PCI—D30:F0) 200
7.1.21 IOBASE_HI—I/O Base Upper 16 Bits Register (HUB-PCI—D30:F0) 200
7.1.22 IOLIM_HI—I/O Limit Upper 16 Bits Register (HUB-PCI—D30:F0) 200
7.1.23 INT_LINE—Interrupt Line Register (HUB-PCI—D30:F0) 201
7.1.24 BRIDGE_CNT—Bridge Control Register (HUB-PCI—D30:F0) 201
7.1.25 BRIDGE_CNT2—Bridge Control Register 2 (HUB-PCI—D30:F0) 202
7.1.26 CNF—82801E C-ICH Configuration Register (HUB-PCI—D30:F0) 202
7.1.27 MTT—Multi-Transaction Timer Register (HUB-PCI—D30:F0) 203
7.1.28 PCI_MAST_STS—PCI Master Status Register (HUB-PCI—D30:F0) 203
7.1.29 ERR_CMD—Error Command Register (HUB-PCI—D30:F0) 204
7.1.30 ERR_STS—Error Status Register (HUB-PCI—D30:F0) 204

8 LPC Interface Bridge Registers (D31:F0) 205

8.1 PCI Configuration Registers (D31:F0) 205
8.1.1 VID—Vendor ID Register (LPC I/F—D31:F0) 206
8.1.2 DID—Device ID Register (LPC I/F—D31:F0) 206
8.1.3 PCICMD—PCI COMMAND Register (LPC I/F—D31:F0) 207
8.1.4 PCISTS—PCI Device Status (LPC I/F—D31:F0) 208
8.1.5 REVID—Revision ID Register (LPC I/F—D31:F0) 209
8.1.6 PI—Programming Interface (LPC I/F—D31:F0) 209
8.1.7 SCC—Sub-Class Code Register (LPC I/F—D31:F0) 209
8.1.8 BCC—Base-Class Code Register (LPC I/F—D31:F0) 209
8.1.9 HEADTYP—Header Type Register (LPC I/F—D31:F0) 209
8.1.10 PMBASE—ACPI Base Address (LPC I/F—D31:F0) 210
8.1.11 ACPI_CNTL—ACPI Control (LPC I/F—D31:F0) 210
8.1.12 BIOS_CNTL (LPC I/F—D31:F0) 211
8.1.13 TCO_CNTL—TCO Control (LPC I/F—D31:F0) 211
8.1.14 GPIOBASE—GPIO Base Address (LPC I/F—D31:F0) 212
8.1.15 GPIO_CNTL—GPIO Control (LPC I/F—D31:F0) 212
8.1.16 PIRQ[n]_ROUT—PIRQ[A,B,C,D] Routing Control (LPC I/F—D31:F0) 212
8.1.17 SERIRQ_CNTL—Serial IRQ Control (LPC I/F—D31:F0) 213
8.1.18 PIRQ[n]_ROUT—PIRQ[E,F,G,H] Routing Control (LPC I/F—D31:F0) 213
8.1.19 D31_ERR_CFG—Device 31 Error Configuration Register (LPC I/F—D31:F0) 214

8.1.20 D31_ERR_STS—Device 31 Error Status Register (LPC I/F—D31:F0) 214
8.1.21 PCI_DMA_CFG—PCI DMA Configuration (LPC I/F—D31:F0) 215
8.1.22 GEN_CNTL—General Control Register (LPC I/F—D31:F0) 215
8.1.23 GEN_STS—General Status (LPC I/F—D31:F0) 217
8.1.24 RTC_CONF—RTC Configuration Register (LPC I/F—D31:F0) 217
8.1.25 COM_DEC—LPC I/F Communication Port Decode Ranges (LPC I/F—D31:F0) 218
8.1.26 FDD/LPT_DEC—LPC I/F FDD & LPT Decode Ranges (LPC I/F—D31:F0) 218
8.1.27 SND_DEC—LPC I/F Sound Decode Ranges (LPC I/F—D31:F0) 219
8.1.28 FWH_DEC_EN1—FWH Decode Enable 1 Register (LPC I/F—D31:F0) 219
8.1.29 GEN1_DEC—LPC I/F Generic Decode Range 1 (LPC I/F—D31:F0) 220
8.1.30 LPC_EN—LPC I/F Enables (LPC I/F—D31:F0) 221
8.1.31 FWH_SEL1—FWH Select 1 Register (LPC I/F—D31:F0) 222
8.1.32 GEN2_DEC—LPC I/F Generic Decode Range 2 (LPC I/F—D31:F0) 223
8.1.33 FWH_SEL2—FWH Select 2 Register (LPC I/F—D31:F0) 223
8.1.34 FWH_DEC_EN2—FWH Decode Enable 2 Register (LPC I/F—D31:F0) 224
8.1.35 FUNC_DIS—Function Disable Register (LPC I/F—D31:F0) 225
8.2 DMA I/O Registers 226
8.2.1 DMABASE_CA—DMA Base and Current Address Registers 227
8.2.2 DMABASE_CC—DMA Base and Current Count Registers 228
8.2.3 DMAMEM_LP—DMA Memory Low Page Registers 228
8.2.4 DMACMD—DMA Command Register 229
8.2.5 DMASTS—DMA Status Register 229
8.2.6 DMA_WRSMSK—DMA Write Single Mask Register 230
8.2.7 DMACH_MODE—DMA Channel Mode Register 230
8.2.8 DMA Clear Byte Pointer Register 231
8.2.9 DMA Master Clear Register 231
8.2.10 DMA_CLMSK—DMA Clear Mask Register 231
8.2.11 DMA_WRMSK—DMA Write All Mask Register 232
8.3 Timer I/O Registers 232
8.3.1 TCW—Timer Control Word Register 233
8.3.1.1 RDBK_CMD—Read Back Command 233
8.3.1.2 LTCH_CMD—Counter Latch Command 234
8.3.2 SBYTE_FMT—Interval Timer Status Byte Format Register 235
8.3.3 Counter Access Ports Register 236
8.4 8259 Interrupt Controller (PIC) Registers 236
8.4.1 Interrupt Controller I/O MAP 236
8.4.2 ICW1—Initialization Command Word 1 Register 237
8.4.3 ICW2—Initialization Command Word 2 Register 238
8.4.4 ICW3—Master Controller Initialization Command Word 3 Register 238
8.4.5 ICW3—Slave Controller Initialization Command Word 3 Register 239
8.4.6 ICW4—Initialization Command Word 4 Register 239
8.4.7 OCW1—Operational Control Word 1 (Interrupt Mask) Register 239
8.4.8 OCW2—Operational Control Word 2 Register 240

8.4.9 OCW3—Operational Control Word 3 Register 241
8.4.10 ELCR1—Master Controller Edge/Level Triggered Register 242
8.4.11 ELCR2—Slave Controller Edge/Level Triggered Register 243
8.5 Advanced Interrupt Controller (APIC) 244
8.5.1 APIC Register Map 244
8.5.2 IND—Index Register 244
8.5.3 DAT—Data Register 245
8.5.4 IRQPA—IRQ Pin Assertion Register 245
8.5.5 EOIR—EOI Register 246
8.5.6 ID—Identification Register 246
8.5.7 VER—Version Register 247
8.5.8 ARBID—Arbitration ID Register 247
8.5.9 BOOT_CONFIG—Boot Configuration Register 247
8.5.10 Redirection Table 248
8.6 Real Time Clock Registers 250
8.6.1 I/O Register Address Map 250
8.6.2 Indexed Registers 250
8.6.2.1 RTC_REGA—Register A 251
8.6.2.2 RTC_REGB—Register B (General Configuration) 252
8.6.2.3 RTC_REGC—Register C (Flag Register) 253
8.6.2.4 RTC_REGD—Register D (Flag Register) 253
8.7 Processor Interface Registers 254
8.7.1 NMI_SC—NMI Status and Control Register 254
8.7.2 NMI_EN—NMI Enable (and Real Time Clock Index) 254
8.7.3 PORT92—Fast A20 and Init Register 255
8.7.4 COPROC_ERR—Coprocessor Error Register 255
8.7.5 RST_CNT—Reset Control Register 255
8.8 Power Management Registers (D31:F0) 256
8.8.1 Power Management PCI Configuration Registers (D31:F0) 256
8.8.1.1 GEN_PMCON_1—General PM Configuration 1 Register (PM—D31:F0) 257
8.8.1.2 GEN_PMCON_2—General PM Configuration 2 Register (PM—D31:F0) 257
8.8.1.3 GEN_PMCON_3—General PM Configuration 3 Register (PM—D31:F0) 258
8.8.1.4 GPI_ROUT—GPI Routing Control Register (PM—D31:F0) 258
8.8.1.5 TRP_FWD_EN—IO Monitor Trap Forwarding Enable Register (PM—D31:F0) 259
8.8.1.6 MON[n]_TRP_RNG—I/O Monitor [4:7] Trap Range Register for Devices 4–7 (PM—D31:F0) 260
8.8.1.7 MON_TRP_MSK—I/O Monitor Trap Range Mask Register for Devices 4–7 (PM—D31:F0) 260
8.8.2 APM I/O Decode 261
8.8.2.1 APM_CNT—Advanced Power Management Control Port Register 261
8.8.2.2 APM_STS—Advanced Power Management Status Port Register 261
8.8.3 Power Management I/O Registers 262
8.8.3.1 PM1_STS—Power Management 1 Status Register 263
8.8.3.2 PM1_EN—Power Management 1 Enable Register 264
8.8.3.3 PM1_CNT—Power Management 1 Control Register 265
8.8.3.4 PM1_TMR—Power Management 1 Timer Register 265
8.8.3.5 PROC_CNT—Processor Control Register 266
8.8.3.6 GPE0_STS—General Purpose Event 0 Status Register 267

8.8.3.7 GPE0_EN—General Purpose Event 0 Enables Register 268
8.8.3.8 GPE1_STS—General Purpose Event 1 Status Register 269
8.8.3.9 GPE1_EN—General Purpose Event 1 Enable Register 270
8.8.3.10 SMI_EN—SMI Control and Enable Register 270
8.8.3.11 SMI_STS—SMI Status Register 272
8.8.3.12 MON_SMI—Device Monitor SMI Status and Enable Register 273
8.8.3.13 DEVACT_STS—Device Activity Status Register 274
8.8.3.14 DEVTRAP_EN—Device Trap Enable Register 275
8.8.3.15 BUS_ADDR_TRACK—Bus Address Tracker Register 276
8.8.3.16 BUS_CYC_TRACK—Bus Cycle Tracker Register 276
8.9 System Management TCO Registers (D31:F0) 277
8.9.1 TCO Register I/O Map 277
8.9.2 TCO1_RLD—TCO Timer Reload and Current Value Register 277
8.9.3 TCO1_TMR—TCO Timer Initial Value Register 278
8.9.4 TCO1_DAT_IN—TCO Data In Register 278
8.9.5 TCO1_DAT_OUT—TCO Data Out Register 278
8.9.6 TCO1_STS—TCO1 Status Register 278
8.9.7 TCO2_STS—TCO2 Status Register 280
8.9.8 TCO1_CNT—TCO1 Control Register 281
8.9.9 TCO2_CNT—TCO2 Control Register 282
8.9.10 TCO_MESSAGE1 and TCO_MESSAGE2 Registers 282
8.9.11 TCO_WDSTATUS—TCO2 Control Register 282
8.9.12 SW_IRQ_GEN—Software IRQ Generation Register 283
8.10 General Purpose I/O Registers (D31:F0) 283
8.10.1 GPIO Register I/O Address Map 285
8.10.2 GPIO_USE_SEL—GPIO Use Select Register 285
8.10.3 GP_IO_SEL—GPIO Input/Output Select Register 286
8.10.4 GP_LVL—GPIO Level for Input or Output Register 287
8.10.5 GPO_BLINK—GPO Blink Enable Register 288
8.10.6 GPI_INV—GPIO Signal Invert Register 289

9 IDE Controller Registers (D31:F1) 291

9.1 PCI Configuration Registers (IDE—D31:F1) 291
9.1.1 VID—Vendor ID Register (IDE—D31:F1) 292
9.1.2 DID—Device ID Register (IDE—D31:F1) 292
9.1.3 CMD—Command Register (IDE—D31:F1) 292
9.1.4 STS—Device Status Register (IDE—D31:F1) 293
9.1.5 RID—Revision ID Register (HUB-PCI—D30:F0) 293
9.1.6 PI—Programming Interface (IDE—D31:F1) 293
9.1.7 SCC—Sub Class Code (IDE—D31:F1) 294
9.1.8 BCC—Base Class Code (IDE—D31:F1) 294
9.1.9 MLT—Master Latency Timer (IDE—D31:F1) 294
9.1.10 BM_BASE—Bus Master Base Address Register (IDE—D31:F1) 294
9.1.11 IDE_SVID—Subsystem Vendor ID (IDE—D31:F1) 295
9.1.12 IDE_SID—Subsystem ID (IDE—D31:F1) 295
9.1.13 IDE_TIM—IDE Timing Register (IDE—D31:F1) 295
9.1.14 SLV_IDETIM—Slave (Drive 1) IDE Timing Register (IDE—D31:F1) 297
9.1.15 SDMA_CNT—Synchronous DMA Control Register (IDE—D31:F1) 298

9.1.16 SDMA_TIM—Synchronous DMA Timing Register (IDE—D31:F1) 298
9.1.17 IDE_CONFIG—IDE I/O Configuration Register 299
9.2 Bus Master IDE I/O Registers (D31:F1) 301
9.2.1 BMIC[P,S]—Bus Master IDE Command Register 301
9.2.2 BMIS[P,S]—Bus Master IDE Status Register 302
9.2.3 BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register 302

10 USB Controller Registers (D31:F2) 303

10.1 PCI Configuration Registers (D31:F2) 303
10.1.1 VID—Vendor Identification Register (USB—D31:F2) 303
10.1.2 DID—Device Identification Register (USB—D31:F2) 304
10.1.3 CMD—Command Register (USB—D31:F2) 304
10.1.4 STA—Device Status Register (USB—D31:F2) 305
10.1.5 RID—Revision Identification Register (USB—D31:F2) 305
10.1.6 PI—Programming Interface (USB—D31:F2) 305
10.1.7 SCC—Sub Class Code Register (USB—D31:F2) 306
10.1.8 BCC—Base Class Code Register (USB—D31:F2) 306
10.1.9 MLT—Master Latency Timer Register (USB—D31:F2) 306
10.1.10 BASE—Base Address Register (USB—D31:F2) 306
10.1.11 SVID—Subsystem Vendor ID (USB—D31:F2) 307
10.1.12 SID—Subsystem ID (USB—D31:F2) 307
10.1.13 INTR_LN—Interrupt Line Register (USB—D31:F2) 307
10.1.14 INTR_PN—Interrupt Pin Register (USB—D31:F2) 307
10.1.15 SB_RELNUM—Serial Bus Release Number Register (USB—D31:F2) 308
10.1.16 USB_LEGKEY—USB Legacy Keyboard/Mouse Control Register (USB—D31:F2) 308
10.1.17 USB_RES—USB Resume Enable Register (USB—D31:F2) 309
10.2 USB I/O Registers 310
10.2.1 USBCMD—USB Command Register 311
10.2.2 USBSTA—USB Status Register 314
10.2.3 USBINTR—Interrupt Enable Register 315
10.2.4 FRNUM—Frame Number Register 315
10.2.5 FRBASEADD—Frame List Base Address 316
10.2.6 SOFMOD—Start of Frame Modify Register 316
10.2.7 PORTSC[0,1]—Port Status and Control Register 317

11 SMBus Controller Registers (D31:F3) 319

11.1 PCI Configuration Registers (SMBUS—D31:F3) 319
11.1.1 VID—Vendor Identification Register (SMBUS—D31:F3) 319
11.1.2 DID—Device Identification Register (SMBUS—D31:F3) 320
11.1.3 CMD—Command Register (SMBUS—D31:F3) 320
11.1.4 STA—Device Status Register (SMBUS—D31:F3) 321
11.1.5 RID—Revision ID Register (SMBUS—D31:F3) 321
11.1.6 PI—Programming Interface (SMBUS—D31:F3) 321
11.1.7 SCC—Sub Class Code Register (SMBUS—D31:F3) 322
11.1.8 BCC—Base Class Code Register (SMBUS—D31:F3) 322
11.1.9 SMB_BASE—SMBus Base Address Register (SMBUS—D31:F3) 322
11.1.10 SVID—Subsystem Vendor ID (SMBUS—D31:F2/F4) 322

11.1.11 SID—Subsystem ID (SMBUS—D31:F2/F4) 323
11.1.12 INTR_LN—Interrupt Line Register (SMBUS—D31:F3) 323
11.1.13 INTR_PN—Interrupt Pin Register (SMBUS—D31:F3) 323
11.1.14 HOSTC—Host Configuration Register (SMBUS—D31:F3) 323
11.2 SMBus I/O Registers 324
11.2.1 HST_STS—Host Status Register 325
11.2.2 HST_CNT—Host Control Register 326
11.2.3 HST_CMD—Host Command Register 328
11.2.4 XMIT_SLVA—Transmit Slave Address Register 328
11.2.5 HST_D0—Data 0 Register 328
11.2.6 HST_D1—Data 1 Register 328
11.2.7 BLOCK_DB—Block Data Byte Register 329
11.2.8 RCV_SLVA—Receive Slave Address Register 329
11.2.9 SLV_DATA—Receive Slave Data Register 329
11.2.10 SMLINK_PIN_CTL—SMLINK Pin Control Register 330
11.2.11 SMBUS_PIN_CTL—SMBus Pin Control Register 330

12 Serial I/O Unit 331

12.1 Features 331
12.2 Pin Description 332
12.2.1 Universal Asynchronous Receive And Transmit (UART0, UART1) 332
12.2.2 SIU LPC Interface 334
12.3 Functional Description 334
12.3.1 Host Processor Interface (LPC) 334
12.4 LPC Interface 335
12.4.1 LPC Cycles 335
12.4.1.1 I/O Read and Write Cycles 335
12.4.1.2 DMA Read and Write Cycles 335
12.4.1.3 DMA Protocol 335
12.4.1.4 DMA Arbitration 336
12.4.2 Reset Policy 336
12.4.3 LPC Transfers 336
12.4.3.1 I/O Transfers 336
12.4.3.2 DMA Transfers 336
12.5 Logical Device 4 & 5: Serial Ports (UARTs) 337
12.5.1 Overview 337
12.5.1.1 UART Feature List 338
12.5.2 UART Operational Description 339
12.5.3 Internal Register Descriptions 340
12.5.3.1 Receive Buffer Register (RBR) 341
12.5.3.2 Transmit Holding Register (THR) 341
12.5.3.3 Interrupt Enable Register (IER) 342
12.5.3.4 Interrupt Identification Register (IIR) 343
12.5.3.5 FIFO Control Register (FCR) 345
12.5.3.6 Line Control Register (LCR) 346
12.5.3.7 Line Status Register (LSR) 348
12.5.3.8 Modem Control Register (MCR) 350
12.5.3.9 Modem Status Register (MSR) 351
12.5.3.10 Scratchpad Register (SCR) 352
12.5.3.11 Programmable Baud Rate Generator 352
12.5.3.12 Divisor Latch Register Low (DLL) 352
12.5.3.13 Divisor Latch Register High (DLH) 352

12.5.4 FIFO Operation 353
12.5.4.1 FIFO Interrupt Mode Operation 353
12.5.4.2 FIFO Polled Mode Operation 354
12.5.5 DMA Requests 354
12.5.5.1 Trailing Bytes in the Receive FIFO 354
12.6 Serial IRQ 355
12.6.1 Timing Diagrams For SIU_SERIRQ Cycle 355
12.6.1.1 SIU_SERIRQ Cycle Control 355
12.6.1.2 SIU_SERIRQ Data Frame 356
12.6.1.3 Stop Cycle Control 357
12.6.1.4 Latency 357
12.6.1.5 EOI/ISR Read Latency 357
12.6.1.6 Reset and Initialization 357
12.7 Configuration 358
12.7.1 Configuration Port Address Selection 358
12.7.2 Primary Configuration Address Decoder 358
12.7.2.1 Entering the Configuration State 358
12.7.2.2 Exiting the Configuration State 359
12.7.2.3 Configuration Sequence 359
12.7.2.4 Configuration Mode 359
12.7.3 SIU Configuration Registers Summary 360
12.7.3.1 Global Control/Configuration Registers [00h — 2Fh] 361
12.7.3.2 Logical Device Configuration Registers [30h — FFh] 362

A Signal Description 365

A.1 Hub Interface to Host Controller 365
A.2 Link to LAN Connect 365
A.3 EEPROM Interface 366
A.4 Firmware Hub Interface 366
A.5 PCI Interface 366
A.6 IDE Interface 369
A.7 LPC Interface 370
A.8 Interrupt Interface 370
A.9 USB Interface 371
A.10 Power Signals 371
A.11 Processor Interface 372
A.12 SMBus Interface 373
A.13 System Management Interface 373
A.14 Real Time Clock Interface 374
A.15 Other Clocks 374
A.16 Universal Asynchronous Receive and Transmit (UART0, UART1) 374
A.17 SIU LPC Interface 376
A.18 Miscellaneous Signals 376
A.19 General Purpose I/O 377
A.20 Power and Ground 378

B I/O Register Index 379

Index 389

Figures

1 Intel® 82801E C-ICH Simplified Block Diagram 30
2 System Configuration 31
3 Primary Device Status Register Error Reporting Logic 37
4 Secondary Device Status Register Error Reporting Logic 37
5 NMI# Generation Logic 38
6 Integrated LAN Controller Block Diagram 41
7 64-Word EEPROM Read Instruction Waveform 48
8 LPC Interface Diagram 52
9 Typical Timing for LFRAME# 56
10 Abort Mechanism 56
11 Intel® 82801E C-ICH DMA Controller 58
12 DMA Serial Channel Passing Protocol 62
13 DMA Request Assertion Through LDRQ# 66
14 Port 60 Read Clearing IRQ1/12 Latch 78
15 Coprocessor Error Timing Diagram 96
16 Signal Strapping 97
17 Latching CPU I/F Signals with STOPCLK# 102
18 Physical Region Descriptor Table Entry 116
19 Transfer Descriptor 123
20 Example Queue Conditions 131
21 USB Data Encoding 134
22 USB Legacy Keyboard Flow Diagram 143
23 FWH Memory Cycle Preamble 157
24 Single Byte Read 157
25 Single Byte Write 158
26 System Clock Domains 159
27 Conceptual System Clock Diagram 160
28 SIU Block Diagram 332
29 Example UART Data Frame 339
30 Start Frame Timing with Source Sampled a Low Pulse on IRQ1 355
31 Stop Frame Timing with Host Using 17 SIU_SERIRQ Sampling Period 355

Tables

1 Related Documents 25
2 Industry Specifications 25
3 PCI Devices and Functions 29
4 Type 0 Configuration Cycle Device Number Translation 39
5 EEPROM Address Map 49
6 LPC Cycle Types Supported 52
7 Start Field Bit Definitions 53
8 Cycle Type Bit Definitions 53
9 Transfer Size Bit Definition 54
10 SYNC Bit Definition 54
11 Intel® 82801E C-ICH Response to Sync Failures 55
12 DMA Transfer Size 59
13 Address Shifting in 16-bit I/O DMA Transfers 60
14 DMA Cycle vs. I/O Address 64
15 PCI Data Bus vs. DMA I/O Port Size 64
16 DMA I/O Cycle Width vs. BE[3:0]# 65
17 Counter Operating Modes 70
18 Interrupt Controller Core Connections 72
19 Interrupt Status Registers 73
20 Content of Interrupt Vector Byte 73
21 APIC Interrupt Mapping 79
22 Arbitration Cycles 81
23 APIC Message Formats 81
24 EOI Message 82
25 Short Message 82
26 APIC Bus Status Cycle Definition 83
27 Lowest Priority Message (Without Focus Processor) 84
28 Remote Read Message 85
29 Interrupt Message Address Format 88
30 Interrupt Message Data Format 89
31 Stop Frame Explanation 90
32 Data Frame Format 91
33 Configuration Bits Reset By RTCRST# Assertion 94
34 INIT# Going Active 96
35 NMI Sources 96
36 Frequency Strap Bit Mapping 97
37 General Power States for Systems Using 82801E C-ICH 98
38 State Transition Rules for 82801E C-ICH 99
39 Causes of SMI# and SCI 100
40 Transitions Due to RI# Signal 104
41 Write Only Registers with Read Paths in Alternate Access Mode 105
42 PIC Reserved Bits Return Values 107
43 Register Write Accesses in Alternate Access Mode 107
44 Intel® 82801E C-ICH Clock Inputs 108
45 Alert on LAN* Message Data 112
46 IDE Transaction Timings (PCI Clocks) 115
47 Interrupt/Active Bit Interaction Definition 118
48 UltraATA/33 Control Signal Redefinitions 119
49 Frame List Pointer Bit Description 122

50 TD Link Pointer 123
51 TD Control and Status 124
52 TD Token 126
53 TD Buffer Pointer 127
54 Queue Head Block 127
55 Queue Head Link Pointer 127
56 Queue Element Link Pointer 128
57 Command Register, Status Register and TD Status Bit Interaction 130
58 Queue Advance Criteria 132
59 USB Schedule List Traversal Decision Table 133
60 PID Format 135
61 PID Types 135
62 Address Field 136
63 Endpoint Field 136
64 Token Format 137
65 SOF Packet 138
66 Data Packet Format 138
67 Bits Maintained 142
68 USB Legacy Keyboard State Transitions 143
69 Quick Protocol 145
70 Send/Receive Byte Protocol 146
71 Write Byte/Word Protocol 146
72 Read Byte/Word Protocol 147
73 Process Call Protocol 148
74 Block Read/Write Protocol 149
75 I^2C Block Read 150
76 Slave Write Cycle Format 153
77 Slave Write Registers 153
78 Command Types 154
79 Read Cycle Format 154
80 Data Values for Slave Read Registers 155
81 PCI Devices and Functions 162
82 Fixed I/O Ranges Decoded by 82801E C-ICH 163
83 Variable I/O Decode Ranges 165
84 Memory Decode Ranges from Processor Perspective 166
85 PCI Configuration Map (LAN Controller—B1:D8/D9:F0) 170
86 Configuration of Subsystem ID and Subsystem Vendor ID via EEPROM 176
87 Data Register Structure 179
88 Intel® 82801E C-ICH Integrated LAN Controller CSR Space 179
89 Self-Test Results Format 184
90 Statistical Counters 189
91 PCI Configuration Map (HUB-PCI—D30:F0) 191
92 PCI Configuration Map (LPC I/F—D31:F0) 205
93 DMA Registers 226
94 PIC Registers 236
95 APIC Direct Registers 244
96 APIC Indirect Registers 244
97 RTC I/O Registers 250
98 RTC (Standard) RAM Bank 250
99 PCI Configuration Map (PM—D31:F0) 256

100 APM Register Map 261
101 ACPI and Legacy I/O Register Map 262
102 TCO I/O Register Map 277
103 Summary of GPIO Implementation 284
104 Registers to Control GPIO 285
105 PCI Configuration Map (IDE—D31:F1) 291
106 Bus Master IDE I/O Registers 301
107 PCI Configuration Map (USB—D31:F2) 303
108 USB I/O Registers 310
109 Run/Stop, Debug Bit Interaction SWDBG (Bit 5), Run/Stop (Bit 0) Operation 312
110 PCI Configuration Registers (SMBUS—D31:F3) 319
111 SMB I/O Registers 324
112 Universal Asynchronous Receive And Transmit (UART0, UART1) 332
113 SIU Interface 334
114 Address Map 334
115 Supported LPC Cycle Types 335
116 I/O Sync Bits Description 336
117 Baud Rate Examples 338
118 SIU Signal Reset States 340
119 Internal Register Descriptions 340
120 Interrupt Conditions 343
121 Interrupt Identification Register Decode 344
122 SIU_SERIRQ Sampling Periods 356
123 Configuration Registers Summary 360
124 Global Control Registers 361
125 Logical Device 4 (Serial Port 0) 362
126 Logical Device 5 (Serial Port 1) 363
127 Hub Interface Signals 365
128 LAN Connect Interface Signals 365
129 EEPROM Interface Signals 366
130 Firmware Hub Interface Signals 366
131 PCI Interface Signals 366
132 IDE Interface Signals 369
133 LPC Interface Signals 370
134 Interrupt Signals 370
135 USB Interface Signals 371
136 Power Signals 371
137 Processor Interface Signals 372
138 SMBus Interface Signals 373
139 System Management Interface Signals 373
140 Real Time Clock Interface 374
141 Other Clocks 374
142 Universal Asynchronous Receive And Transmit (UART 0, 1) 374
143 SIU Interface 376
144 Miscellaneous Signals 376
145 General Purpose I/O Signals 377
146 Power and Ground Signals 378
147 Intel® 82801E C-ICH Fixed I/O Registers 379
148 Intel® 82801E C-ICH Variable I/O Registers 384
149 Serial I/O Unit Registers 388

Revision History

Date	Revision	Description
January 2002	001	First release of this manual.

Guide to this Manual 1

This document provides a detailed description of the 82801E C-ICH's functions and capabilities, including, signals, registers, and on-chip functional units. This document is intended for original equipment manufacturers and BIOS vendors who are creating 82801E C-ICH-based products. This document assumes a working knowledge of the vocabulary and principles of USB, IDE, SMBus, PCI, ACPI, LAN, LPC, and serial I/O. For thermal, electrical, and mechanical information, refer to the *Intel® 82801E Communications I/O Controller Hub (C-ICH)* datasheet (order number 273598).

Although some details of these features are described in this document, refer to the individual industry specifications listed in Table 2 for complete details.

1.1 Manual Contents

This manual contains 12 chapters, two appendices, and an index. This section summarizes the content of the remaining chapters. The remainder of the chapter describes conventions and special terminology used throughout the manual and provides references to related documentation.

Chapter 2, "Introduction"
Chapter 2 introduces the 82801E C-ICH and provides information on document organization. This chapter also describes the key features of the 82801E C-ICH and provides a brief description of its major functions.

Chapter 3, "Functional Description"
Chapter 3 provides a detailed description of the functions in the 82801E C-ICH. All PCI buses, devices, and functions in this manual are abbreviated using the following nomenclature; Bus:Device:Function. This datasheet abbreviates buses as B0 and B1, devices as D8, D30 and D31 and functions as F0, F1, F2, F3, F4, F5 and F6. For example, Device 31 Function 5 is abbreviated as D31:F5, and Bus 1 Device 8 Function 0 is abbreviated as B1:D8:F0. Generally, the bus number will not be used, and can be considered to be Bus 0. Note that the 82801E C-ICH's external PCI bus is typically Bus 1; however, it may be assigned a different number depending on system configuration.

Chapter 4, "System Clock Domains"
Chapter 4 provides a list of each clock domain associated with the 82801E C-ICH in an 82801E C-ICH-based system.

Chapter 5, "Register and Memory Mapping"
Chapter 5 provides an overview of the registers, fixed I/O ranges, variable I/O ranges, and memory ranges decoded by the 82801E C-ICH.

Chapter 6, "LAN Controller Registers (B1:D8/D9:F0)"
Chapter 6 provides a detailed description of all registers that reside in the 82801E C-ICH's integrated LAN Controller. The integrated LAN Controller resides on the 82801E C-ICH's external PCI bus (typically Bus 1) at Device 8, Function 0 (B1:D8:F0).

Chapter 7, "Hub Interface to PCI Bridge Registers (D30:F0)"
Chapter 7 provides a detailed description of all registers that reside in the Hub Interface to PCI bridge. This bridge resides at Device 30, Function 0 (D30:F0).

Chapter 8, “LPC Interface Bridge Registers (D31:F0)”
Chapter 8 provides a detailed description of all registers that reside in the LPC bridge. This bridge resides at Device 31, Function 0 (D31:F0). This function contains registers for many different units within the 82801E C-ICH, including DMA, Timers, Interrupts, CPU Interface, GPIO, Power Management, System Management and RTC.

Chapter 9, “IDE Controller Registers (D31:F1)”
Chapter 9 provides a detailed description of all registers that reside in the IDE controller. This controller resides at Device 31, Function 1 (D31:F1).

Chapter 10, “USB Controller Registers (D31:F2)”
Chapter 10 provides a detailed description of all registers that reside in the two USB controllers. These controllers reside at Device 31, Functions 2 and 4 (D31:F2/F4).

Chapter 11, “SMBus Controller Registers (D31:F3)”
Chapter 11 provides a detailed description of all registers that reside in the SMBus controller. This controller resides at Device 31, Function 3 (D31:F3).

Chapter 12, “Serial I/O Unit”
Chapter 12 provides a detailed description of all registers that reside in the Serial I/O unit. This unit contains the LPC interface and the UARTs.

Appendix A, “Signal Description”
Appendix A provides a detailed description of each 82801E C-ICH signal. Signals are arranged according to interface. Details are provided about the drive characteristics (Input/Output, Open Drain, etc.) of all signals.

Appendix B, “I/O Register Index”
This index lists registers.

1.2 Text Conventions

The following notations may be used throughout this manual.

#	The pound symbol (#) appended to a signal name indicates that the signal is active low.
Variables	Variables are shown in italics. Variables must be replaced with correct values.
Instructions	Instruction mnemonics are shown in uppercase. When you are programming, instructions are not case-sensitive. You may use either uppercase or lowercase.
Numbers	Hexadecimal numbers are represented by a string of hexadecimal digits followed by the character *h*. A zero prefix may be added to numbers that begin with *A* through *F*. (For example, *FF* is shown as *0FFh*.) Decimal and binary numbers are represented by their customary notations. (That is, 255 is a decimal number and 1111 1111 is a binary number. A letter *B* or *b* may be added to binary numbers for clarity.)
Units of Measure	The following abbreviations are used to represent units of measure:

A	amps, amperes
Gbyte	gigabytes
Kbyte	kilobytes
KΩ	kilo-ohms
mA	milliamps, milliamperes
Mbyte	megabytes
MHz	megahertz
ms	milliseconds
mW	milliwatts
ns	nanoseconds
pF	picofarads
W	watts
V	volts
µA	microamps, microamperes
µF	microfarads
µs	microseconds
µW	microwatts

Signal Names	Signal names are shown in uppercase. When several signals share a common name, an individual signal is represented by the signal name followed by a number (in some cases the number is in brackets), while the group is represented by the signal name followed by a variable (*n*). For example, AD[0] through AD[31] may be collectively called AD*n*. A pound symbol (#) appended to a signal name identifies an active-low signal.
Bit Mnemonics	For register bits with a mnemonic, the mnemonic is shown in uppercase. When several bits share a common mnemonic, an individual bit is represented by the bit name followed by a number, while the group is represented by the bit name followed by a variable (*n*).

1.3 Technical Support

1.3.1 Electronic Support Systems

Intel's site on the World Wide Web (http://www.intel.com/) provides up-to-date technical information and product support. This information is available 24 hours per day, 7 days per week, providing technical information whenever you need it.

1.3.2 Telephone Technical Support

In the U.S. and Canada, technical support representatives are available to answer your questions between 5 a.m. and 5 p.m. PST. You can also fax your questions. (Please include your voice telephone number and indicate whether you prefer a response by phone or by fax). Outside the U.S. and Canada, please contact your local distributor.

1-800-628-8686	U.S. and Canada
916-356-7599	U.S. and Canada
916-356-6100 (fax)	U.S. and Canada

1.4 Product Literature

You can order product literature from the following Intel literature centers.

1-800-548-4725	U.S. and Canada
708-296-9333	U.S. (from overseas)
44(0)1793-431155	Europe (U.K.)
44(0)1793-421333	Germany
44(0)1793-421777	France
81(0)120-47-88-32	Japan (fax only)

1.4.1 Related Documents

The following Intel documents contain additional information about designing systems that incorporate the 82801E C-ICH. A complete list of documents can be found on the Intel Developer's Web site (http://www.intel.com/design).

Table 1. Related Documents

Document	Order Number
Intel® 82801E Communications I/O Controller Hub (C-ICH) Datasheet	273598
Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update	273645
Intel® 82801E Communications I/O Controller Hub (C-ICH) Platform Design Guide	273671
Intel® 810E Chipset: 82810E Graphics and Memory Controller Hub (GMCH) Datasheet	290676
82802AB/82802AC Firmware Hub (FWH) Datasheet	290658

Table 2 lists industry specifications referenced in this manual.

Table 2. Industry Specifications

Specification	Location
LPC	http://developer.intel.com/design/chipsets/industry/lpc.htm
WfM	http://developer.intel.com/ial/WfM/usesite.htm
SMBus	http://www.sbs-forum.org/specs/
PCI	http://pcisig.com/
USB	http://www.usb.org
ACPI	http://www.teleport.com/~acpi/

Introduction 2

The Intel® 82801E Communications I/O Controller Hub (82801E C-ICH) is a highly integrated multifunctional communications I/O controller hub that provides the interface to the PCI bus and integrates many of the functions needed in today's communications applications. This document provides a detailed description of the 82801E C-ICH functions and capabilities including, signals, registers, on-chip functional units, and bus interfaces. For electrical, thermal and mechanical specifications, refer to the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Datasheet* (order number 273598).

2.1 Feature Summary

- Supports Intel Celeron™ processors, Pentium® III processors, future 0.13 micron processors, and 82810E GMCH and 82815E GMCH chipsets
  - — Includes speed strapping logic to reduce external glue
- 8-Bit hub interface
  - — 266 Mbyte/s maximum throughput
  - — Parallel termination scheme for longer trace lengths
- Two integrated LAN controllers
  - — External LAN Connect using new Jordan I/F
  - — 10/100 Mbit/s Ethernet support
- PCI Bus interface
  - — Supports PCI Rev 2.2 specification at 33 MHz
  - — 133 Mbyte/s maximum throughput
  - — Supports four master devices on PCI (LANs do *not* count against these four)
  - — One PCI REQ/GNT pair can be given higher arbitration priority. This is intended for the external 1394 host controller.
  - — Support for 44-bit addressing on PCI, using DAC protocol.
- Low-Pincount (LPC) interface
  - — Allows connection of devices such as Super I/O, microcontrollers, customer ASICs
  - — Supports two master/DMA devices
- Firmware Hub (FWH) interface
  - — Multiplexed with LPC interface
  - — Supports 8-Mbyte memory size
- Integrated IDE controller
  - — Supports Ultra100 DMA mode transfers up to 100 Mbytes/s for reads from the disk; 88.88 Mbytes/s for writes to the disk
  - — Supports the Ultra66 and Ultra33 DMA Mode(s) with transfers up to 66 Mbytes/s
  - — PIO Mode 4 transfers up to 14 Mbytes/s
  - — Independent timing of up to four drives, with separate IDE connections for primary and secondary cables
  - — Supports tri-state modes to enable mobile swap bay

- USB
  - Includes one UHCI Host Controller with a total of two ports
  - USB 1.1 compliant
  - Supports legacy keyboard and mouse software with USB-based keyboard and mouse
  - 5 V tolerance on overcurrent input signals
- Interrupt Controller
  - Supports four extra PCI Interrupt pins (total of eight) for workstations and servers
  - Two cascaded 82C59 interrupt controllers
  - Integrated I/O (x) APIC supporting 24 interrupts
  - 15 interrupts supported in 8259 mode
  - Serial interrupt input for ISA-compatible and PCI interrupts
  - Supports PCI scheme for delivering interrupts as write cycles (rather than via PIRQ [A:H]#)
  - Supports front-side message interrupt delivery
- DMA Controller
  - Two cascaded 8237 DMA controllers
  - PCI DMA: Supports PC/PCI protocol with up to two PC/PCI REQ#/GNT# pairs
  - Supports LPC DMA
  - Supports DMA collection buffer (CAT mode) to emulate Type-F DMA for all DMA channels
- Integrated 82C54-compatible timers
  - System timer, refresh request, speaker tone output
- Real-time clock with 256-byte battery-backed CMOS RAM
  - Includes SMI# for century rollover
- System TCO Reduction Circuits
  - Timers to generate SMI# and reset upon detection of halted CPU
  - Timers to detect improper CPU reset
  - Interrupt capability to OS-specific manageability extension and OS capability to call TCO BIOS
  - Alert on LAN (AOL) to enable heartbeats and system event reporting via LAN controller
  - Supports CPU BIST
  - Detection of unprogrammed Flash
- System Management Bus (SMBus)
  - Host interface allows CPU to communicate via SMBus
  - Compatible with most two-wire components that are also I^2C compatible
  - Slave interface allows external microcontroller to access system resources
- GPIO
  - Exact number varies by configuration. Maximum: 12 inputs, eight outputs, four I/O
- Integrated 16550 compatible UARTs
  - Two UARTs
  - Serial Interrupts
  - Can disable when external serial I/O controller is used

- Supports IRQ1/IRQ12 emulation to remove the need for an external keyboard controller
- 1.8 V operation with 3.3 V I/O. 5 V tolerance on many buffers, including PCI, IDE
- Package: 421 BGA

2.2 Overview

The 82801E C-ICH provides extensive I/O support. Functions and capabilities include:

- PCI Rev 2.2 compliant with support for 33 MHz PCI operations
- PCI slots support up to four Req/Gnt pairs
- Enhanced DMA Controller, Interrupt Controller, and Timer Functions
- Integrated IDE controller supports Ultra ATA100/66/33
- USB host interface with support for two USB ports; one host controller
- Two integrated LAN controllers
- System Management Bus (SMBus) with additional support for I^2C devices
- Low Pin Count (LPC) interface
- Firmware Hub (FWH) interface support
- Alert On LAN* (AOL) and Alert On LAN 2 (AOL2)*
- Serial I/O unit containing two UARTs

The 82801E C-ICH incorporates a variety of PCI functions; these are divided into two logical devices (30 and 31) on PCI Bus 0, and one device on Bus 1. Device 30 is the Hub Interface-To-PCI bridge. Device 31 contains all the other PCI functions, except the LAN Controller, as shown in Table 3. The LAN controllers are located on Bus 1.

Table 3. PCI Devices and Functions

Bus:Device:Function	Function Description
Bus 0:Device 30:Function 0	Hub Interface to PCI Bridge
Bus 0:Device 31:Function 0	PCI to LPC Bridge (includes: DMA, Timers, compatible interrupt controller, APIC, RTC, processor interface control, power management control, system management control, and GPIO control)
Bus 0:Device 31:Function 1	IDE Controller
Bus 0:Device 31:Function 2	USB Controller
Bus 0:Device 31:Function 3	SMBus Controller
Bus 1:Device 8:Function 0	LAN0 Controller
Bus 1:Device 9:Function 0	LAN1 Controller

Figure 1. Intel® 82801E C-ICH Simplified Block Diagram

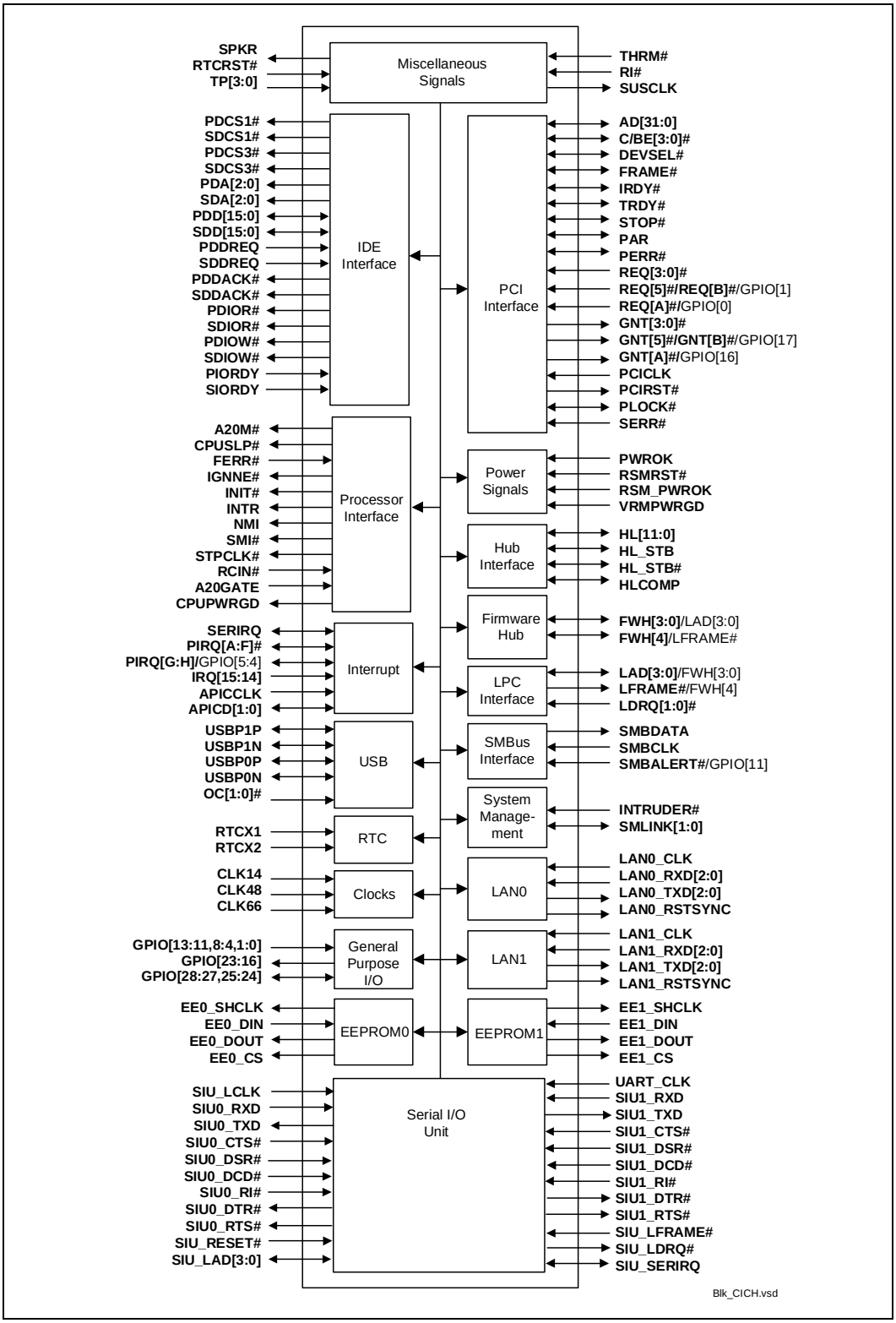
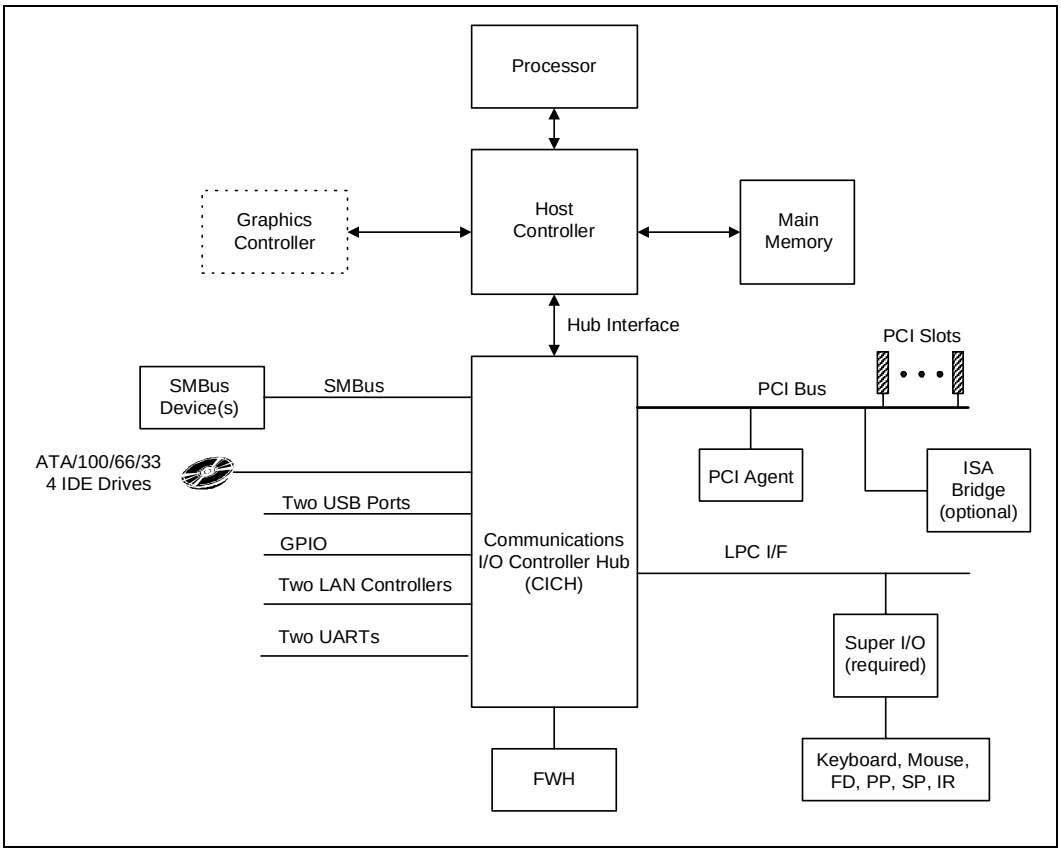


Figure 2. System Configuration



2.3 Capability Summary

The following sections provide an overview of the 82801E C-ICH's capabilities.

Hub Architecture

As I/O speeds increase, the demand placed upon the PCI bus by the I/O bridge has become significant. With the addition of Ultra ATA/100, coupled with the existing USB, I/O requirements could impact PCI bus performance. The chipset's hub interface architecture ensures that the I/O subsystem; both PCI and the integrated I/O features (IDE, USB, etc.), will receive adequate bandwidth. By placing the I/O bridge on the hub interface (instead of PCI), the hub architecture ensures that the I/O functions integrated into the 82801E C-ICH and the PCI peripherals obtain the bandwidth necessary for peak performance.

PCI Interface

The 82801E C-ICH PCI interface provides a 33 MHz, Rev. 2.2 compliant implementation. All PCI signals are 5 V tolerant. The 82801E C-ICH integrates a PCI arbiter that supports up to four external PCI bus masters in addition to the internal 82801E C-ICH requests.

IDE Interface (Bus Master Capability and Synchronous DMA Mode)

The fast IDE interface supports up to four IDE devices; this provides an interface for IDE hard disks and CD ROMs. Each IDE device can have independent timings. The IDE interface supports PIO IDE transfers up to 14 Mbytes/s and Bus Master IDE transfers up 100 Mbytes/s. It does not consume any ISA DMA resources. The IDE interface integrates 16x32-bit buffers for optimal transfers.

The 82801E C-ICH's IDE system contains two independent IDE signal channels. They can be electrically isolated independently. They can be configured to the standard primary and secondary channels (four devices). There are integrated series resistors on the data and control lines (see "IDE Controller (D31:F1)" on page 113 for details).

Low Pin Count (LPC) Interface

The 82801E C-ICH implements an LPC Interface as described in the LPC 1.0 specification. The Low Pin Count (LPC) Bridge function of the 82801E C-ICH resides in PCI Device 31:Function 0. In addition to the LPC bridge interface function, D31:F0 contains other functional units, including DMA, Interrupt Controllers, Timers, Power Management, System Management, GPIO, and RTC.

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

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

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

The 82801E C-ICH supports two types of DMA (LPC and PC/PCI). DMA via LPC is similar to ISA DMA. LPC DMA and PC/PCI DMA use the 82801E C-ICH's DMA controller. The PC/PCI protocol allows PCI-based peripherals to initiate DMA cycles by encoding requests and grants via two PC/PCI REQ#/GNT# pairs.

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

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

The 82801E C-ICH provides an ISA-Compatible interrupt controller that incorporates the functionality of two 82C59 interrupt controllers. The two interrupt controllers are cascaded so that 14 external and two internal interrupts are possible. In addition, the 82801E C-ICH supports a serial interrupt scheme.

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

Advanced Programmable Interrupt Controller (APIC)

In addition to the standard ISA compatible interrupt controller described in the previous section, the 82801E C-ICH incorporates the Advanced Programmable Interrupt Controller (APIC). While the standard interrupt controller is intended for use in a uni-processor system, APIC can be used in either a uni-processor or multi-processor system.

Enhanced Universal Serial Bus (USB) Controller

The USB controller provides enhanced support for the Universal Host Controller Interface (UHCI). This includes support that allows legacy software to use a USB-based keyboard and mouse. The 82801E C-ICH is USB Revision 1.1 compliant. The 82801E C-ICH contains a USB Host Controller. The Host Controller includes a root hub with two separate USB ports. See “USB Controller (Device 31:Function 2)” on page 122 for details.

LAN Controllers

Each of the 82801E C-ICH’s integrated LAN Controllers includes a 32-bit PCI controller that provides enhanced scatter-gather bus mastering capabilities and enables the LAN Controller to perform high speed data transfers over the PCI bus. Its bus master capabilities enable the component to process high-level commands and perform multiple operations; this lowers processor utilization by off-loading communication tasks from the processor. Two large transmit and receive FIFOs of 3 Kbytes each help prevent data underruns and overruns while waiting for bus accesses. This enables the integrated LAN Controller to transmit data with minimum interframe spacing (IFS).

The LAN Controller can operate in full duplex or half duplex mode. In full duplex mode the LAN Controller adheres to the IEEE 802.3x Flow Control specification. Half duplex performance is enhanced by a proprietary collision reduction mechanism. See “LAN Controllers (B1:D8/D9:F0)” on page 40 for details.

RTC

The 82801E C-ICH contains a Motorola* MC146818A-compatible real-time clock with 256 bytes of battery-backed RAM. The real-time clock performs two key functions: keeping track of the time of day and storing system data, even when the system is powered down. The RTC operates on a 32.768 KHz crystal and a separate 3V lithium battery that provides up to seven years of protection.

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

The RTC also supports a date alarm that allows for scheduling a wake up event up to 30 days in advance.

GPIO

Various general purpose inputs and outputs are provided for custom system design. The number of inputs and outputs varies depending on 82801E C-ICH configuration.

System Management Bus (SMBus)

The 82801E C-ICH contains an SMBus Host interface that allows the processor to communicate with SMBus slaves. This interface is compatible with most I^2C devices. Special I^2C commands are implemented (e.g., the I^2C Read that allows the 82801E C-ICH to perform block reads of I^2C devices).

The 82801E C-ICH's SMBus host controller provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). The host controller supports seven SMBus interface command protocols for communicating with SMBus slave devices (see System Management Bus Specifications, Rev 1.0): Quick Command, Send Byte, Receive Byte, Write Byte/Word, Read Byte/Word, Process Call, and Block Read/Write.

Manageability

The 82801E C-ICH integrates several functions designed to manage the system and lower the total cost of ownership (TCO) of the system. These system management functions are designed to report errors, diagnose the system, and recover from system lockups without the aid of an external microcontroller.

- **TCO Timer.** The 82801E C-ICH's integrated programmable TCO Timer is used to detect system locks. The first expiration of the timer generates an SMI# that the system can use to recover from a software lock. The second expiration of the timer causes a system reset to recover from a hardware lock.
- **Processor Present Indicator.** The 82801E C-ICH looks for the processor to fetch the first instruction after reset. If the processor does not fetch the first instruction, the 82801E C-ICH will reboot the system at the safe-mode frequency multiplier.
- **ECC Error Reporting.** When detecting an ECC error, the host controller has the ability to send one of several messages to the 82801E C-ICH. The host controller can instruct the 82801E C-ICH to generate either an SMI#, NMI, SERR#, or TCO interrupt.
- **Function Disable.** The 82801E C-ICH provides the ability to disable the following functions: IDE, USB, or SMBus. Once disabled, these functions no longer decode I/O, memory, or PCI configuration space. Also, no interrupts or power management events are generated from the disable functions.
- **Intruder Detect.** The 82801E C-ICH provides an input signal (INTRUDER#) that can be attached to a switch that is activated by the system case being opened. The 82801E C-ICH can be programmed to generate an SMI# or TCO interrupt due to an active INTRUDER# signal.
- **SMBus.** The 82801E C-ICH integrates an SMBus controller that provides an interface to manage peripherals (e.g., serial presence detection (SPD) and thermal sensors).
- **Alert-On-LAN*.** The 82801E C-ICH supports Alert-On-LAN* and Alert-On-LAN* 2. In response to a TCO event (intruder detect, thermal event, processor not booting) the 82801E C-ICH sends a message over the SMBus. A LAN controller can decode this SMBus message and send a message over the network to alert the network manager.

Serial I/O Unit

The SIU is similar to currently available Super I/O controllers. It is connected externally via the LPC bus. It consists of two UARTS, a Serial Interrupt Controller and the LPC interface.

Functional Description 3

3.1 Hub Interface to PCI Bridge (D30:F0)

The hub interface to PCI Bridge resides in PCI Device 30, Function 0 on bus #0. This portion of the 82801E C-ICH implements the buffering and control logic between PCI and the hub interface. The arbitration for the PCI bus is handled by this PCI device. The PCI decoder in this device must decode the ranges for the hub interface. All register contents will be lost when core well power is removed.

3.1.1 PCI Bus Interface

The 82801E C-ICH PCI interface provides a 33 MHz, Rev. 2.2 compliant implementation. All PCI signals are 5V tolerant. The 82801E C-ICH integrates a PCI arbiter that supports up to four external PCI bus masters in addition to the internal 82801E C-ICH requests.

Note that most transactions targeted to the 82801E C-ICH will first appear on the external PCI bus before being claimed back by the 82801E C-ICH. The exceptions are I/O cycles involving USB, and IDE. These transactions will complete over the hub interface without appearing on the external PCI bus. *Configuration cycles* targeting USB, or IDE *will* appear on the PCI bus. If the 82801E C-ICH is programmed for positive decode, the 82801E C-ICH will claim the cycles appearing on the external PCI bus in medium decode time. If the 82801E C-ICH is programmed for subtractive decode, the 82801E C-ICH will claim these cycles in subtractive time. If the 82801E C-ICH is programmed for subtractive decode, these cycles can be claimed by another positive decode agent on the PCI bus. This architecture enables the ability to boot off of a PCI card that positively decodes the boot cycles. To boot off a PCI card it is necessary to keep the 82801E C-ICH in subtractive decode mode. When booting off a PCI card, the BOOT_STS bit (bit 2, TCO2 Status Register) will be set.

Note: The 82801E C-ICH's IDE, and USB Controllers cannot access PCI address ranges.

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

PCI configuration write cycles, initiated by the processor, with the following characteristics will be converted to a Special Cycle with the Shutdown message type.

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

If the processor issues a locked cycle to a resource that is too slow (e.g., PCI), the 82801E C-ICH will not allow upstream requests to be performed until the cycle completes. This may be critical for isochronous buses that assume certain timing for their data flow (e.g., 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 82801E C-ICH is not able to issue a request for the next data. Snoop cycles are not permitted while the processor 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 microseconds at most). If a system has a large number of locked cycles and some that are very long, the system will experience underruns and overruns. The unit most likely to have problems is the USB controller. Other units could get underruns/overruns, but it is much less likely. The IDE controller (due to its stalling capability on the cable) should not get any underruns or overruns.

The 82801E C-ICH was designed to provide high-performance support to PCI peripherals through use of its data prefetch capabilities. If a PCI master is burst reading and is disconnected by the 82801E C-ICH to pre-fetch the requested cache line, the 82801E C-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 82801E C-ICH, the 82801E C-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 reduces the discard timer from 1024 PCI clocks (32 µs) to 128 clocks (4 µs) and improves latency for masters that exhibit this behavior.

3.1.2 PCI-to-PCI Bridge Model

From a software perspective, the 82801E C-ICH contains a PCI-to-PCI bridge. This bridge connects the hub interface to the PCI bus. By using the PCI-to-PCI bridge software model, the 82801E C-ICH can have its decode ranges programmed by existing plug-and-play software so that PCI ranges do not conflict with AGP and graphics aperture ranges in the Host controller.

3.1.3 IDSEL to Device Number Mapping

When addressing devices on the external PCI bus (with the PCI slots), the 82801E C-ICH asserts one address signal as an IDSEL. When accessing device 0, the 82801E C-ICH asserts AD16. When accessing Device 1, the 82801E C-ICH asserts AD17. This mapping continues up to device 15 where the 82801E C-ICH asserts AD31. Note that the 82801E C-ICH's internal functions (IDE, USB, and PCI Bridge) are enumerated like they are on a separate PCI bus (the hub interface) from the external PCI bus. The integrated LAN Controllers are Device 8 and Device 9 on the 82801E C-ICH's PCI bus and use AD24 and AD25 for IDSEL, respectively.

3.1.4 SERR# Functionality

There are several internal and external sources that can cause SERR#. The 82801E C-ICH can be programmed to cause an NMI when it detects that an SERR# condition has occurred. The NMI can also be routed to cause an SMI# instead. Note that the 82801E C-ICH does not drive the external PCI bus SERR# signal active onto the PCI bus. The external SERR# signal is an input into the 82801E C-ICH. It is driven only by external PCI devices. The conceptual logic diagrams in Figure 3 and Figure 4 illustrate all sources of SERR#, along with their respective enable and status bits. Figure 5 shows how the 82801E C-ICH error reporting logic is configured for NMI# generation.

Figure 3. Primary Device Status Register Error Reporting Logic

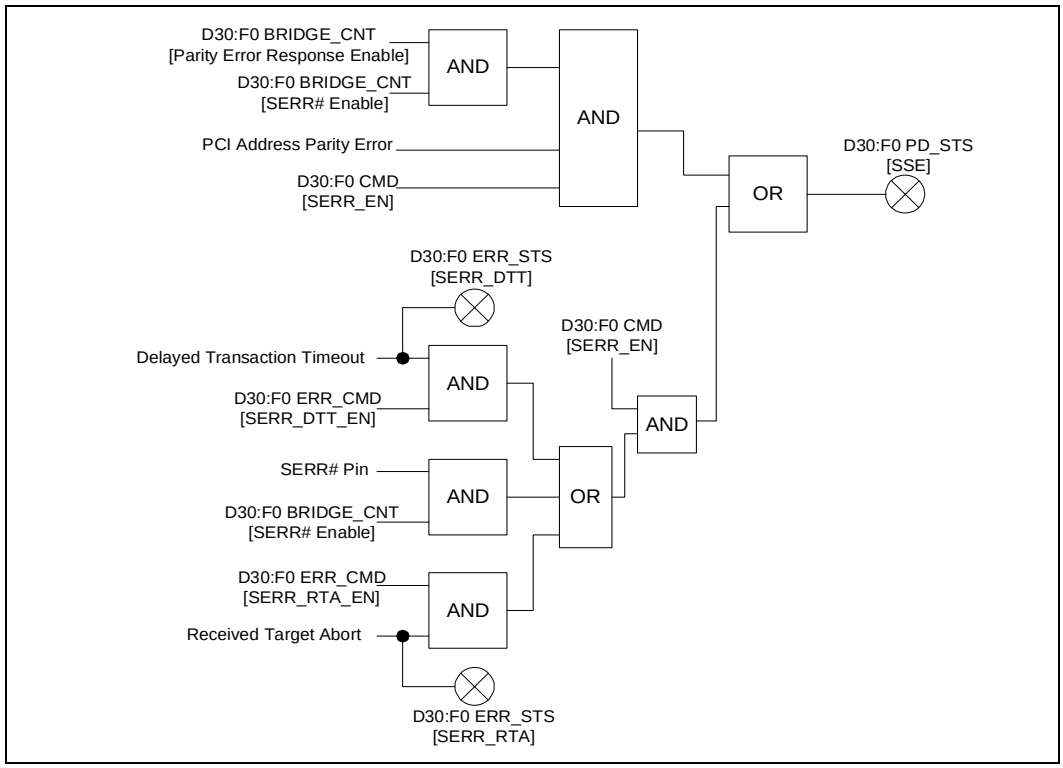


Figure 4. Secondary Device Status Register Error Reporting Logic

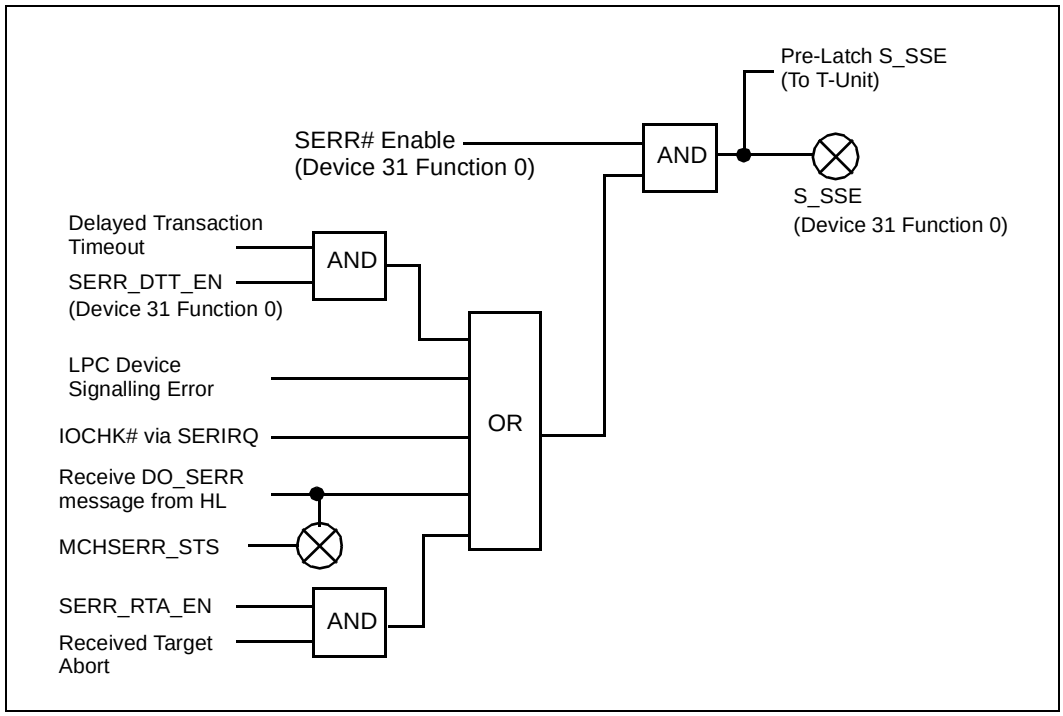
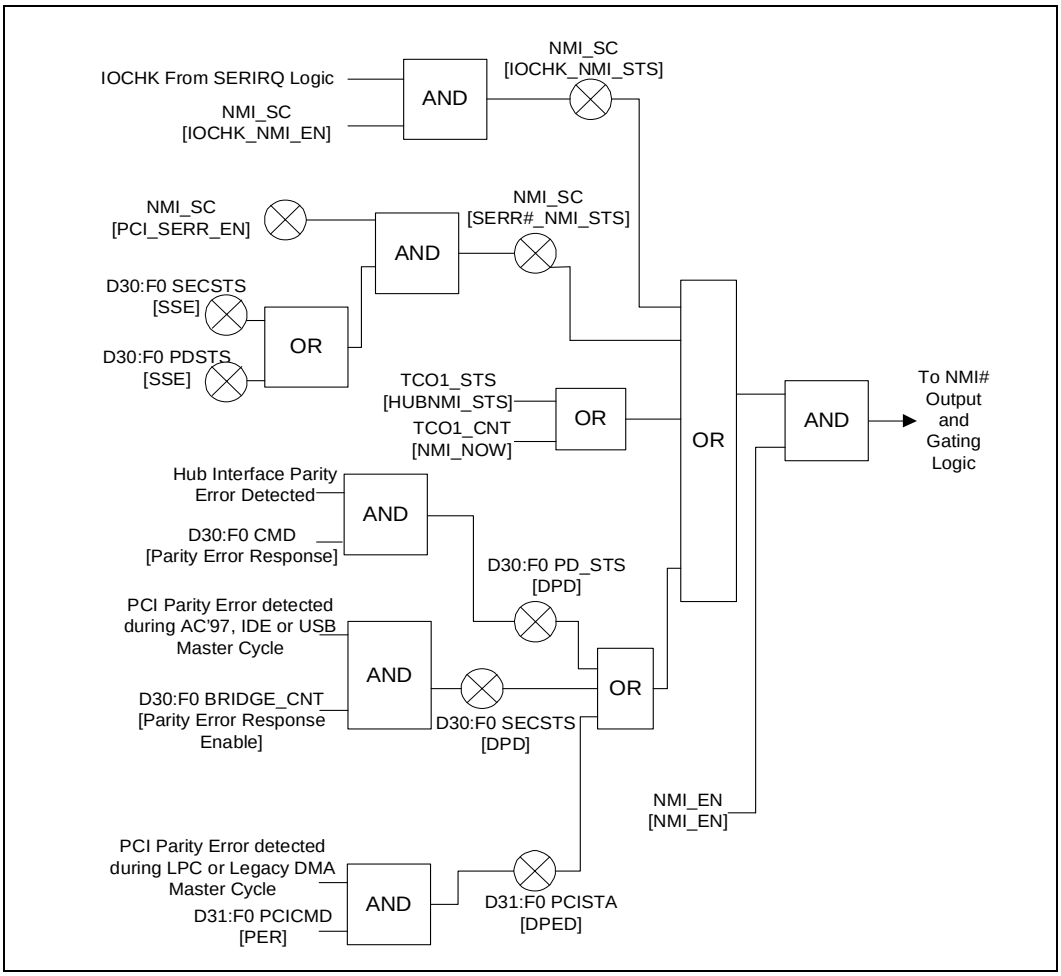


Figure 5. NMI# Generation Logic



Note: D30:F0; 1EH:[14] (SEC_STS:[SSE]) is only set when SERR# is detected asserted.

3.1.5 Parity Error Detection

The 82801E C-ICH can detect and report different parity errors in the system. The 82801E C-ICH can be programmed to cause an NMI (or SMI# if NMI is routed to SMI#) based on detecting a parity error. The conceptual logic diagram in Figure 5 details all the parity errors that the 82801E C-ICH can detect, along with their respective enable bits, status bits, and the results.

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

Note: The 82801E C-ICH does not generate a valid Hub Interface Parity message when Parity Error Response is disabled for the Hub/PCI Bridge function. The system may hang if Hub Interface Parity Error Response is enabled in the MCH when it is disabled in the 82801E C-ICH. This issue only affects platforms in which the MCH supports Hub Interface Parity checking. Software must

not disable Hub Interface Parity Error Response in the ICH while it is enabled in the MCH. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.

3.1.6 Standard PCI Bus Configuration Mechanism

The PCI Bus defines a slot based "configuration space" that allows each device to contain up to eight functions, with each function containing up to 256 8-bit configuration registers. The PCI specification defines two bus cycles that access the PCI configuration space: Configuration Read and Configuration Write. Memory and I/O spaces are supported directly by the processor. Configuration space is supported by a mapping mechanism implemented within the 82801E C-ICH. The PCI specification defines two mechanisms to access configuration space (Mechanism #1 and Mechanism #2). The 82801E C-ICH only supports Mechanism #1.

Configuration cycles for PCI Bus #0 devices #2 through #31 and for PCI Bus numbers greater than 0 will be sent towards the 82801E C-ICH from the host controller. The 82801E C-ICH compares the non-zero Bus Number with the Secondary Bus Number and Subordinate Bus number registers of its P2P bridge to determine if the configuration cycle is meant for Primary PCI or a downstream PCI bus.

Type 0 to Type 0 Forwarding

When a Type 0 configuration cycle is received on the hub interface, the 82801E C-ICH forwards these cycles to PCI and then reclaims them. The 82801E C-ICH uses address bits AD[15:14] to communicate the 82801E C-ICH device numbers in Type 0 configuration cycles. If the Type 0 cycle on the hub interface specifies any device number other than 30 or 31, the 82801E C-ICH will not set any address bits in the range AD[31:11] during the corresponding transaction on PCI. Table 4 shows the device number translation.

Table 4. Type 0 Configuration Cycle Device Number Translation

Device # In Hub Interface Type 0 Cycle	AD[31:11] During Address Phase of Type 0 Cycle on PCI
0 through 29	0000000000000000_00000b
30	0000000000000000_01000b
31	0000000000000000_10000b

The 82801E C-ICH logic generates single DWord configuration read and write cycles on the PCI bus. The 82801E C-ICH generates a Type 0 configuration cycle for configurations to the bus number matching the PCI bus. Type 1 configuration cycles are converted to Type 0 cycles in this case. If the cycle is targeting a device behind an external bridge, the 82801E C-ICH runs a Type 1 cycle on the PCI bus.

Type 1 to Type 0 Conversion

When the bus number for the Type 1 configuration cycle matches the PCI (Secondary) bus number, the 82801E C-ICH converts the address as follows:

- For device numbers 0 through 15, only one bit of the PCI address [31:16] is set. If the device number is 0, AD[16] is set; if the device number is 1, AD[17] is set; etc.
- The 82801E C-ICH always drives 0s on bits AD[15:11] when converting Type 1 configurations cycles to Type 0 configuration cycles on PCI.

- Address bits [10:1] are also passed unchanged to PCI.
- Address bit [0] is changed to 0.

3.1.7 PCI Dual Address Cycle (DAC) Support

The 82801E C-ICH supports Dual Address Cycle (DAC) format on PCI for cycles from PCI initiators to main memory. This allows PCI masters to generate an address up to 44 bits. The size of the actual supported memory space is determined by the memory controller and the processor.

The DAC mode is only supported for PCI adapters. It is not supported for any of the internal PCI masters (IDE, LAN, USB, SIU, 8237 DMA, etc.). 82801E C-ICH does not support DAC for processor-initiated cycles.

When a PCI master wants to initiate a cycle with an address above 4 Gbytes, it follows the following rules (See PCI 2.2 Specification, section 3.9 for more details):

1. On the first clock of the cycle (when FRAME# is first active), the peripheral uses the DAC encoding on the C/BE# signals. This unique encoding is 1101.
2. Also during the first clock, the peripheral drives the AD[31:0] signals with the low address.
3. On the second clock, the peripheral drives AD[31:0] with the high address. The address is right justified: A[43:32] appear on AD[12:0]. The value of AD[31:13] is expected to be 0, however the 82801E C-ICH will ignore these bits. C/BE# indicate the bus command type (Memory Read, Memory Write, etc.)
4. The rest of the cycle proceeds normally.

3.2 LAN Controllers (B1:D8/D9:F0)

The 82801E C-ICH integrates the digital functions associated with two 82559 LAN Controllers. This chapter covers the features of that logic, as well as the differences between the 82801E C-ICH implementation and the LAN Connect implementation. In this document, the LAN function within the 82801E C-ICH will be called the LAN Controller.

The functionality of each of the integrated LAN Controller devices is identical (two complete cores are integrated into the 82801E C-ICH architecture) so only one is described below. A few interconnection differences between the devices and the remaining 82801E C-ICH core logic and other related variances to support two devices instead of one are noted.

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

The 82801E C-ICH integrated LAN Controllers can operate in full duplex mode or half duplex mode. In full duplex mode the LAN Controllers adhere to the IEEE 802.3x Flow Control specification. Half duplex performance is enhanced by a proprietary collision reduction mechanism.

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

From a software perspective, the integrated LAN Controllers appear to reside on the secondary side of the 82801E C-ICH's virtual PCI-to-PCI Bridge (see Section 3.1.2). This is typically Bus 1; it may be assigned a different number depending on system configuration.

Feature Summary

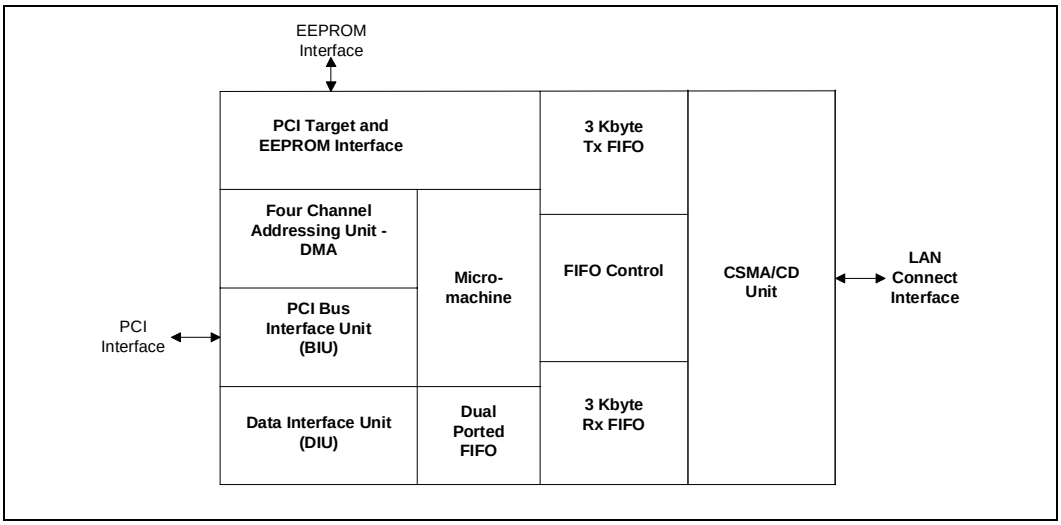
- Compliance with Advanced Configuration standards and Power Interface and PCI Power Management standards
- Support for wake-up on interesting packets and link status change
- Support for remote power-up using Wake on LAN* (WOL) technology
- Support of Wired for Management (WfM) Rev 2.0
- Backward compatible software with 82557, 82558, and 82559
- TCP/UDP checksum offload capabilities
- Support for Intel's Adaptive Technology
- External LAN Connect using new Jordan I/F

Note: **The 82801E C-ICH does not support low power or sleep states**.

3.2.1 LAN Controller Architectural Overview

Figure 6 is a high-level block diagram of the 82801E C-ICH integrated LAN Controller. It is divided into four main subsystems: a Parallel subsystem, a FIFO subsystem, and the Carrier-Sense Multiple Access with Collision Detect (CSMA/CD) unit.

Figure 6. Integrated LAN Controller Block Diagram



3.2.1.1 Parallel Subsystem Overview

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

The PCI bus master interface provides a complete interface to the PCI bus and is compliant with the PCI Bus Specification, Revision 2.2. The LAN Controller provides 32 bits of addressing and data, as well as the complete control interface to operate on the PCI bus. As a PCI target, it follows the PCI configuration format that allows all accesses to the LAN Controller to be automatically mapped into free memory and I/O space upon initialization of a PCI system. For processing of transmit and receive frames, the integrated LAN Controller operates as a master on the PCI bus, initiating zero wait state transfers for accessing these data parameters.

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

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

The LAN Controller contains an interface to an external serial EEPROM. The EEPROM is used to store relevant information for a LAN connection such as node address, as well as board manufacturing and configuration information. Both read and write accesses to the EEPROM are supported by the LAN Controller. Information on the EEPROM interface is detailed in "Serial EEPROM Interface" on page 48.

3.2.1.2 FIFO Subsystem Overview

The 82801E C-ICH LAN Controller FIFO subsystem consists of a 3 Kbytes transmit FIFO and 3 Kbytes receive FIFO. Each FIFO is unidirectional and independent of the other. The FIFO subsystem serves as the interface between the LAN Controller parallel side and the serial CSMA/CD unit. It provides a temporary buffer storage area for frames as they are either being received by or transmitted by the LAN Controller. This improves performance:

- Transmit frames can be queued within the transmit FIFO. This allows back-to-back transmission within the minimum Interframe Spacing (IFS).
- The storage area in the FIFO allows the LAN Controller to withstand long PCI bus latencies without losing incoming data or corrupting outgoing data.
- The 82801E C-ICH LAN Controller's transmit FIFO threshold allows the transmit start threshold to be tuned to eliminate underruns while concurrent transmits are being performed.

- The FIFO subsection allows extended PCI zero wait state burst accesses to or from the LAN Controller for both transmit and receive frames since the transfer is to the FIFO storage area rather than directly to the serial link.
- Transmissions that result in errors (collision detection or data underrun) are retransmitted directly from the LAN Controller's FIFO. This increases performance and eliminates the need to re-access this data from the host system.
- Incoming runt receive frames (in other words, frames that are less than the legal minimum frame size) can be discarded automatically by the LAN Controller without transferring this faulty data to the host system.

3.2.1.3 Serial CSMA/CD Unit Overview

The CSMA/CD unit of the 82801E C-ICH LAN Controller allows it to be connected to the 82562ET/EM 10/100 Mbps Ethernet LAN Connect components or the 82562EH 1 Mbps HomePNA*-compliant LAN Connect component. The CSMA/CD unit performs all of the functions of the 802.3 protocol such as frame formatting, frame stripping, collision handling, deferral to link traffic, etc. The CSMA/CD unit can also be placed in a full duplex mode; this enables simultaneous transmission and reception of frames.

3.2.2 LAN Controller PCI Bus Interface

As a Fast Ethernet Controller, the role of the 82801E C-ICH integrated LAN Controller is to access transmitted data or to deposit received data. The LAN Controller, as a bus master device, initiates memory cycles via the PCI bus to fetch or deposit the required data.

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

3.2.2.1 Bus Slave Operation

The 82801E C-ICH integrated LAN Controller serves as a target device in one of the following cases:

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

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

Control/Status Register (CSR) Accesses

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

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

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

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

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

Retry Premature Accesses

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

Error Handling

Data Parity Errors: The LAN Controller checks for data parity errors while it is the target of the transaction. If an error was detected, the LAN Controller always sets the Detected Parity Error bit in the PCI Configuration Status register, bit 15. The LAN Controller also asserts PERR#, if the Parity Error Response bit is set (PCI Configuration Command register, bit 6). The LAN Controller does not attempt to terminate a cycle in which a parity error was detected. This gives the initiator the option of recovery.

Target-Disconnect: The LAN Controller terminates a cycle in the following cases:

- After accesses to its CSR

- After accesses to the configuration space

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

After detecting a system error, the LAN Controller will claim the cycle if it was the target of the transaction and continue the transaction as if the address was correct.

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

3.2.2.2 Bus Master Operation

As a PCI Bus Master, the 82801E C-ICH integrated LAN Controller initiates memory cycles to fetch data for transmission or to deposit received data, and for accessing the memory resident control structures. The LAN Controller performs zero wait state burst read and write cycles to the host main memory. For bus master cycles, the LAN Controller is the initiator and the host main memory (or the PCI host bridge, depending on the configuration of the system) is the target.

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

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

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

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

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

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

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

Memory Write and Invalidate

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

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

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

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

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

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

Read Align

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

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

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

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

Note: This feature is not recommended for use in non-cache line oriented systems since it may cause shorter bursts and lower performance.

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

Note: The LAN Controller reads all control data structures (including Receive Buffer Descriptors) from the first DWord (even if it is not required) to maintain cache line alignment.

Error Handling

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

3.2.2.3 PCI Reset Signal

The PCIRST# signal may be activated by a CF9h reset.

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

3.2.2.4 Wake-up Events

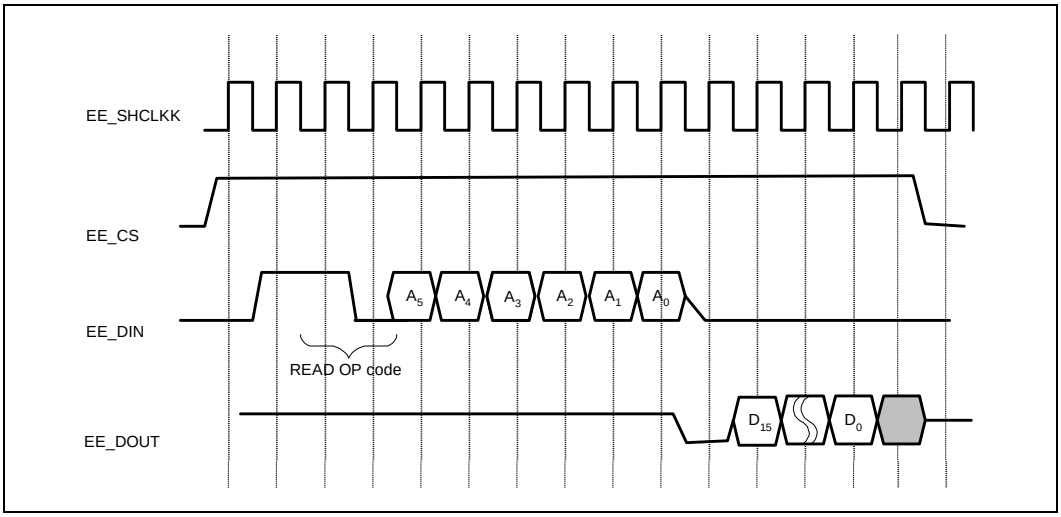
The 82801E C-ICH does not support sleep states and suspend mode; it is an “on” or “off” device. Wake-up events and incoming packets are still processed, since the 82801E C-ICH is always “awake” while the device is powered on.

3.2.3 Serial EEPROM Interface

The serial EEPROM stores configuration data for the 82801E C-ICH integrated LAN Controller and is a serial in/serial out device. The LAN Controller supports a 64 word size or 256 register size EEPROM and automatically detects the EEPROM’s size. A 256 word EEPROM device is required for a Cardbus system and contains the CIS information. The EEPROM should operate at a frequency of at least 1 MHz.

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

Figure 7. 64-Word EEPROM Read Instruction Waveform



The LAN Controller performs an automatic read of seven words (0h, 1h, 2h, Ah, Bh, Ch and Dh) of the EEPROM after the deassertion of Reset. The 82801E C-ICH integrated LAN Controller’s EEPROM format is shown below in Table 5. For additional information, refer to Application Note AP-409, “*I/O Controller Hub EEPROM Map and Programming Information*”.

Table 5. EEPROM Address Map

Word	HIgh Byte (Bits 15:8)		Low Byte (Bits 7:0)	Used by
00h	Ethernet Individual Address Byte 2		Ethernet Individual Address Byte 1	Hardware
01h	Ethernet Individual Address Byte 4		Ethernet Individual Address Byte 3	Hardware
02h	Ethernet Individual Address Byte 6		Ethernet Individual Address Byte 5	Hardware
03h	Compatibility Byte 1		Compatibility Byte 0	Intel® driver
04h	Reserved.			
05h	Controller Type (02 for 82801E C-ICH)		Connector Type	Intel driver
06h	PHY Device Record			
07h	Reserved.			
08h	PWA Number Byte 4		PWA Number Byte 3	Factory
09h	PWA Number Byte 2		PWA Number Byte 1	Factory
0Ah	EEPROM ID			Hardware
0Bh	Subsystem ID			Hardware
0Ch	Subsystem Vendor ID			Hardware
0Dh	0000b	Heartbeat Packet Pointer	SMB Address Field	Alert on LAN* driver or hardware
0Eh–2Fh	Reserved.			
30h	Reserved for Boot Agent ROM Configuration (PXE and RPL version)			Firmware
31h	Reserved for Boot Agent ROM Configuration (PXE and RPL version)			Firmware
32h	Reserved for Boot Agent ROM Configuration (PXE and RPL version)			Firmware
33h–3Ah	Reserved.			
3Bh	Reserved for Boot ROM Configuration (PXE only)			Firmware
3Ch–3Fh	Reserved.			
40h–FAh	Alert on LAN alert packet structure			Alert on LAN driver
FFh	Checksum			Driver

Words 00h through 02h are used by the hardware and are common to all controllers.

3.2.4 CSMA/CD Unit

The 82801E C-ICH integrated LAN Controller CSMA/CD unit implements both the IEEE 802.3 Ethernet 10 Mbps and IEEE 802.3u Fast Ethernet 100 Mbps standards. It also supports the 1 Mbps Home Phoneline Networking Alliance (HomePNA*) specification effort. It performs all the CSMA/CD protocol functions, such as transmission, reception, collision handling, etc. The LAN Controller CSMA/CD unit interfaces to the 82562ET/EM 10/100 Mbps Ethernet or to the 82562EH 1 Mbps HomePNA*-compliant LAN Connect component through the 82801E C-ICH's LAN Connect interface signals.

Full Duplex

The LAN Controller operates in either half duplex mode or in full duplex mode. When operating in full duplex mode, the LAN Controller can transmit and receive frames simultaneously. Transmission starts regardless of the state of the internal receive path. Reception starts when the LAN Connect component detects a valid frame on its receive differential pair. When in Full Duplex mode, the 82801E C-ICH integrated LAN Controller also supports the IEEE 802.3x flow control standard.

For proper operation, both the LAN Controller CSMA/CD module and the discrete LAN Connect component must be set to the same duplex mode. The CSMA duplex mode is set by the LAN Controller Configure command or is forced by automatically tracking the mode in the LAN Connect component. Following reset, the CSMA will default to automatically track the LAN Connect component duplex mode.

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

Flow Control

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

Flow control is optional in full duplex mode. It is selected through software configuration. There are three flow control mode options: frame-based transmit flow control, frame-based receive flow control, and none.

Address Filtering Modifications

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

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

VLAN Support

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

3.2.5 Media Management Interface

The management interface allows the processor to control the LAN Connect component via a control register in the 82801E C-ICH integrated LAN Controller. This allows the software driver to place the LAN Connect in specific modes such as full duplex, loopback, etc., without the need for specific hardware pins to select the desired mode. This structure allows the LAN Controller to query the LAN Connect component for status of the link. This register is the MDI Control Register and resides at offset 10h in the LAN Controller CSR. The MDI registers reside within the LAN Connect component, and are described in detail in the LAN Connect component's datasheet. The processor writes commands to this register and the LAN Controller reads or writes the control/status parameters to the LAN Connect component through the MDI register.

3.2.6 TCO Functionality

The 82801E C-ICH integrated LAN controller supports management communication to reduce Total Cost of Ownership (TCO). It has a System Management Bus (SMB) on which the LAN controller is a slave device. The SMB is used as an interface between the LAN controller and the integrated host controller. An EEPROM of 256 words is required to support the heartbeat command.

Receive Functionality

The LAN controller transfers TCO packets to the host the same way as any other packet. These packets include a new status indication bit in the Receive Frame Descriptor (RFD) status register and have a specific port number that indicates TCO packet recognition.

Transmit Functionality

The 82801E C-ICH integrated LAN controller supports the Heartbeat (HB) transmission command from the SMB interface. The send HB packet command includes a system health status issued by the integrated system controller. The LAN controller computes a matched checksum and CRC, and transmits the HB packet from its serial EEPROM. The HB packet size and structure are not limited as long as it fits within the EEPROM size. In this case, the EEPROM size is 256 words to enable the storage of the HB packet (the first 64 words are used for driver specific data).

Note: On the SMB, the send heartbeat packet command is not normally used in the D0 power state. The one exception in which it is used in the D0 state is when the system is hung. In normal operating mode, the heartbeat packets are transmitted through the 82801E C-ICH integrated LAN controller software similar to other packets.

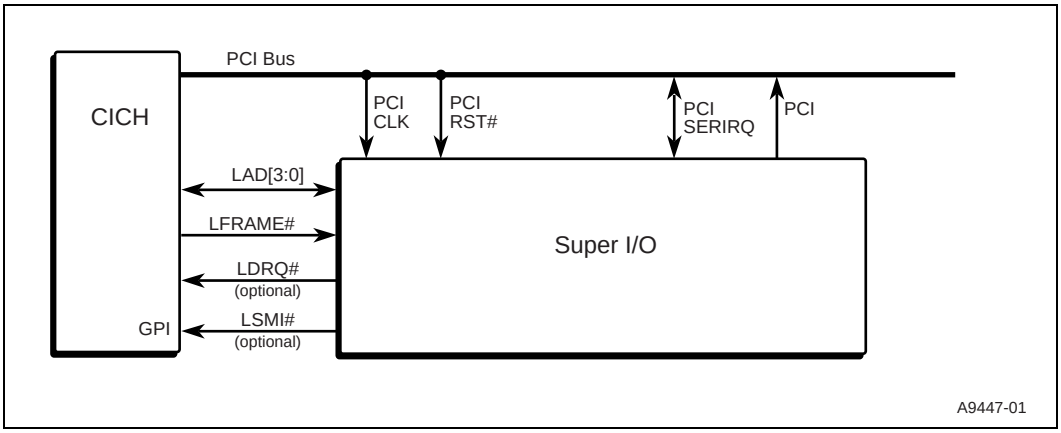
3.3 LPC Bridge (with System and Management Functions) (D31:F0)

The LPC Bridge function of the 82801E C-ICH resides in PCI Device 31:Function 0. In addition to the LPC bridge function, D31:F0 contains other functional units; these functions include DMA, Interrupt Controllers, Timers, Power Management, System Management, GPIO, and RTC. In this chapter, registers and functions associated with other functional units (power management, GPIO, USB, IDE, etc.) are described in their respective sections.

3.3.1 LPC Interface

The 82801E C-ICH implements an LPC interface as described in the LPC 1.0 specification. The LPC interface to the 82801E C-ICH is shown in Figure 8. Note that the 82801E C-ICH implements all of the signals that are shown as optional, but peripherals are not required to do so.

Figure 8. LPC Interface Diagram



3.3.1.1 LPC Cycle Types

The 82801E C-ICH implements all of the cycle types described in the LPC Interface 1.0 specification. Table 6 shows the cycle types supported by the 82801E C-ICH.

Table 6. LPC Cycle Types Supported

Cycle Type	Comment
Memory Read	Single: 1 byte only
Memory Write	Single: 1 byte only
I/O Read	1 byte only. 82801E C-ICH breaks up 16 and 32-bit processor cycles into multiple 8-bit transfers. See Note 1 below.
I/O Write	1 byte only. 82801E C-ICH breaks up 16 and 32-bit processor cycles into multiple 8-bit transfers. See Note 1 below.
DMA Read	Can be 1 or 2 bytes
DMA Write	Can be 1 or 2 bytes
Bus Master Read	Can be 1, 2, or 4 bytes. (See Note 2 below)
Bus Master Write	Can be 1, 2, or 4 bytes. (See Note 2 below)

NOTES:
1. For memory cycles below 16 Mbyte which do not target enabled FWH ranges, the 82801E C-ICH will perform standard LPC memory cycles. It will only attempt 8-bit transfers. If the cycle appears on PCI as a 16-bit transfer, it will appear as two consecutive 8-bit transfers on the LPC bus. Likewise, if the cycle appears as a 32-bit transfer on the PCI bus, it will appear as four consecutive 8-bit transfers on the LPC bus. If the cycle is not claimed by any peripheral, it will be subsequently aborted, and the 82801E C-ICH will return a value of all 1s to the processor. This is done to maintain compatibility with ISA memory cycles where pull-up resistors would keep the bus high if no device responds.
2. Bus Master Read or Write cycles must be naturally aligned. For example, a 1-byte transfer can be to any address. However, the 2-byte transfer must be word aligned (i.e., with an address where A0=0). A DWord transfer must be DWord aligned (i.e., with an address where A1and A0 are both 0).

3.3.1.2 Start Field Definition

Table 7. Start Field Bit Definitions

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

NOTE: All other encodings are Reserved.

3.3.1.3 Cycle Type/Direction (CYCTYPE + DIR)

The 82801E C-ICH always drives bit 0 of this field to 0. Peripherals running bus master cycles must also drive bit 0 to 0. Table 8 shows the valid bit encodings.

Table 8. Cycle Type Bit Definitions

Bits[3:2]	Bit[1]	Definition
00	0	I/O Read
00	1	I/O Write
01	0	Memory Read
01	1	Memory Write
10	0	DMA Read
10	1	DMA Write
11	x	Reserved. If a peripheral performing a bus master cycle generates this value, the 82801E C-ICH will abort the cycle.

3.3.1.4 Size

Bits[3:2] are reserved. The 82801E C-ICH always drives them to 00. Peripherals running bus master cycles are also supposed to drive 00 for bits 3:2; however, the 82801E C-ICH ignores those bits. Table 9 shows the bit encodings for Bits[1:0].

Table 9. Transfer Size Bit Definition

Bits[1:0]	Size
00	8 bit transfer (1 byte)
01	16-bit transfer (2 bytes)
10	Reserved. The 82801E C-ICH never drives this combination. If a peripheral running a bus master cycle drives this combination, the 82801E C-ICH may abort the transfer.
11	32 bit transfer (4 bytes)

3.3.1.5 SYNC

Valid values for the SYNC field are listed in Table 10.

Table 10. SYNC Bit Definition

Bits[3:0]	Indication
0000	**Ready:** SYNC achieved with no error. For DMA transfers, this also indicates DMA request deassertion and no more transfers desired for that channel.
0101	**Short Wait:** Part indicating wait states. For bus master cycles, the 82801E C-ICH does not use this encoding. It will instead use the Long Wait encoding (see next encoding below).
0110	**Long Wait:** Part indicating wait states; many wait states will be added. This encoding is driven by the 82801E C-ICH for bus master cycles, rather than the Short Wait (0101).
1001	**Ready More (Used only by peripheral for DMA cycle):** SYNC achieved with no error and more DMA transfers desired to continue after this transfer. This value is valid only on DMA transfers and is not allowed for any other type of cycle.
1010	**Error:** Sync achieved with error. This is generally used to replace the SERR# or IOCHK# signal on the PCI/ISA bus. It indicates that the data is to be transferred, but there is a serious error in this transfer. For DMA transfers, this not only indicates an error, but also indicates DMA request deassertion and no more transfers desired for that channel.

NOTE: All other combinations are Reserved.

3.3.1.6 SYNC Time-out

There are several error cases that can occur on the LPC interface. Table 11 indicates the failing case and the 82801E C-ICH response.

Table 11. Intel® 82801E C-ICH Response to Sync Failures

Possible Sync Failure	82801E C-ICH Response
82801E C-ICH starts a Memory, I/O, or DMA cycle, but no device drives a valid SYNC after 4 consecutive clocks. This could occur if the processor tries to access an I/O location to which no device is mapped.	82801E C-ICH aborts the cycle after the fourth clock.
82801E C-ICH drives a Memory, I/O, or DMA cycle, and a peripheral drives more than 8 consecutive valid SYNC patterns to insert wait states using the Short ('0101b') encoding for SYNC. This could occur if the peripheral is not operating properly.	Continues waiting
82801E C-ICH starts a Memory, I/O, or DMA cycle, and a peripheral drives an invalid SYNC pattern. This could occur if the peripheral is not operating properly or if there is excessive noise on the LPC interface.	82801E C-ICH aborts the cycle when the invalid Sync is recognized.

NOTE: There may be other peripheral failure conditions; however, these are not handled by the 82801E C-ICH.

3.3.1.7 SYNC Error Indication

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

If the 82801E C-ICH was reading data from a peripheral, data will still be transferred in the next two nibbles. This data may be invalid; however, it must be transferred by the peripheral. If the 82801E C-ICH was writing data to the peripheral, the data had already been transferred.

In the case of multiple byte cycles (e.g., for memory and DMA cycles) an error SYNC terminates the cycle. Therefore, if the 82801E C-ICH is transferring 4 bytes from a device and the device returns the error SYNC in the first byte, the other three bytes are not transferred.

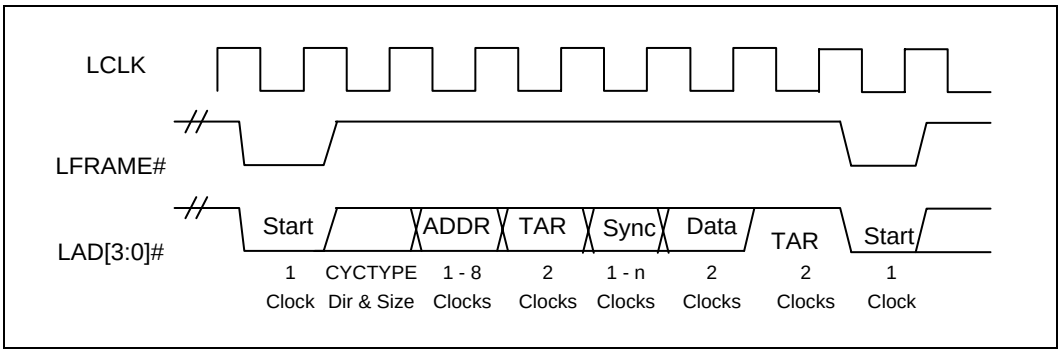
When recognizing the SYNC field indicating an error, the 82801E C-ICH treats this the same as if IOCHK# goes active on the ISA bus.

3.3.1.8 LFRAME# Usage

Start of Cycle

For Memory, I/O, and DMA cycles, the 82801E C-ICH asserts LFRAME# for 1 clock at the beginning of the cycle (Figure 9). During that clock, the 82801E C-ICH drives LAD[3:0] with the proper START field.

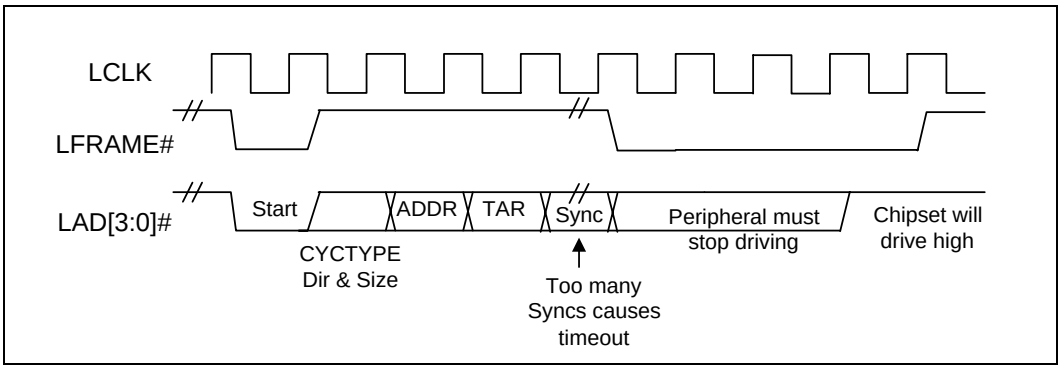
Figure 9. Typical Timing for LFRAME#



Abort Mechanism

When performing an abort, the 82801E C-ICH drives LFRAME# active for four consecutive clocks. On the fourth clock, the 82801E C-ICH drives LAD[3:0] to '1111b'.

Figure 10. Abort Mechanism



The 82801E C-ICH performs an abort for the following cases (possible failure cases):

- 82801E C-ICH starts a Memory, I/O, or DMA cycle and no device drives a valid SYNC after four consecutive clocks.
- 82801E C-ICH starts a Memory, I/O, or DMA cycle, and the peripheral drives an invalid SYNC pattern.
- A peripheral drives an illegal address when performing bus master cycles.
- A peripheral drives an invalid value.

3.3.1.9 I/O Cycles

For I/O cycles targeting registers specified in the 82801E C-ICH's decode ranges, the 82801E C-ICH performs I/O cycles as defined in the LPC specification. These are 8-bit transfers. If the processor attempts a 16-bit or 32-bit transfer, the 82801E C-ICH will break the cycle up into multiple 8-bit transfers to consecutive I/O addresses.

Note: If the cycle is not claimed by any peripheral (and subsequently aborted), the 82801E C-ICH returns all 1s (FFh) to the processor. This is to maintain compatibility with ISA I/O cycles where pull-up resistors would keep the bus high if no device responds.

3.3.1.10 Bus Master Cycles

The 82801E C-ICH supports Bus Master cycles and requests (using LDRQ#) as defined in the LPC specification. The 82801E C-ICH has two LDRQ# inputs; thus, 82801E C-ICH supports two separate bus master devices. It uses the associated START fields for Bus Master 0 ('0010b') or Bus Master 1 ('0011b').

Note: The 82801E C-ICH does not support LPC Bus Masters performing I/O cycles. LPC Bus Masters should only perform memory read or memory write cycles.

3.3.1.11 Configuration and 82801E C-ICH Implications

LPC Interface Decoders

To allow the I/O cycles and memory mapped cycles to go to the LPC interface, the 82801E C-ICH includes several decoders. During configuration, the 82801E C-ICH must be programmed with the same decode ranges as the peripheral. The decoders are programmed via the Device 31:Function 0 configuration space.

Note: The 82801E C-ICH can not accept PCI write cycles from PCI-to-PCI bridges or devices with similar characteristics (specifically those with a "Retry Read" feature which is enabled) to an LPC device if there is an outstanding LPC read cycle towards the same PCI device or bridge. These cycles are not part of normal system operation; however, they may be encountered as part of platform validation testing that uses custom test fixtures.

Bus Master Device Mapping and START Fields

Bus Masters must have a unique START field. In the case of the 82801E C-ICH, which supports two LPC bus masters, it will drive 0010b for the START field for grants to bus master #0 (requested via LDRQ[0]#) and 0011b for grants to bus master #1 (requested via LDRQ[1]#.). Thus, no registers are needed to configure the START fields for a particular bus master.

3.3.1.12 BIOS Mapping and START Fields

To reduce decoding logic in the FWH, the 82801E C-ICH will use a unique START field for each EPROM. To do this, the 82801E C-ICH has configuration registers to assign a particular BIOS range to a particular START field.

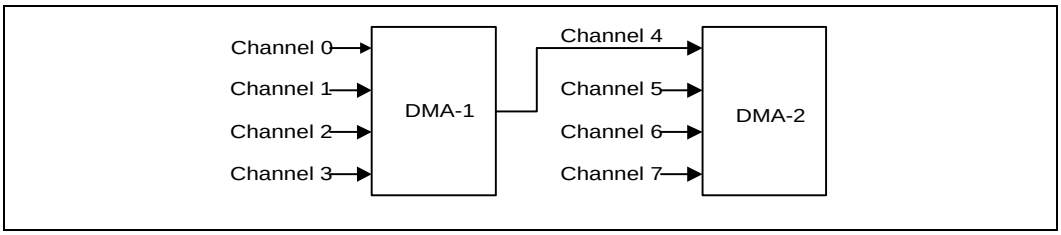
3.4 DMA Operation (D31:F0)

The 82801E C-ICH supports two types of DMA: LPC and PC/PCI. DMA via LPC is similar to ISA DMA. LPC DMA and PC/PCI DMA use the 82801E C-ICH's DMA controller. The DMA controller has registers that are fixed in the lower 64 Kbytes of I/O space.

The DMA controller is configured using registers in the PCI configuration space. These registers allow configuration of individual channels for use by LPC or PC/PCI DMA.

The DMA circuitry incorporates the functionality of two 82C37 DMA controllers with seven independently programmable channels (Figure 11). DMA Controller 1 (DMA-1) corresponds to DMA Channels 0–3 and DMA Controller 2 (DMA-2) corresponds to Channels 5–7. DMA Channel 4 is used to cascade the two controllers and will default to cascade mode in the DMA Channel Mode (DCM) Register. Channel 4 is not available for any other purpose. In addition to accepting requests from DMA slaves, the DMA controller also responds to requests that software initiates. Software may initiate a DMA service request by setting any bit in the DMA Channel Request Register to a 1.

Figure 11. Intel® 82801E C-ICH DMA Controller



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

82801E C-ICH provides 24-bit addressing in compliance with the ISA-Compatible specification. Each channel includes a 16-bit ISA-Compatible Current Register that holds the 16 least-significant bits of the 24-bit address, and an ISA-Compatible Page Register that contains the eight next most significant bits of address.

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

3.4.1 Channel Priority

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

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

Fixed Priority

The initial fixed priority structure is as follows:

High priority.....Low priority
(0, 1, 2, 3) (5, 6, 7)

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

Rotating Priority

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

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

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

3.4.2 Address Compatibility Mode

When the DMA is operating, the addresses do not increment or decrement through the High and Low Page Registers. Therefore, if a 24-bit address is 01FFFFh and increments, the next address will be 010000h, not 020000h. Similarly, if a 24-bit address is 020000h and decrements, the next address will be 02FFFFh, not 01FFFFh. This is compatible with the 82C37 and Page Register implementation used in the PC-AT. This mode is set after CPURST is valid.

3.4.3 Summary of DMA Transfer Sizes

Table 12 lists each of the DMA device transfer sizes. The column labeled “Current Byte/Word Count Register” indicates that the register contents represent either the number of bytes to transfer or the number of 16-bit words to transfer. The column labeled “Current Address Increment/Decrement” indicates the number added to or taken from the Current Address Register after each DMA transfer cycle. The DMA Channel Mode Register determines if the Current Address Register is incremented or decremented.

Table 12. DMA Transfer Size

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

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

The 82801E C-ICH maintains compatibility with the implementation of the DMA in the PC-AT that use the 82C37. The DMA shifts the addresses for transfers to/from a 16-bit device count-by-words. Note that the least significant bit of the Low Page Register is dropped in 16-bit shifted mode. When programming the Current Address Register (when the DMA channel is in this mode), the current address must be programmed to an even address with the address value shifted right by one bit. This address shifting is shown in Table 13.

Table 13. Address Shifting in 16-bit I/O DMA Transfers

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

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

3.4.4 Autoinitialize

By programming a bit in the DMA Channel Mode Register, a channel may be set up as an autoinitialize channel. When a channel undergoes autoinitialization, the original values of the Current Page, Current Address and Current Byte/Word Count Registers are automatically restored from the Base Page, Address, and Byte/Word Count Registers of that channel following TC. The Base Registers are loaded simultaneously with the Current Registers by the processor when the DMA channel is programmed and remain unchanged throughout the DMA service. The mask bit is not set when the channel is in autoinitialize. Following autoinitialize, the channel is ready to perform another DMA service, without processor intervention, as soon as a valid DREQ is detected.

3.4.5 Software Commands

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

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

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

Clear Byte Pointer Flip-Flop

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

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

DMA Master Clear

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

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

Clear Mask Register

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

3.5 PCI DMA

The 82801E C-ICH provides support for the PC/PCI DMA protocol. PC/PCI DMA uses dedicated request and grant signals to permit PCI devices to request transfers associated with specific DMA channels. Upon receiving a request and getting control of the PCI bus, the 82801E C-ICH performs a two-cycle transfer. For example, if data is to be moved from the peripheral to main memory, the 82801E C-ICH first reads data from the peripheral and then writes the data to main memory. The location in main memory is the Current Address Registers in the 8237.

82801E C-ICH supports up to two PC/PCI REQ/GNT pairs, REQ[A:B]# and GNT[A:B]#.

A 16-bit register is included in the 82801E C-ICH Function 0 PCI configuration space at offset 90h. It is divided into seven 2-bit fields that are used to configure the seven DMA channels.

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

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

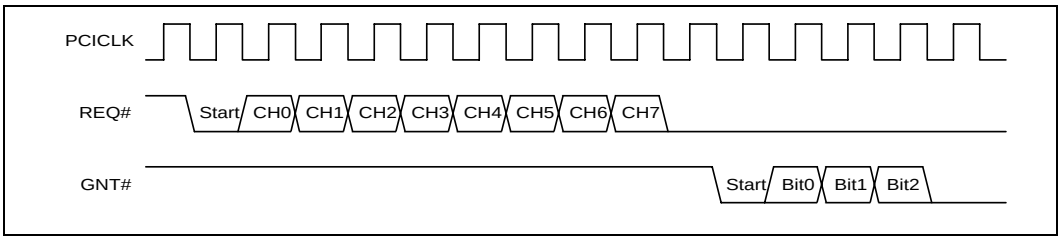
It is not possible for a particular DMA channel to be configured for more than one style of DMA; however, the seven channels can be programmed independently. For example, channel 3 can be set up for PC/PCI and channel 5 set up for LPC DMA.

The 82801E C-ICH REQ[A:B]# and GNT[A:B]# can be configured for support of a PC/PCI DMA Expansion agent. The PCI DMA Expansion agent can then provide DMA service or ISA Bus Master service using the 82801E C-ICH DMA controller. The REQ#/GNT# pair must follow the PC/PCI serial protocol described in the following section.

3.5.1 PCI DMA Expansion Protocol

The PCI expansion agent must support the PCI expansion Channel Passing Protocol defined in Figure 12 for both the REQ# and GNT# pins.

Figure 12. DMA Serial Channel Passing Protocol



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

The 82801E C-ICH encodes the granted channel on the GNT# line as shown above where the bits have the same meaning as shown in Figure 12. For example, the sequence
[start, bit 0, bit 1, bit 2]=[0,1,0,0] grants DMA channel 1 to the requesting device, and the sequence [start, bit 0, bit 1, bit 2]=[0,0,1,1] grants DMA channel 6 to the requesting device.

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

1. If a PCI DMA expansion agent has more than one request active, it must resend the request serial protocol after one of the requests has been granted the bus and it has completed its transfer. The expansion device should drive its REQ# inactive for two clocks and then transmit the serial channel passing protocol again, even if there are no new requests from the PCI expansion agent to 82801E C-ICH. For example, if a PCI expansion agent had active requests for DMA Channel 1 and Channel 5, it would pass this information to the 82801E C-ICH through the expansion channel passing protocol. If, after receiving GNT# (assume for CH5) and having the device finish its transfer (device stops driving request to PCI expansion agent), it would then need to re-transmit the expansion channel passing protocol to inform the 82801E C-ICH that DMA channel 1 was still requesting the bus, even if that was the only request the expansion device had pending.
2. If a PCI DMA expansion agent has a request go inactive before 82801E C-ICH asserts GNT#, it must resend the expansion channel passing protocol to update the 82801E C-ICH with this new request information. For example, if a PCI expansion agent has DMA channel 1 and 2 requests pending, it sends them serially to 82801E C-ICH using the expansion channel passing protocol. If, however, DMA channel 1 goes inactive into the expansion agent before the expansion agent receives a GNT# from 82801E C-ICH, the expansion agent must pull its REQ# line high for 1clock and resend the expansion channel passing information with only DMA channel 2 active. Note that the 82801E C-ICH does not do anything special to catch this case because a DREQ going inactive before a DACK# is received is not allowed in the ISA DMA protocol and, therefore, does not need to work properly in this protocol either. This requirement is needed to be able to support Plug-n-Play ISA devices that toggle DREQ# lines to determine if those lines are free in the system.
3. If a PCI expansion agent has sent its serial request information and receives a new DMA request before receiving GNT#, the agent must resend the serial request with the new request active. For example, if a PCI expansion agent has already passed requests for DMA channel 1 and 2 and detects DREQ 3 active before a GNT is received, the device must pull its REQ# line high for one clock and resend the expansion channel passing information with all three channels active.

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

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

3.5.2 PCI DMA Expansion Cycles

82801E C-ICH's support of the PC/PCI DMA Protocol currently consists of four types of cycles: Memory-to-I/O, I/O-to-Memory, Verify, and ISA Master cycles. ISA Masters are supported through the use of a DMA channel that has been programmed for cascade mode.

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

The I/O portion of the DMA cycle generates a PCI I/O cycle to one of four I/O addresses (Table 14). Note that these cycles must be qualified by an active GNT# signal to the requesting device.

Table 14. DMA Cycle vs. I/O Address

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

3.5.3 DMA Addresses

The memory portion of the cycle generates a PCI memory read or memory write bus cycle; its address representing the selected memory. The I/O portion of the DMA cycle generates a PCI I/O cycle to one of the four I/O addresses listed in Table 14.

3.5.4 DMA Data Generation

The data generated by PC/PCI devices on I/O reads when they have an active GNT# is on the lower two bytes of the PCI AD bus. Table 15 lists the PCI pins that the data appears for 8-bit and 16-bit channels. Each I/O read results in one memory write and each memory read results in one I/O write. If the I/O device is 8 bit, the 82801E C-ICH performs an 8-bit memory write. The 82801E C-ICH does not assemble the I/O read into a DWord for writing to memory. Similarly, the 82801E C-ICH does not disassemble a DWord read from memory to the I/O device.

Table 15. PCI Data Bus vs. DMA I/O Port Size

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

3.5.5 DMA Byte Enable Generation

The byte enables generated by the 82801E C-ICH on I/O reads and writes must correspond to the size of the I/O device. Table 16 defines the byte enables asserted for 8-bit and 16-bit DMA cycles.

Table 16. DMA I/O Cycle Width vs. BE[3:0]#

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

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

3.5.6 DMA Cycle Termination

DMA cycles are terminated when a terminal count is reached in the DMA controller and the channel is not in autoinitialize mode or when the PC/PCI device deasserts its request. The PC/PCI device must follow explicit rules when deasserting its request or the 82801E C-ICH may not see it in time and run an extra I/O and memory cycle.

The PC/PCI device must deassert its request seven PCICLKs before it generates TRDY# on the I/O read or write cycle or the 82801E C-ICH is allowed to generate another DMA cycle. For transfers to memory, this means that the memory portion of the cycle will be run without an asserted PC/PCI REQ#.

3.5.7 LPC DMA

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

3.5.8 Asserting DMA Requests

Peripherals that need DMA service encode their requested channel number on the LDRQ# signal. To simplify the protocol, each peripheral on the LPC I/F has its own dedicated LDRQ# signal (they may not be shared between two separate peripherals). The 82801E C-ICH has two LDRQ# inputs, allowing at least two devices to support DMA or bus mastering.

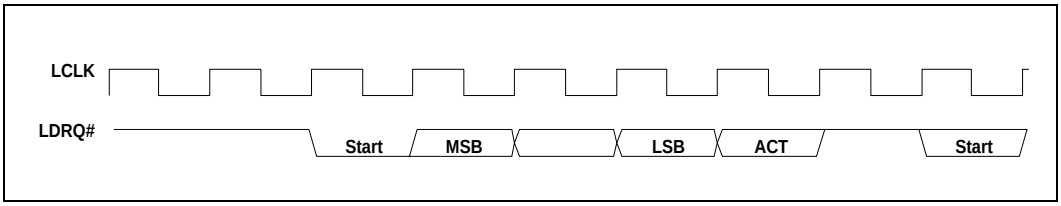
LDRQ# is synchronous with LCLK (PCI clock). As shown in Figure 13 the peripheral uses the following serial encoding sequence:

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

- After the active/inactive indication, the LDRQ# signal must go high for at least one clock. After that one clock, LDRQ# signal can be brought low to the next encoding sequence.

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

Figure 13. DMA Request Assertion Through LDRQ#



3.5.9 Abandoning DMA Requests

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

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

In these cases, the peripheral wishes to stop further DMA activity. It may do so by sending an LDRQ# message with the ACT bit as 0. However, since the DMA request was seen by the 82801E C-ICH, there is no guarantee that the cycle has not been granted and will shortly run on LPC. Therefore, peripherals must take into account that a DMA cycle may still occur. The peripheral can choose not to respond to this cycle, in which case the host aborts it or the host can choose to complete the cycle normally with any random data.

This method of DMA deassertion should be prevented when possible to limit boundary conditions both on the 82801E C-ICH and the peripheral.

3.5.10 General Flow of DMA Transfers

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

1. 82801E C-ICH starts transfer by asserting 0000b on LAD[3:0] with LFRAME# asserted.
2. 82801E C-ICH asserts 'cycle type' of DMA, direction based on DMA transfer direction.
3. 82801E C-ICH asserts channel number and, if applicable, terminal count.
4. 82801E C-ICH indicates the size of the transfer: 8 bits.
5. If a DMA read,
   — The 82801E C-ICH drives the first 8 bits of data and turns the bus around.

— The peripheral acknowledges the data with a valid SYNC.

6. If a DMA write,
   — The 82801E C-ICH turns the bus around and waits for data.
   — The peripheral indicates data ready through SYNC and transfers the first byte.
7. The peripheral turns around the bus.

3.5.11 Terminal Count

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

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

3.5.12 Verify Mode

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

3.5.13 DMA Request Deassertion

An end of transfer is communicated to the 82801E C-ICH through a special SYNC field transmitted by the peripheral. An LPC device must not attempt to signal the end of a transfer by deasserting LDREQ#. If a DMA transfer is several bytes (e.g., a transfer from a demand mode device), the 82801E C-ICH needs to know when to deassert the DMA request based on the data currently being transferred.

The DMA agent uses a SYNC encoding on each byte of data being transferred that indicates to the 82801E C-ICH whether this is the last byte of transfer or if more bytes are requested. To indicate the last byte of transfer, the peripheral uses a SYNC value of 0000b (ready with no error), or ‘1010b’ (ready with error). These encodings tell the 82801E C-ICH that this is the last piece of data transferred on a DMA read (82801E C-ICH to peripheral), or that the byte which follows is the last piece of data transferred on a DMA write (peripheral to 82801E C-ICH).

When the 82801E C-ICH sees one of these two encodings, it ends the DMA transfer after this byte and deasserts the DMA request to the 8237. Therefore, if the 82801E C-ICH indicated a 16 bit transfer, the peripheral can end the transfer after one byte by indicating a SYNC value of 0000b or 1010b. The 82801E C-ICH will not attempt to transfer the second byte, and will deassert the DMA request internally.

When the peripheral indicates a 0000b or 1010b SYNC pattern on the last byte of the indicated size, the 82801E C-ICH only deasserts the DMA request to the 8237, since it does not need to end the transfer.

When the peripheral wishes to keep the DMA request active, it uses a SYNC value of 1001b (ready plus more data). This indicates to the 8237 that more data bytes are requested after the current byte has been transferred; the 82801E C-ICH keeps the DMA request active to the 8237. Therefore, on an 8-bit transfer size, if the peripheral indicates a SYNC value of 1001b to the 82801E C-ICH, the data will be transferred and the DMA request remains active to the 8237. At a later time, the 82801E C-ICH will then come back with another START - CYCTYPE - CHANNEL - SIZE etc. combination to initiate another transfer to the peripheral.

The peripheral must not assume that the next START indication from the 82801E C-ICH is another grant to the peripheral if it had indicated a SYNC value of 1001b. On a single mode DMA device, the 8237 re-arbitrates after every transfer. Only demand mode DMA devices can be guaranteed that they will receive the next START indication from the 82801E C-ICH.

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

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

3.5.14 SYNC Field/LDRQ# Rules

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

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

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

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

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

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

3.6 8254 Timers (D31:F0)

The 82801E C-ICH contains three counters that have fixed uses. All registers and functions associated with the 8254 timers are in the Core well. The 8254 unit is clocked by a 14.31818 MHz clock.

Counter 0, System Timer

This counter functions as the system timer by controlling the state of IRQ0. It is typically programmed for Mode 3 operation. The counter produces a square wave with a period equal to the product of the counter period (838 ns) and the initial count value. The counter loads the initial count value one counter period after software writes the count value to the counter I/O address. The counter initially asserts IRQ0 and decrements the count value by two each counter period. The counter negates IRQ0 when the count value reaches 0. It then reloads the initial count value and again decrements the initial count value by two each counter period. The counter then asserts IRQ0 when the count value reaches 0, reloads the initial count value, and repeats the cycle; alternately asserting and negating IRQ0.

Counter 1, Refresh Request Signal

This counter provides the refresh request signal and is typically programmed for Mode 2 operation. The counter negates refresh request for one counter period (838 ns) during each count cycle. The initial count value is loaded one counter period after being written to the counter I/O address. The counter initially asserts refresh request and negates it for 1 counter period when the count value reaches 1. The counter then asserts refresh request and continues counting from the initial count value.

Counter 2, Speaker Tone

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

3.6.1 Timer Programming

The counter/timers are programmed in the following fashion:

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

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

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

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

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

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

Table 17 lists the six operating modes for the interval counters.

Table 17. Counter Operating Modes

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

3.6.2 Reading from the Interval Timer

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

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

Simple Read

The first method is to perform a simple read operation. The counter is selected through port 40h (counter 0), 41h (counter 1), or 42h (counter 2).

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

Counter Latch Command

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

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

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

Read Back Command

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

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

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

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

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

3.7 8259 Programmable Interrupt Controllers (PIC) (D31:F0)

The 82801E C-ICH incorporates the functionality of two 8259 interrupt controllers that provide system interrupts for the ISA compatible interrupts. These interrupts are: system timer, keyboard controller, serial ports, parallel ports, floppy disk, IDE, mouse, and DMA channels. In addition, this interrupt controller can support the PCI-based interrupts, by mapping the PCI interrupt onto the compatible ISA interrupt line. Each 8259 core supports 8 interrupts, numbered 0–7. Table 18 shows how the cores are connected.

Table 18. Interrupt Controller Core Connections

8259	8259 Input	Typical Interrupt Source	Connected Pin/Function
Master	0	Internal	Internal Timer/Counter 0 output
	1	Keyboard	IRQ1 via SERIRQ
	2	Internal	Slave Controller INTR output
	3	Serial Port A	IRQ3 via SERIRQ
	4	Serial Port B	IRQ4 via SERIRQ
	5	Parallel Port/Generic	IRQ5 via SERIRQ
	6	Floppy Disk	IRQ6 via SERIRQ
	7	Parallel Port/Generic	IRQ7 via SERIRQ
Slave	0	Internal Real Time Clock	Internal RTC
	1	Generic	IRQ9 via SERIRQ
	2	Generic	IRQ10 via SERIRQ
	3	Generic	IRQ11 via SERIRQ
	4	PS/2 Mouse	IRQ12 via SERIRQ
	5	Internal	State Machine output based on processor FERR# assertion.
	6	Primary IDE cable	IRQ14 from input signal or via SERIRQ
	7	Secondary IDE Cable	IRQ15 from input signal or via SERIRQ

The 82801E C-ICH cascades the slave controller onto the master controller through master controller interrupt input 2. This means there are only 15 possible interrupts for the 82801E C-ICH PIC. Interrupts can individually be programmed to be edge or level, except for IRQ[0, 2, 8#, 13].

Note that previous PIIX*n* devices internally latched IRQ[12 and 1] and required a port 60h read to clear the latch. The 82801E C-ICH can be programmed to latch IRQ[12 or 1] (see bit 11 and bit 12 in General Control Register, D31:F0, offset D0h).

3.7.1 Interrupt Handling

3.7.1.1 Generating Interrupts

The PIC interrupt sequence involves three bits, from the IRR, ISR, and IMR for each interrupt level. These bits are used to determine the interrupt vector returned, and status of any other pending interrupts. Table 19 defines the IRR, ISR, and IMR.

Table 19. Interrupt Status Registers

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

3.7.1.2 Acknowledging Interrupts

The processor generates an interrupt acknowledge cycle that is translated by the host bridge into a PCI Interrupt Acknowledge Cycle to the 82801E C-ICH. The PIC translates this command into two internal INTA# pulses expected by the 8259 cores. The PIC uses the first internal INTA# pulse to freeze the state of the interrupts for priority resolution. On the second INTA# pulse, the master or slave sends the interrupt vector to the processor with the acknowledged interrupt code. This code is based on bits [7:3] of the corresponding ICW2 register combined with three bits representing the interrupt within that controller.

Table 20. Content of Interrupt Vector Byte

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

3.7.1.3 Hardware/Software Interrupt Sequence

1. One or more of the Interrupt Request lines (IRQ) are raised high in edge mode, or seen high in level mode, setting the corresponding IRR bit.
2. The PIC sends INTR active to the processor if an asserted interrupt is not masked.
3. The processor acknowledges the INTR and responds with an interrupt acknowledge cycle. The cycle is translated into a PCI interrupt acknowledge cycle by the host bridge. This command is broadcast over PCI by the 82801E C-ICH.

4. Upon observing its own interrupt acknowledge cycle on PCI, the 82801E C-ICH converts it into the two cycles that the internal 8259 pair can respond to. Each cycle appears as an interrupt acknowledge pulse on the internal INTA# pin of the cascaded interrupt controllers.
5. Upon receiving the first internally generated INTA# pulse, the highest priority ISR bit is set and the corresponding IRR bit is reset. On the trailing edge of the first pulse a slave identification code is broadcast by the master to the slave on a private, internal three bit wide bus. The slave controller uses these bits to determine if it must respond with an interrupt vector during the second INTA# pulse.
6. Upon receiving the second internally generated INTA# pulse, the PIC returns the interrupt vector. If no interrupt request is present because the request was too short in duration, the PIC will return vector 7 from the master controller.
7. This completes the interrupt cycle. In AEOI mode the ISR bit is reset at the end of the second INTA# pulse. Otherwise, the ISR bit remains set until an appropriate EOI command is issued at the end of the interrupt subroutine.

3.7.2 Initialization Command Words (ICWx)

Before the operation can begin, each 8259 must be initialized. In the 82801E C-ICH this is a four-byte sequence. The four initialization command words are referred to by their acronyms: ICW1, ICW2, ICW3, and ICW4.

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

ICW1

An I/O write to the master or slave controller base address with data bit 4 equal to 1 is interpreted as a write to ICW1. Upon sensing this write, the 82801E C-ICH PIC expects three more byte writes to 21h for the master controller, or A1h for the slave controller, to complete the ICW sequence.

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

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

ICW2

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

ICW3

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

- For the master controller, ICW3 is used to indicate which IRQ input line is used to cascade the slave controller. Within the 82801E C-ICH, IRQ2 is used. Therefore, bit 2 of ICW3 on the master controller is set to a 1 and the other bits are set to 0s.
- For the slave controller, ICW3 is the slave identification code used during an interrupt acknowledge cycle. On interrupt acknowledge cycles, the master controller broadcasts a code to the slave controller if the cascaded interrupt won arbitration on the master controller. The slave controller compares this identification code to the value stored in its ICW3, and if it matches, the slave controller assumes responsibility for broadcasting the interrupt vector.

ICW4

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

3.7.3 Operation Command Words (OCW)

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

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

3.7.4 Modes of Operation

Fully Nested Mode

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

Special Fully Nested Mode

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

- When an interrupt request from a certain slave is in service, this slave is not locked out from the master's priority logic and further interrupt requests from higher priority interrupts within the slave will be recognized by the master and will initiate interrupts to the processor. In the normal nested mode, a slave is masked out when its request is in service.

- When exiting the Interrupt Service routine, software must check whether the interrupt serviced was the only one from that slave. This is done by sending a Non-Specific EOI command to the slave and then reading its ISR. If it is 0, a non-specific EOI can also be sent to the master.

Automatic Rotation Mode (Equal Priority Devices)

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

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

Specific Rotation Mode (Specific Priority)

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

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

Poll Mode

Poll Mode can be used to conserve space in the interrupt vector table. Multiple interrupts that can be serviced by one interrupt service routine do not need separate vectors if the service routine uses the poll command. Polled Mode can also be used to expand the number of interrupts. The polling interrupt service routine can call the appropriate service routine, instead of providing the interrupt vectors in the vector table. In this mode, the INTR output is not used and the microprocessor internal Interrupt Enable flip-flop is reset, disabling its interrupt input. Service to devices is achieved by software using a Poll Command.

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

Cascade Mode

The PIC in the 82801E C-ICH has one master 8259 and one slave 8259 cascaded onto the master through IRQ2. This configuration can handle up to 15 separate priority levels. The master controls the slaves through a 3-bit internal bus. In the 82801E C-ICH, when the master drives 010b on this bus, the slave controller takes responsibility for returning the interrupt vector. An EOI Command must be issued twice: once for the master and once for the slave.

Edge-Triggered and Level-Triggered Mode

In ISA systems this mode is programmed using bit 3 in ICW1, which sets level or edge for the entire controller. In the 82801E C-ICH, this bit is disabled and a new register for edge-triggered and level-triggered mode selection, per interrupt input, is included. This is the Edge/Level control Registers ELCR1 and ELCR2.

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

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

End of Interrupt Operations

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

Normal End of Interrupt

In Normal EOI, software writes an EOI command before leaving the interrupt service routine to mark the interrupt as completed. There are two forms of EOI commands: Specific and Non-Specific. When a Non-Specific EOI command is issued, the PIC will clear the highest ISR bit of those that are set to 1. Non-Specific EOI is the normal mode of operation of the PIC within the 82801E C-ICH, as the interrupt being serviced currently is the interrupt entered with the interrupt acknowledge. When the PIC is operated in modes that preserve the fully nested structure, software can determine which ISR bit to clear by issuing a Specific EOI. An ISR bit that is masked will not be cleared by a Non-Specific EOI if the PIC is in the Special Mask Mode. An EOI command must be issued for both the master and slave controller.

Automatic End of Interrupt Mode

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

3.7.5 Masking Interrupts

Masking on an Individual Interrupt Request

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

Special Mask Mode

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

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

3.7.6 Steering PCI Interrupts

The 82801E C-ICH can be programmed to allow PIRQA#–PIRQH# to be internally routed to interrupts [3:7, 9:12, 14 or 15]. The assignment is programmable through the PIRQ*n* Route Control registers, located at 60–63h and 68–6Bh in function 0. One or more PIRQ*n*# lines can be routed to the same IRQx input. If interrupt steering is not required, the Route Registers can be programmed to disable steering.

The PIRQ*n*# lines are defined as active low, level sensitive to allow multiple interrupts on a PCI Board to share a single line across the connector. When a PIRQ*n*# is routed to a specified IRQ line, software must change the IRQ's corresponding ELCR bit to level sensitive mode. The 82801E C-ICH will internally invert the PIRQ*n*# line to send an active high level to the PIC. When a PCI interrupt is routed onto the PIC, the selected IRQ can no longer be used by an ISA device (through SERIRQ). However, active low non-ISA interrupts can share their interrupt with PCI interrupts.

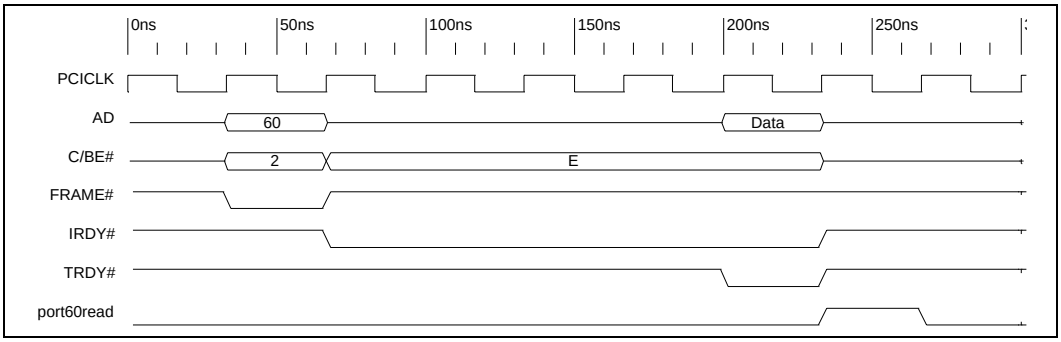
3.7.7 Special Handling of IRQ1 and IRQ12

IRQ1 and IRQ12 interrupts are treated in a slightly different fashion from other interrupts in the system. In a legacy PC environment, these interrupts were not held active until serviced, but rather pulsed whenever a key or button was pressed. In newer systems, this pulsing is no longer done. However, the 82801E C-ICH must still handle old keyboard controllers that perform the pulse operation. Therefore, the 82801E C-ICH contains logic that can sample and hold these interrupts if so required.

Two register bits in configuration register D0h in function 0 enable the latching of IRQ1 and 12. IRQ1 can optionally be latched through bit 12, and IRQ12 can optionally be latched through bit 11. When these bits are set, the corresponding interrupt is held to the 8259 until an I/O read from port 60 is seen. The port 60 read is an indication to the keyboard controller that the interrupt has been serviced.

With PIIX1 to PIIX4 devices, it was always guaranteed that the keyboard controller would exist behind the PIIX on the ISA bus. With the 82801E C-ICH, this is not the case. Therefore, clearing the latch must be done through a snoop of port 60h. The waveform that performs this snoop is shown in Figure 14. Note that the signal that indicates that a port 60 read occurred is only one PCI clock wide. This cannot be a handshake signal because the 82801E C-ICH is not necessarily responding to the cycle.

Figure 14. Port 60 Read Clearing IRQ1/12 Latch



3.8 Advanced Interrupt Controller (APIC) (D31:F0)

In addition to the standard ISA compatible interrupt controller (PIC) described in the previous section, the 82801E C-ICH incorporates the Advanced Programmable Interrupt Controller (APIC). While the standard interrupt controller is intended for use in a uni-processor system, APIC can be used in either a uni-processor or multi-processor system.

3.8.1 Interrupt Handling

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

- **Method of Interrupt Transmission.** The I/O APIC transmits interrupts through a 3-wire bus and interrupts are handled without the need for the processor to run an interrupt acknowledge cycle.
- **Interrupt Priority.** The priority of interrupts in the I/O APIC is independent of the interrupt number. For example, interrupt 10 can be given a higher priority than interrupt 3.
- **More Interrupts.** The I/O APIC in the 82801E C-ICH supports a total of 24 interrupts.
- **Multiple Interrupt Controllers.** The I/O APIC interrupt transmission protocol has an arbitration phase that allows for multiple I/O APICs in the system with their own interrupt vectors. The 82801E C-ICH I/O APIC must arbitrate for the APIC bus before transmitting its interrupt message.

3.8.2 Interrupt Mapping

The I/O APIC within the 82801E C-ICH supports 24 APIC interrupts. Each interrupt has its own unique vector assigned by software. The interrupt vectors are mapped as shown in Table 21 and match “configuration 6” of the multi-processor specification.

Table 21. APIC Interrupt Mapping (Sheet 1 of 2)

IRQ #	Via SERIRQ	Direct from pin	Via PCI message	Internal Modules
0	No	No	No	Cascade from 8259 #1
1	Yes	No	Yes	
2	No	No	No	8254 Counter 0
3	Yes	No	Yes	
4	Yes	No	Yes	
5	Yes	No	Yes	
6	Yes	No	Yes	
7	Yes	No	Yes	
8	No	No	No	RTC
9	Yes	No	Yes	Option for SCI, TCO
10	Yes	No	Yes	Option for SCI, TCO
11	Yes	No	Yes	Option for SCI, TCO
12	Yes	No	Yes	

Table 21. APIC Interrupt Mapping (Sheet 2 of 2)

IRQ #	Via SERIRQ	Direct from pin	Via PCI message	Internal Modules
13	No	No	No	FERR# logic
14	Yes	Yes	Yes	
15	Yes	Yes	Yes	
16	PIRQA	PIRQA	No	
17	PIRQB	PIRQB	No	Option for SMbus
18	PIRQC	PIRQC	No	
19	PIRQD	PIRQD	No	USB
20	N/A	PIRQE	Yes	LAN0
21	N/A	PIRQF	Yes	LAN1
22	N/A	PIRQG	Yes	Option for SCI, TCO
23	N/A	PIRQH	Yes	Option for SCI, TCO

3.8.3 APIC Bus Functional Description

3.8.3.1 Physical Characteristics of APIC

The APIC bus is a three-wire synchronous bus that connects all I/O and local APICs. Two of these wires are used for data transmission and one wire is a clock. For bus arbitration, the APIC uses only one of the data wires. The bus is logically a wire-OR and electrically an open-drain connection. This provides for both bus use arbitration and arbitration for lowest priority. The APIC bus speed is 16.6667 MHz.

3.8.3.2 APIC Bus Arbitration

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

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

When 82801E C-ICH detects a bus idle condition on the APIC Bus and it has an interrupt to send over the APIC bus, it drives a start cycle to begin arbitration, by driving bit 0 to a '0' on an APICCLK rising edge. It then samples bit 1. If Bit 1 was a '0', then a local APIC started arbitration for an EOI message on the same clock edge that the 82801E C-ICH started arbitration. Thus, the 82801E C-ICH has lost arbitration and stops driving the APIC bus.

When the 82801E C-ICH does not detect an EOI message start, it starts transferring its arbitration ID, located in bits [27:24] of its Arbitration ID register (ARBID). Starting in Cycle 2 through Cycle 5, it will tri-state bit 0, and drive bit 1 to a '0' if ARBID[27] is a '1'. If ARBID[27] is a '0', it will also tri-state bit 1. At the end of each cycle, the 82801E C-ICH samples the state of Bit 1 on the APIC bus. If the 82801E C-ICH did not drive Bit 1 (ARBID[27] = '0') and it samples a '0', then another APIC agent started arbitration for the APIC bus at the same time as the 82801E C-ICH, and it has higher priority. The 82801E C-ICH will stop driving the APIC bus. Table 22 describes the arbitration cycles.

Table 22. Arbitration Cycles

Cycle	Bit 1	Bit 0	Comment
1	EOI	0	Bit 1 = 1: Normal, Bit 1 = 0: EOI
2	NOT (**ARBID[27]**)	1	Arbitration ID. If 82801E C-ICH samples a different value than it sent, it lost arbitration.
3	NOT (**ARBID[26]**)	1	
4	NOT (**ARBID[25]**)	1	
5	NOT (**ARBID[24]**)	1	

3.8.3.3 Bus Message Formats

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

Table 23. APIC Message Formats

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

EOI Message For Level-Triggered Interrupts

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

Table 24. EOI Message

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

Short Message

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

Table 25. Short Message

Cycle	Bit 1	Bit 0	Comments
1	1	0	Normal Arbitration
2–5	ARBID	1	Arbitration ID
6	NOT(DM)	NOT(M2)	DM[1] = Destination Mode from bit 11 of the redirection table register
7	NOT(M1)	NOT(M0)	M2-M0 = Delivery Mode from bits 10:8 of the redirection table register
8	NOT(L)	NOT(TM)	L = Level, TM = Trigger Mode
9	NOT(V7)	NOT(V6)	Interrupt vector bits V7–V0 from redirection table register
10	NOT(V5)	NOT(V4)	
11	NOT(V3)	NOT(V2)	
12	NOT(V1)	NOT(V0)	
13	NOT(D7)	NOT(D6)	Destination field from bits 63:56 of redirection table register[1]
14	NOT(D5)	NOT(D4)	
15	NOT(D3)	NOT(D2)	
16	NOT(D1)	NOT(D0)	
17	NOT(C1)	NOT(C0)	Checksum for Cycles 6–16[2]
18	1	1	Postamble[3]
19	NOT(A)	NOT(A)	Status Cycle 0. See Table 26.
20	NOT(A1)	NOT(A1)	Status Cycle 1. See Table 26.
21	1	1	Idle

NOTES:
1. If DM is 0 (physical mode), then cycles 15 and 16 are the APIC ID and cycles 13 and 14 are sent as '1'. If DM is 1 (logical mode), then cycles 13 through 16 are the 8-bit Destination field. The interpretation of the logical mode 8-bit Destination field is performed by the local units using the Destination Format Register. Shorthands of "all-incl-self" and "all-excl-self" both use Physical Destination mode and a destination field containing APIC ID value of all ones. The sending APIC knows whether it should (incl) or should not (excl) respond to its own message.
2. The checksum field is the cumulative add (mod 4) of all data bits (DM, M0-3, L, TM, V0-7,D0-7). The APIC driving the message provides this checksum. This, in essence, is the lower two bits of an adder at the end of the message.
3. This cycle allows all APICs to perform various internal computations based on the information contained in the received message. One of the computations takes the checksum of the data received in cycles 6 through 16 and compares it with the value in cycle 18. If any APIC computes a different checksum than the one passed in cycle 17, then that APIC will signal an error on the APIC bus ("00") in cycle 19. If this happens, all APICs will assume the message was never sent and the sender must try sending the message again, which includes re-arbitrating for the APIC bus. In lowest priority delivery when the interrupt has a focus processor, the focus processor will signal this by driving a "01" during cycle 19. This tells all the other APICs that the interrupt has been accepted, the arbitration is preempted, and short message format is used. Cycle 19 and 20 indicates the status of the message (i.e., accepted, check sum error, retry or error). Table 26 shows the status signal combinations and their meanings for all delivery modes.

Table 26. APIC Bus Status Cycle Definition

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

Lowest Priority without Focus Processor (FP) Message

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

Table 27. Lowest Priority Message (Without Focus Processor)

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

NOTES:
1. Cycle 21 through 28 are used to arbitrate for the lowest priority processor. The processor that takes part in the arbitration drives the processor priority on the bus. Only the local APICs that have "free interrupt slots" will participate in the lowest priority arbitration.
2. Cycles 29 through 32 are used to break tie in case two more processors have lowest priority. The bus arbitration IDs are used to break the tie.

Remote Read Message

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

Table 28. Remote Read Message

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

NOTE: Cycles 21 through 36 contain the remote register data. The status information in cycle 37 specifies if the data is good or not. Remote read cycle is always successful (although the data may be valid or invalid) in that it is never retried. The reason for this is that Remote Read is a debug feature, and a “hung” remote APIC that is unable to respond should not cause the debugger to hang.

3.8.4 PCI Message-Based Interrupts

3.8.4.1 Theory of Operation

The following scheme is only supported when the internal I/O(x) APIC is used (rather than just the 8259). The 82801E C-ICH supports the new method for PCI devices to deliver interrupts as write cycles, rather than using the PIRQ[A:H] signals. Essentially, the PCI devices are given a write path directly to a register that will cause the desired interrupt. This mode is only supported when the 82801E C-ICH's internal I/O APIC is enabled. Upon recognizing the write from the peripheral, the 82801E C-ICH sends the interrupt message to the processor using the I/O APIC's serial bus.

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

The following sequence is used:

1. During PCI PnP, the PCI peripheral is first programmed with an address (MESSAGE_ADDRESS) and data value (MESSAGE_DATA) that will be used for the interrupt message delivery. For the 82801E C-ICH, the MESSAGE_ADDRESS is the IRQ Pin Assertion Register, which is mapped to memory location: FEC0_0020h (same as APIC).
2. To cause the interrupt, the PCI peripheral requests the PCI bus and when granted, writes the MESSAGE_DATA value to the location indicated by the MESSAGE_ADDRESS. The MESSAGE_DATA value indicates which interrupt occurred. This MESSAGE_DATA value is a binary encoded. For example, to indicate that interrupt 7 should go active, the peripheral will write a binary value of 0000111. The MESSAGE_DATA will be a 32-bit value, although only the lower 5 bits are used.
3. If the PRQ bit in the APIC Version Register is set, the 82801E C-ICH positively decodes the cycles (as a slave) in medium time.
4. The 82801E C-ICH decodes the binary value written to MESSAGE_ADDRESS and sets the appropriate IRR bit in the internal I/O APIC. The corresponding interrupt must be set up for edge-triggered interrupts. The 82801E C-ICH supports interrupts 00h through 23h. Binary values outside this range will not cause any action.
5. After sending the interrupt message to the processor, the 82801E C-ICH automatically clears the interrupt.

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

The 82801E C-ICH ignores interrupt messages sent by PCI masters that attempt to use IRQ[0,2,8, or 13].

3.8.4.2 Registers and Bits Associated with PCI Interrupt Delivery

Capabilities Indication

The capability to support PCI interrupt delivery will be indicated via APIC configuration techniques. This involves the BIOS creating a data structure that gets reported to the APIC configuration software. The operating system reads the PRQ bit in the APIC Version Register to see if the 82801E C-ICH is capable of support PCI-based interrupt messages. As a precaution, the PRQ bit is not set if the XAPIC_EN bit is not set.

Interrupt Message Register

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

3.8.5 Processor-Side Bus Interrupt Delivery

3.8.5.1 Theory of Operation

For processors that support Processor-Side Bus (PSB) interrupt delivery, the 82801E C-ICH has an option to let the integrated I/O APIC behave as an I/O (x) APIC. In this case, it delivers interrupt messages to the processor in a parallel manner, rather than using the I/O APIC serial scheme. The 82801E C-ICH is intended to be compatible with the I/O (x) APIC specification, Revision 1.1.

This is done by the 82801E C-ICH writing (via the Hub Interface) directly to a memory location that is snooped by the processor(s). The processor(s) snoop the cycle to know which one goes active.

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

The following sequence is used:

1. When the 82801E C-ICH detects an interrupt event (active edge for edge-triggered mode or a change for level-triggered mode), it sets or resets the internal IRR bit associated with that interrupt.
2. Internally, the 82801E C-ICH requests to use the bus in a way the automatically flushes upstream buffers. This can be internally implemented similar to a DMA device request.
3. The 82801E C-ICH then delivers the message by performing a write cycle to the appropriate address with the appropriate data. The address and data formats are described in “Interrupt Message Format” on page 88.

Notes:

1. PSB Interrupt Delivery compatibility with processor clock control depends on the processor, not the 82801E C-ICH.
2. The local APIC (in the processor) has a delivery mode option to interpret Processor-Side Bus (PSB) messages as an SMI in which case the processor treats the incoming interrupt as an SMI instead of as an interrupt. This does not mean that the 82801E C-ICH has any way to have an SMI source from the 82801E C-ICH power management logic cause the I/O APIC to send an SMI message; there is no way to do this. The 82801E C-ICH’s I/O APIC can only send interrupts due to interrupts which do not include SMI, NMI or INIT. This means that in IA32/IA64 based platforms, FSB interrupt message format delivery modes 010 (SMI/PMI), 100 (NMI), and 101 (INIT), must not be used and is not supported. Only the hardware pin connection is supported by the 82801E C-ICH.

3.8.5.2 Edge-Triggered Operation

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

3.8.5.3 Level-Triggered Operation

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

3.8.5.4 Registers Associated with Processor-Side Bus Interrupt Delivery

Capabilities Indication

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

DT bit in the Boot Configuration Register

This enables the 82801E C-ICH to deliver interrupts as memory writes. This bit is ignored if the APIC mode is not enabled.

3.8.5.5 Interrupt Message Format

82801E C-ICH writes the message to PCI (and to the Host Controller) as a 32-bit memory write cycle. It uses the formats shown in Table 29 and Table 30 for the address and data.

Table 29. Interrupt Message Address Format

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

Table 30. Interrupt Message Data Format

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

3.9 Serial Interrupt (D31:F0)

82801E C-ICH supports a serial IRQ scheme. This allows a single signal to be used to report interrupt requests. The signal (used to transmit this information) is shared between the host, the 82801E C-ICH, and all peripherals that support serial interrupts. The signal line (SERIRQ) is synchronous to PCI clock and follows the sustained tri-state protocol that is used by all PCI signals. This means that if a device has driven SERIRQ low, it will first drive it high synchronous to PCI clock and release it the following PCI clock. The serial IRQ protocol defines this sustained tri-state signaling in the following fashion:

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

The 82801E C-ICH supports a message for 21 serial interrupts. These represent the 15 ISA interrupts (IRQ[0,1, 2:15]), the four PCI interrupts, and the SMI# and IOCHK# control signals. The serial IRQ protocol does not support the additional APIC interrupts (20–23).

3.9.1 Start Frame

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

- Continuous, where the 82801E C-ICH is solely responsible for generating the start frame
- Quiet, where a serial IRQ peripheral is responsible for beginning the start frame.

The mode that must first be entered when enabling the serial IRQ protocol is continuous mode. In this mode, the 82801E C-ICH will assert the start frame. This start frame is 4, 6, or 8 PCI clocks wide based upon the Serial IRQ Control Register, bits 1:0 at 64h in Device 31:Function 0 configuration space. This is a polling mode.

When the serial IRQ stream enters quiet mode (signaled in the Stop Frame), the SERIRQ line remains inactive and pulled up between the Stop and Start Frame until a peripheral drives the SERIRQ signal low. The 82801E C-ICH senses the line low and continues to drive it low for the remainder of the Start Frame. Since the first PCI clock of the start frame was driven by the peripheral in this mode, the 82801E C-ICH drives the SERIRQ line low for one PCI clock less than in continuous mode. This mode of operation allows for a quiet and, therefore, lower power operation.

3.9.2 Data Frames

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

- **Sample Phase.** During this phase, the SERIRQ device drives SERIRQ low if the corresponding interrupt signal is low. If the corresponding interrupt is high, the SERIRQ devices tri-state the SERIRQ signal. The SERIRQ line remains high due to pull-up resistors. A low level during the IRQ0-1 and IRQ2-15 frames indicates that an active-high ISA interrupt is not being requested, but a low level during the PCI INT[A:D], SMI#, and IOCHK# frame indicates that an active-low interrupt is being requested.
- **Recovery Phase.** During this phase, the device drives the SERIRQ line high if in the Sample Phase it was driven low. If it was not driven in the sample phase, it is tri-stated in this phase.
- **Turn-around Phase.** The device will tri-state the SERIRQ line.

3.9.3 Stop Frame

After all data frames, a Stop Frame is driven by 82801E C-ICH. The SERIRQ signal is driven low by 82801E C-ICH for two or three PCI clocks. The number of clocks is determined by the SERIRQ configuration register. The number of clocks determines the next mode.

Table 31. Stop Frame Explanation

Stop Frame Width	Next Mode
Two PCI clocks	**Quiet Mode.** Any SERIRQ device may initiate a Start Frame
Three PCI clocks	**Continuous Mode.** Only the host (82801E C-ICH) may initiate a Start Frame

3.9.4 Specific Interrupts Not Supported via SERIRQ

There are three interrupts seen through the serial stream that are not supported by the 82801E C-ICH. These interrupts are generated internally and are not sharable with other devices within the system. These interrupts are:

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

82801E C-ICH ignores the state of these interrupts in the serial stream, and does not adjust their level based on the level seen in the serial stream. In addition, the interrupts IRQ14 and IRQ15 from the serial stream are treated differently than their ISA counterparts. These two frames are not passed to the Bus Master IDE logic. The Bus Master IDE logic expects IDE to be behind the 82801E C-ICH.

3.9.5 Data Frame Format

Table 32 shows the format of the data frames. For the PCI interrupts (A-D), the output from the 82801E C-ICH is ANDed with the PCI input signal. Thus, the interrupt can be signaled via both the PCI interrupt input signal and via the SERIRQ signal (they are shared).

Table 32. Data Frame Format

Data Frame #	Interrupt	Clocks Past Start Frame	Comment
1	IRQ0	2	Ignored. IRQ0 can only be generated via the internal 8524
2	IRQ1	5	Before port 60h latch
3	SMI#	8	Causes SMI# if low. Sets the SERIRQ_SMI_STS bit
4	IRQ3	11	
5	IRQ4	14	
6	IRQ5	17	
7	IRQ6	20	
8	IRQ7	23	
9	IRQ8	26	Ignored. IRQ8# can only be generated internally or on ISA
10	IRQ9	29	
11	IRQ10	32	
12	IRQ11	35	
13	IRQ12	38	Before port 60h latch
14	IRQ13	41	Ignored. IRQ13 can only be generated from FERR#
15	IRQ14	44	Do not include in BM IDE interrupt logic
16	IRQ15	47	Do not include in BM IDE interrupt logic
17	IOCHCK#	50	Same as ISA IOCHCK# going active
18	PCI INTA#	53	Drive PIRQA#
19	PCI INTB#	56	Drive PIRQB#
20	PCI INTC#	59	Drive PIRQC#
21	PCI INTD#	62	Drive PIRQD#

3.10 Real Time Clock (D31:F0)

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

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

Note: The 82801E C-ICH supports the ability to generate an SMI# based on century rollover. See "Century Rollover" on page 93 for more information on the century rollover.

The 82801E C-ICH does not implement month/year alarms.

3.10.1 Update Cycles

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

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

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

3.10.2 Interrupts

The real-time clock interrupt is internally routed within the 82801E C-ICH both to the I/O APIC and the 8259. It is mapped to interrupt vector 8. This interrupt does not leave the 82801E C-ICH, nor is it shared with any other interrupt. IRQ8# from the SERIRQ stream is ignored.

3.10.3 Lockable RAM Ranges

The RTC's battery-backed RAM supports two 8-byte ranges that can be locked via the configuration space. If the locking bits are set, the corresponding range in the RAM are not readable or writable. A write cycle to those locations has no effect. A read cycle to those locations does not return the location's actual value (may be all 0s or all 1s).

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

3.10.4 Century Rollover

82801E C-ICH detects a rollover when the Year byte (RTC I/O space, index offset 09h) transitions form 99 to 00. Upon detecting the rollover, the 82801E C-ICH sets the NEWCENTURY_STS bit (TCOBASE + 04h, bit 7).

3.10.5 Clearing Battery-Backed RTC RAM

Clearing CMOS RAM in an 82801E C-ICH-based platform can be done by using a jumper on RTCRST# or GPI or using the SAFEMODE strap. Implementations should not attempt to clear CMOS by using a jumper to pull VCCRTC low.

Using RTCRST# to clear CMOS

A jumper on RTCRST# can be used to clear CMOS values, and to reset to the default the state of the configuration bits that reside in the RTC power well. When the RTCRST# is strapped to ground, the RTC_PWR_STS bit (D31:F0:A4h bit 2) is set and the configuration bits in the RTC power well are set to their default state. BIOS can monitor the state of this bit and manually clear the RTC CMOS array once the system is booted. The normal position would cause RTCRST# to be pulled up through a weak pull-up resistor. Table 33 shows the bits that are set to their default state when RTCRST# is asserted.

RTCRST# should be used to reset configuration bits (and signal BIOS to clear CMOS) ONLY in a G3 state. Additionally, 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 test mode. Assertion for any other number of clocks may put the component into an indeterminate state, which is not supported.

Table 33. Configuration Bits Reset By RTCRST# Assertion

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

Using a GPI to Clear CMOS

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

Using the SAFEMODE Strap to Clear CMOS

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

Note: Both the GPI and SAFEMODE strap techniques to clear CMOS require multiple steps to implement. The system is booted with the jumper in a new position, then powered back down. The jumper is replaced back to the normal position, then the system is rebooted again. The RTCRST# jumper technique allows the jumper to be moved and then replaced, all while the system is powered off. Once the system is booted, the RTC_PWR_STS can be detected in the set state.

Note: Clearing CMOS, using a jumper on VCCRTC, must NOT be implemented.

3.11 Processor Interface (D31:F0)

The 82801E C-ICH interfaces to the processor with a variety of signals:

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

Most 82801E C-ICH outputs to the processor use standard buffers. The 82801E C-ICH has a separate Vcc signal that is pulled up at the system level to the processor voltage and thus, determines Voh for the outputs to the processor. Note that this is different than previous generations of chips that have used open-drain outputs. This new method saves up to 12 external pull-up resistors.

The 82801E C-ICH also handles the speed setting for the processor by holding specific signals at certain states just prior to CPURST going inactive. This removes the need for the external logic that is often required with other chipsets.

The 82801E C-ICH does not support the processor's FRC mode.

3.11.1 Processor Interface Signals

This section describes each of the signals that interface between the 82801E C-ICH and the processor. Note that the behavior of some signals may vary during processor reset, because the signals are used for frequency strapping.

3.11.1.1 A20M#

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

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

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

3.11.1.2 INIT#

The INIT# signal is active (driven low) based on any one of several events described in Table 34. When any of these events occur, INIT# is driven low for 16 PCI clocks, then driven high.

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

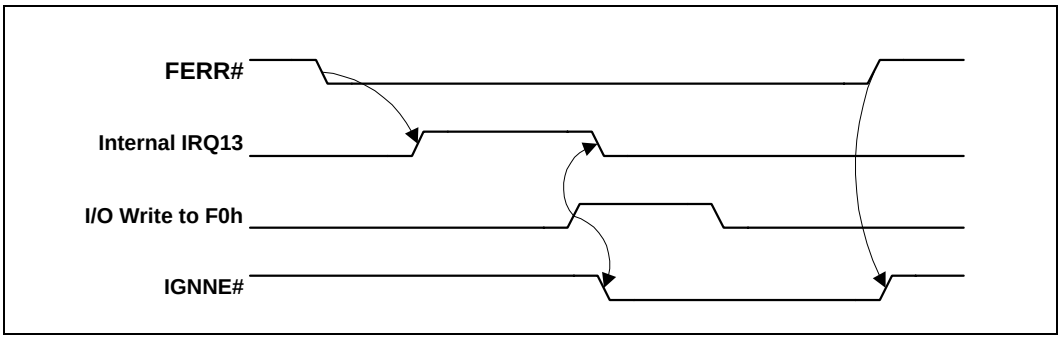
Table 34. INIT# Going Active

Cause of INIT# Going Active	Comment
Shutdown special cycle from the processor.	
PORT92 write, where INIT_NOW (bit 0) transitions from a 0 to a 1.	
PORTCF9 write, where RST_CPU (bit 2) was a 0 and SYS_RST(bit 1) transitions from 0 to 1.	
RCIN# input signal goes low. RCIN# is expected to be driven by the external microcontroller (KBC).	0 to 1 transition on RCIN# must occur before the 82801E C-ICH arms INIT# to be generated again.
Processor BIST	To enter BIST, the software sets CPU_BIST_EN bit and then does a full processor reset using the CF9 register.

3.11.1.3 FERR#/IGNNE# (Coprocessor Error)

The 82801E C-ICH supports the coprocessor error function with the FERR#/IGNNE# pins. The function is enabled via the COPROC_ERR_EN bit (Device 31:Function 0, Offset D0, bit 13). FERR# is tied directly to the Coprocessor Error signal of the processor. If FERR# is driven active by the processor, IRQ13 goes active (internally). When it detects a write to the COPROC_ERR register, the 82801E C-ICH negates the internal IRQ13 and drives IGNNE# active. IGNNE# remains active until FERR# is driven inactive. IGNNE# is never driven active unless FERR# is active.

Figure 15. Coprocessor Error Timing Diagram



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

3.11.1.4 NMI

Non-Maskable Interrupts (NMIs) can be generated by several sources, as described in Table 35.

Table 35. NMI Sources

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

3.11.1.5 STPCLK# and CPUSLP# Signals

The 82801E C-ICH power management logic controls these active-low signals. Refer to “Power Management (D31:F0)” on page 98 for more information on the functionality of these signals.

3.11.1.6 CPUPWRGOOD Signal

This signal is connected to the processor’s PWRGOOD input. This is an open-drain output signal (external pull-up resistor required) that represents a logical AND of the 82801E C-ICH’s PWROK and VRMPWRGD signals.

3.11.2 Speed Strapping for Processor

The 82801E C-ICH sets the speed straps for the processor; this removes the need for external logic that was needed with prior PCIsets. Refer to the processor specification for the speed strapping definition. The 82801E C-ICH performs the following to set the speed straps for the processor:

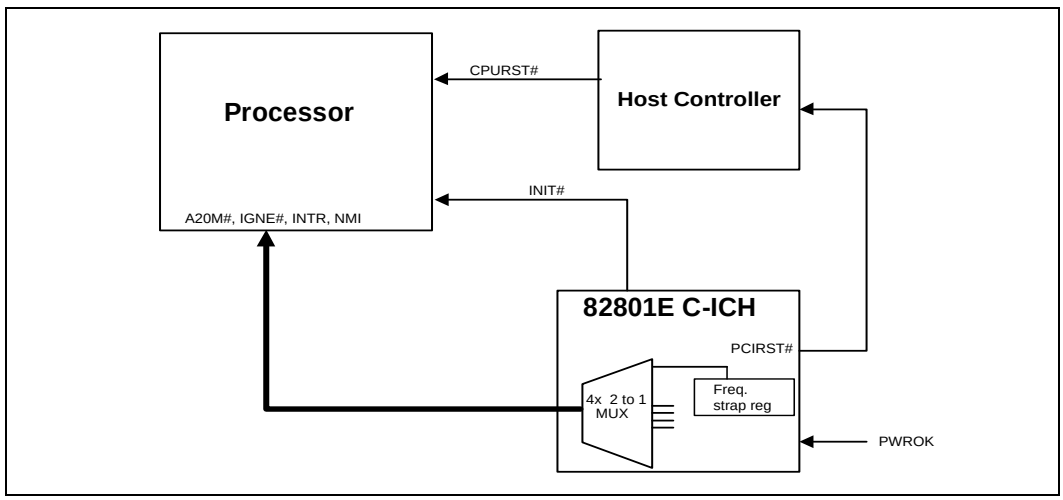
1. While PWROK is active, the 82801E C-ICH drives A20M#, IGNNE#, NMI, and INTR high.
2. As soon as PWROK goes active, the 82801E C-ICH reads the FREQ_STRAP field contents.
3. Based on PWROK going active, the 82801E C-ICH deasserts PCIRST#, and based on the value of the FREQ_STRAP field (D31:F0,Offset D4), the 82801E C-ICH drives the intended core frequency values on A20M#, IGNNE#, NMI, and INTR. The 82801E C-ICH holds these signals for 120 ns after CPURST# is deasserted by the Host controller.

Table 36. Frequency Strap Bit Mapping

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

NOTE: The FREQ_STRAP register is in the RTC well. The value in the register can be forced to 1111 via a pinstrap (AC_SDOUT signal), or the 82801E C-ICH can automatically force the speed strapping to 1111 if the processor fails to boot.

Figure 16. Signal Strapping



3.12 Power Management (D31:F0)

The 82801E C-ICH is an “on” or “off” device; it does not support power management features or sleep states. The following functions, mechanisms and signals are not supported.

- Wake Events
- Suspend Power Planes
- Sleep Lines/States s1, s2, s3, s4, s5
- SUS_STAT#
- PME#
- SLP_S3#
- SLP_S5#
- PWRBTN# (Soft Off/g2, s5)

Note: Care must be taken to properly set and disable Power Management Status/Enable and control registers in the OS and BIOS.

Features

— Software initiated throttling of processor performance for Thermal and Power Reduction

— Hardware Override to throttle processor performance if the system is too hot

— SCI and SMI# Generation

Warning: This section is not part of the 82801E C-ICH behavioral specification. It is intended to identify key areas and steps taken by the BIOS to make sure there are no conflicts with the functional definition.

3.12.1 82801E C-ICH and System Power States

Table 37 shows the power states defined for 82801E C-ICH-based platforms. The state names generally match the corresponding ACPI states.

Table 37. General Power States for Systems Using 82801E C-ICH

State/ Substates	Legacy Name/Description
G0	**Full On:** Processor operating. Individual devices may be shut down to save power.
G0/C1	**Auto-Halt:** The processor has executed an AutoHalt instruction and is not executing code. The processor snoops the bus and maintains cache coherency.
G3	**Mechanical OFF (MOFF):** System context not maintained. All power is shut off except for the RTC. No “Wake” events are possible, because the system does not have any power. This state occurs if the user removes the batteries, turns off a mechanical switch, or if the system power supply is at a level that is insufficient to power the “waking” logic. When system power returns, transition depends on the state just prior to the entry to G3 and the AFTERG3 bit in the GEN_PMCON3 register (D31:F0, offset A4).

Table 38. State Transition Rules for 82801E C-ICH

Present State	Transition Trigger	Next State
G0	• Processor halt instruction • Mechanical Off/Power Failure	• G0/C1 • G3
G0/C1	• Any Enabled Break Event • Power Failure	• G0/C0 • G3
G3	• Power Returns	

3.12.2 82801E C-ICH Power Planes

The 82801E C-ICH power planes are fully defined in the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Datasheet* (order number 273598).

Although not specific power planes within the 82801E C-ICH, there are many interface signals that go to devices that may be powered down. These include:

- IDE: 82801E C-ICH can tri-state or drive low all IDE output signals and shut off input buffers.
- USB: 82801E C-ICH can tri-state USB output signals and shut off input buffers if USB wakeup is not desired

3.12.3 SMI#/SCI Generation

When any SMI# event takes place, the 82801E C-ICH asserts SMI# to the processor, which causes it to enter SMM space. SMI# remains active until the EOS bit is set. When the EOS bit is set, SMI# goes inactive for a minimum of four PCICLKs. If another SMI event occurs, SMI# is driven active again.

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

In systems using the APIC, the SCI can still be routed to IRQ[9, 10, or 11] or it can be instead routed to one of the APIC interrupts 20:23. In the case where the SCI is routed to IRQ[20, 21, 22, or 23], the interrupt generated internally is an active low level. The interrupt remains low until all SCI sources are removed. In the case where the SCI is routed to IRQ[9, 10, or 11], the interrupt generated internally is active high. The interrupt remains high until all SCI sources are removed.

Table 39 shows which events can cause an SMI# and SCI. Note that some events can be programmed to cause either an SMI# or SCI. The usage of the event for SCI (instead of SMI#) is typically associated with an ACPI-based system. Each SMI# or SCI source has a corresponding enable and status bits.

Warning: This section is not to be considered as part of the 82801E C-ICH behavioral specification. It is intended to identify key areas and steps taken by the BIOS to make sure there are no conflicts with the functional definition.

Table 39. Causes of SMI# and SCI (Sheet 1 of 2)

Cause	SCI	SMI	Additional Enables	Where Reported	Comment
RTC Alarm	Yes	Yes	RTC_EN=1	RTC_STS	
Ring Indicate	Yes	Yes	RI_EN=1	RI_STS	
USB#1	Yes	Yes	USB1_EN=1	USB1_STS	
THRM# pin active	Yes	Yes	THRM_EN=1	THRM_STS	The THRM# can cause an SMI# or SCI on either the rising or falling edge. Causes SCI if SCI_EN is set, causes SMI# if SCI_EN not set.
Any GPI	Yes	Yes	GPI[x]_Route=10 (SCI) GPI[x]_Route=01 (SMI) GPE1[x]_EN=1	GPI[x]_STS GPE1_STS	
TCO SCI Logic	Yes	No	TCOSCI_EN=1	TCOSCI_STS	Can also cause IRQ (other than SCI).
TCO SCI message from MCH	Yes	No	none	MCHSCI_STS	
TCO SMI Logic	No	Yes	TCO_EN=1	TCO_STS	Can also cause IRQ (other than SCI).
TCO SMI: Century Rollover	No	Yes	none	NEWCENTURY_STS	
TCO SMI: TCO TIMEROUT	No	Yes	none	TIMEOUT	
TCO SMI: OS writes to TCO_DAT_IN register	No	Yes	none	OS_TCO_SMI	
TCO SMI: Message from MCH	No	Yes	none	MCHSMI_STS	
TCO SMI: NMI occurred (and NMIs mapped to SMI)	No	Yes	NMI2SMI_EN=1	NMI2SMI_STS	
TCO SMI: INTRUDER# signal goes active	No	Yes	INTRD_SEL=10	INTRD_DET	
TCO SMI: Change of the BIOSWP bit from 0 to 1	No	Yes	BLD=1	BIOSWR_STS	
TCO SMI: Write attempted to BIOS	No	Yes	BIOSWP=1	BIOSWR_STS	
BIOS_RLS written to	Yes	No	GBL_EN=1	GBL_STS	ACPI code in OS sets GBL_RLS bit to cause BIOS_STS bit active, which causes SMI#.
GBL_RLS written to	No	Yes	BIOS_EN=1	BIOS_STS	This bit is set when the BIOS sets the BIOS_RLS bit. The ACPI handler will clear the bit.

NOTES:
1. SCI_EN must be 1 to enable SCI. SCI_EN must be 0 to enable SMI.
2. SCI can be routed to cause interrupt 9:11 or 20:23 (20:23 only available in APIC mode).
3. GBL_SMI_EN must be 1 to enable SMI.
4. EOS must be written to 1 to re-enable SMI for the next one.
5. The GPI[x]_ Route bits can enable GPIs to generate SMIs regardless of the state of SMI_EN.

Table 39. Causes of SMI# and SCI (Sheet 2 of 2)

Cause	SCI	SMI	Additional Enables	Where Reported	Comment
Write to B2h register	No	Yes	none	APM_STS	OS or BIOS writes to the APMC register. SMM handler clears.
Periodic timer expires	No	Yes	PERIODIC_EN=1	PERIODIC_STS	
64 ms timer expires	No	Yes	SWSMI_TMR_EN=1	SWSMI_TMR_STS	Allows SMM handler to exit temporarily. Another SMI# occurs about 64 ms later.
Legacy USB logic	No	Yes	LEGACY_USB_EN=1	LEGACY_USB_STS	Bit set based on address decode or incoming USB IRQ.
Serial IRQ SMI reported	No	Yes	none	SERIRQ_SMI_STS	
Device Trap: Device monitors match address in its range	No	Yes	DEV[n]_TRAP_EN=1	DEVMON_STS, DEV[n]_TRAP_STS	Indicates that subsystems may need to be powered back on.
SMBus Host Controller	No	Yes	SMB_SMI_EN	SMBus host status reg.	
SMBus Slave SMI	No	Yes	none	SMBUS_SMI_STS	
Access microcontroller 62h/66h	No	Yes	MCSMI_EN	MCSMI_STS	Access to Microcontroller range (62h/66h) with MCSMI_EN set.
SLP_EN bit written to 1	No	Yes	SMI_ON_SLP_EN=1	SMI_ON_SLP_EN_STS	

NOTES:
1. SCI_EN must be 1 to enable SCI. SCI_EN must be 0 to enable SMI.
2. SCI can be routed to cause interrupt 9:11 or 20:23 (20:23 only available in APIC mode).
3. GBL_SMI_EN must be 1 to enable SMI.
4. EOS must be written to 1 to re-enable SMI for the next one.
5. The GPI[x]_ Route bits can enable GPIs to generate SMIs regardless of the state of SMI_EN.

3.12.4 Dynamic Processor Clock Control

82801E C-ICH has extensive control for dynamically starting and stopping system clocks. The clock control is used for processor throttling. Each dynamic clock control method is described in this section.

The 82801E C-ICH supports the ACPI C0 and C1.

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

- STPCLK#: Used to halt processor instruction stream.

In the C0 processor power state, the processor executes instructions.

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

Note: CPUs are free to perform their own dynamic clock controls. However, this is done without any coordination by the 82801E C-ICH.

Due to a break event, the C1 state ends. Based on the break event, the 82801E C-ICH returns the system to a C0 state.

3.12.4.1 Throttling Using STPCLK#

Throttling is used to lower power consumption or to reduce heat. The 82801E C-ICH asserts STPCLK# to throttle the processor clock and the processor appears to temporarily enter a C2 state. After a programmable time elapses, the 82801E C-ICH deasserts STPCLK# and the processor appears to return to the C0 state. This allows the processor to operate at reduced average power, with a corresponding decrease in performance. Two methods are included to start throttling:

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

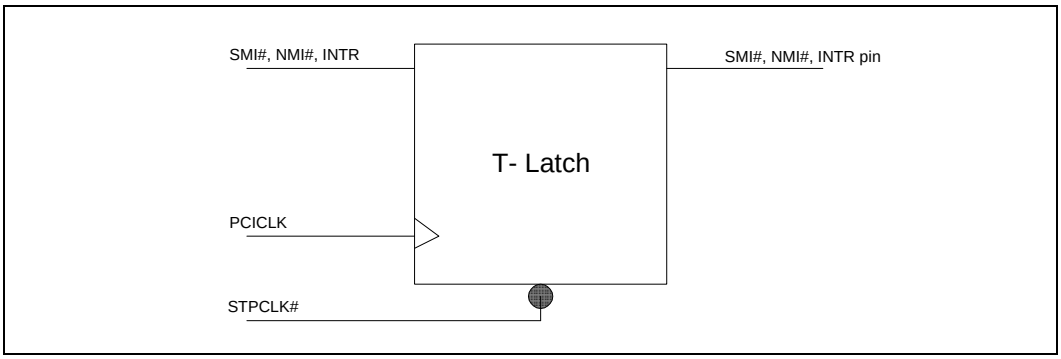
Throttling due to the THRM# signal has higher priority than the software-initiated throttling.

3.12.5 STPCLK# Implementation Notes

The CPU treats STPCLK# like an interrupt and recognizes it on instruction boundaries (no INTA cycles are run). When it recognizes STPCLK# active, the CPU stops execution on the next instruction boundary, stops the pre-fetch unit, empties internal pipelines and write buffers, and generates a Stop-Grant bus cycle before entering the Stop Grant or Quickstart state. The CPU may stop the clock to most of its internal modules, and no instructions are executed. The CPU exits the Stop Grant or Quickstart state when it is reset, or upon sampling STPCLK# inactive.

The CPU will latch transitions on the external interrupt signals (SMI#, NMI, INTR, and INIT#) while in Stop Grant state. These interrupts are taken after the de-assertion of STPCLK#. (Please see the implementation notes below.)

Figure 17. Latching CPU I/F Signals with STOPCLK#



Other Implementation Notes:

- The state machine must insure that the STPCLK# signal stays high for a minimum period of time. If STPCLK# is to go low due to throttling (regular or due to the THRM# signal), this could be very soon after it was driven high. If the Memory Controller Hub (MCH) guarantees that the STPGNT# cycle coming down the hub link occurs after the BRDY# is seen by the processor, then no special care is needed in the 82801E C-ICH. However, if the MCH cannot

guarantee the delay from Stop-Grant to BRDY#, then this needs to be guaranteed in the 82801E C-ICH.

- Exception: For SMI#s that are caused by a CPU I/O cycle, if STPCLK# is active, the 82801E C-ICH will still drive SMI# active. This is because the STPCLK# was obviously too late to be recognized at the instruction boundary. The I/O cycles that can cause SMI# include: writes to the APM register (B2h), accesses to 60/64h when "Legacy USB KBC scheme" is used, traps for Monitors 4, 5, 6, and 7, the SMI# on SLP_EN bit, accesses to 62/66h when the MCSMI_EN bit is set, access to registers with their associated enable set in the DEVTRAP_EN register, and the BIOS_STS bit (which is set by the CPU writing a 1 to the GLB_RLS bit when the BIOS_EN bit is also set).

3.12.5.1 Transition Rules Among S0/Cx and Throttling States

- The Host controller must post Stop-Grant cycles so that the processor gets an indication of the end of the special cycle prior to the 82801E C-ICH observing the Stop-Grant cycle. This ensures that the STPCLK# signals stays active for a sufficient period after the processor observes the response phase.

3.12.6 Thermal Management

The 82801E C-ICH has mechanisms to assist with managing thermal problems in the system.

3.12.6.1 THRM# Signal for SMI# or SCI

The THRM# signal is used as a status input for a thermal sensor. Based on the THRM# signal going active, the 82801E C-ICH generates an SMI# or SCI (depending on SCI_EN).

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

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

By setting the THRM_POL bit, another SMI# or SCI can optionally be generated when the THRM# signal goes back high. This allows the software (BIOS) to turn off the cooling methods.

3.12.6.2 THRM# Initiated Passive Cooling

If the THRM# signal remains active for some time longer than two seconds and the 82801E C-ICH is in the S0/G0/C0 state, then the 82801E C-ICH enters an auto-throttling mode, in which it provides a duty cycle on the STPCLK# signal. This will reduce the overall power consumption by the system, and should cool the system. The intended result of the cooling is that the THRM# signal should go back inactive.

For all programmed values (001–111), THRM# going active will result in STPCLK# active for a minimum time of 12.5% and a maximum of 87.5%. The period is 1024 PCI clocks. Thus, the STPCLK# signal can be active for as little as 128 PCI clocks or as much as 896 PCI clocks. The actual slowdown (and cooling) of the processor will depend on the instruction stream, because the processor is allowed to finish the current instruction. Furthermore, the 82801E C-ICH waits for the STOP-GRANT cycle before starting the count of the time the STPCLK# signal is active.

When THRM# goes inactive, throttling stops. When the 82801E C-ICH is already attempting throttling because the THTL_EN bit is set, the duty cycle associated with the THRM# signal will have higher priority.

3.12.6.3 THRM# Override Software Bit

The FORCE_THTL bit allows BIOS to force passive cooling, just as if the THRM# signal had been active for 2 seconds. If this bit is set, the 82801E C-ICH starts throttling using the ratio in the THRM_DTY field.

When this bit is cleared, the 82801E C-ICH stops throttling, unless the THRM# signal has been active for two seconds or if the THTL_EN bit is set (indicating that ACPI software is attempting throttling).

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

Using the THTL_EN and THTL_DTY bits, the 82801E C-ICH can force a programmed duty cycle on the STPCLK# signal. This reduces the effective instruction rate of the processor and cut its power consumption and heat generation.

3.12.6.5 Active Cooling

Active cooling typically involves fans. The GPIO signals from the 82801E C-ICH can be used to turn a fan on or off.

3.12.7 Event Input Signals and Their Usage

The 82801E C-ICH has various input signals that trigger specific events. This section describes those signals and how they should be used.

3.12.7.1 RI# — Ring Indicate

The 82801E C-ICH generates an interrupt based on RI# active and the interrupt is set up as a break event. Note that there is no filtering on the RI# signal. Any debounce filtering must be done externally.

3.12.8 Alt Access Mode

Table 40. Transitions Due to RI# Signal

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

Before entering a low power state, several registers from powered down portions of the 82801E C-ICH may need to be saved. In the majority of cases, this is not an issue, since registers have read and write paths. However, several of the ISA compatible registers are either read only or write only. To get data out of write-only registers and to restore data into read-only registers, the 82801E C-ICH implements an alternate access mode.

3.12.8.1 Write Only Registers with Read Paths in Alternate Access Mode

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

Table 41. Write Only Registers with Read Paths in Alternate Access Mode (Sheet 1 of 2)

Restore Data				Restore Data			
I/O Addr	# of Rds	Access	Data	I/O Addr	# of Rds	Access	Data
00h	2	1	DMA Chan 0 base address low byte	40h	7	1	Timer Counter 0 status, bits [5:0]
		2	DMA Chan 0 base address high byte			2	Timer Counter 0 base count low byte
01h	2	1	DMA Chan 0 base count low byte			3	Timer Counter 0 base count high byte
		2	DMA Chan 0 base count high byte			4	Timer Counter 1 base count low byte
02h	2	1	DMA Chan 1 base address low byte			5	Timer Counter 1 base count high byte
		2	DMA Chan 1 base address high byte			6	Timer Counter 2 base count low byte
03h	2	1	DMA Chan 1 base count low byte			7	Timer Counter 2 base count high byte
		2	DMA Chan 1 base count high byte	41h	1		Timer Counter 1 status, bits [5:0]
04h	2	1	DMA Chan 2 base address low byte	42h	1		Timer Counter 2 status, bits [5:0]
		2	DMA Chan 2 base address high byte	70h	1		Bit 7 = NMI Enable, Bits [6:0] = RTC Address
05h	2	1	DMA Chan 2 base count low byte	C4h	2	1	DMA Chan 5 base address low byte
		2	DMA Chan 2 base count high byte			2	DMA Chan 5 base address high byte
06h	2	1	DMA Chan 3 base address low byte	C6h	2	1	DMA Chan 5 base count low byte
		2	DMA Chan 3 base address high byte			2	DMA Chan 5 base count high byte
07h	2	1	DMA Chan 3 base count low byte	C8h	2	1	DMA Chan 6 base address low byte
		2	DMA Chan 3 base count high byte			2	DMA Chan 6 base address high byte

Table 41. Write Only Registers with Read Paths in Alternate Access Mode (Sheet 2 of 2)

Restore Data				Restore Data			
I/O Addr	**# of Rds**	**Access**	**Data**	**I/O Addr**	**# of Rds**	**Access**	**Data**
08h	6	1	DMA Chan 0–3 Command[2]	CAh	2	1	DMA Chan 6 base count low byte
		2	DMA Chan 0–3 Request			2	DMA Chan 6 base count high byte
		3	DMA Chan 0 Mode: Bits(1:0) = “00”	CCh	2	1	DMA Chan 7 base address low byte
		4	DMA Chan 1 Mode: Bits(1:0) = “01”			2	DMA Chan 7 base address high byte
		5	DMA Chan 2 Mode: Bits(1:0) = “10”	CEh	2	1	DMA Chan 7 base count low byte
		6	DMA Chan 3 Mode: Bits(1:0) = “11”.			2	DMA Chan 7 base count high byte
20h	12	1	PIC ICW2 of Master controller	D0h	6	1	DMA Chan 4–7 Command[2]
		2	PIC ICW3 of Master controller			2	DMA Chan 4–7 Request
		3	PIC ICW4 of Master controller			3	DMA Chan 4 Mode: Bits(1:0) = “00”
		4	PIC OCW1 of Master controller[1]			4	DMA Chan 5 Mode: Bits(1:0) = “01”
		5	PIC OCW2 of Master controller			5	DMA Chan 6 Mode: Bits(1:0) = “10”
		6	PIC OCW3 of Master controller			6	DMA Chan 7 Mode: Bits(1:0) = “11”.
		7	PIC ICW2 of Slave controller				
		8	PIC ICW3 of Slave controller				
		9	PIC ICW4 of Slave controller				
		10	PIC OCW1 of Slave controller[1]				
		11	PIC OCW2 of Slave controller				
		12	PIC OCW3 of Slave controller				

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

3.12.8.2 PIC Reserved Bits

Many bits within the PIC are reserved, and must have certain values written for the PIC to operate properly. Therefore, there is no need to return these values in alternate access mode. When reading PIC registers from 20h and A0h, the reserved bits return the values listed in Table 42.

Table 42. PIC Reserved Bits Return Values

PIC Reserved Bits	Value Returned
ICW2(2:0)	000
ICW4(7:5)	000
ICW4(3:2)	00
ICW4(0)	0
OCW2(4:3)	00
OCW3(7)	0
OCW3(5)	Reflects bit 6
OCW3(4:3)	01

3.12.8.3 Read Only Registers with Write Paths in Alternate Access Mode

The registers described in Table 43 have write paths in alternate access mode. Software restores these values after returning from a powered down state. These registers must be handled specially by software. When in normal mode, writing to the Base Address and Count Register also writes to the Current Address and Count Register. Therefore, the Base Address and Count must be written first, then the part is put into alternate access mode and the Current Address and Count Register is written.

Table 43. Register Write Accesses in Alternate Access Mode

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

PWROK Signal

The PWROK input should go active based on the core supply voltages becoming valid. PWROK should go active no sooner than 10 ms after VCC3_3 and VCC1_8 have reached their nominal values.

Note: Traditional designs have a reset button logically ANDed with the PWROK signal from the power supply and the processor's voltage regulator module. If this is done with the 82801E C-ICH, the PWROK_FLR bit will be set. The 82801E C-ICH treats this internally as if the RSMRST# signal had gone active. However, it is not treated as a full power failure. If PWROK goes inactive and then active (but RSMRST# stays high), the 82801E C-ICH reboots (regardless of the state of the

AFTERG3 bit). If the RSMRST# signal also goes low before PWROK goes high, this is a full power failure and the reboot policy is controlled by the AFTERG3 bit.

VRMPWRGD Signal

This signal is connected to the processor's VRM and is internally ANDed with the PWROK signal that comes from the system power supply. This removes the need for the external AND gate found in some systems.

3.12.9 Clock Generators

The clock generator is expected to provide the frequencies shown in Table 44.

Table 44. Intel® 82801E C-ICH Clock Inputs

Clock Domain	Frequency	Source	Usage
CLK66	66 MHz	Main Clock Generator	Should be running in all Cx states.
PCICLK	33 MHz	Main Clock Generator	Free-running PCI Clock to 82801E C-ICH.
CLK48	48 MHz	Main Clock Generator	Used by USB Controllers.
CLK14	14.318 MHz	Main Clock Generator	Used by ACPI timers.
APICCLK	16.67 MHz or 33 MHz	Main Clock Generator	Used for 82801E C-ICH-processor interrupt messages.
LAN0_CLK LAN1_CLK	0.8 to 50 MHz	LAN Connect	LAN Connect link. Control policy is determined by the clock source.

3.12.10 Legacy Power Management Theory of Operation

Warning: This section is not part of the 82801E C-ICH behavioral specification. It is intended to identify key steps taken by the BIOS to make sure there are no conflicts with the functional definition.

Instead of relying on ACPI software, legacy power management uses BIOS and various hardware mechanisms. 82801E C-ICH has a greatly simplified method for legacy power management compared to previous generations (e.g., PIIX4).

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

However, the operating system is assumed to be at least APM enabled. Without APM calls, there is no quick way to know when the system is idle between keystrokes. The 82801E C-ICH does not support the burst modes found in previous components (e.g., PIIX4).

3.12.10.1 Advanced Power Management (APM)

Warning: This section is not part of the 82801E C-ICH behavioral specification. It is intended to identify key steps taken by the BIOS to make sure there are no conflicts with the functional definition.

The 82801E C-ICH has a timer that generates an SMI# once per minute when enabled by the 1MIN_EN bit in the SMI Control and Enable Register. The SMI handler can check for system activity by reading the DEVACT_STS register. If none of the system bits are set, the SMI handler can increment a software counter. When the counter reaches a sufficient number of consecutive minutes with no activity, the SMI handler can then put the system into a lower power state.

If there is activity, various bits in the DEVACT_STS register are set. Software clears the bits by writing a 1 to the bit position.

The DEVACT_STS Register allows for monitoring various internal devices, or Super I/O devices (SP, PP, FDC) on LPC or PCI, keyboard controller accesses, or audio functions on LPC or PCI. Other PCI activity can be monitored by checking the PCI interrupts.

3.13 System Management (D31:F0)

The 82801E C-ICH provides various functions to make a system easier to manage and to lower the Total Cost of Ownership (TCO) of the system. Features and functions can be augmented via external A/D converters and GPIO, as well as an external microcontroller. The following features and functions are supported by the 82801E C-ICH:

- Processor present detection.
  - — Detects if processor fails to fetch the first instruction after reset.
- Various Error detection (e.g., ECC Errors) indicated by Host Controller
  - — Can generate SMI#, SCI, SERR, NMI, or TCO interrupt
- Intruder Detect input
  - — Can generate TCO interrupt or SMI# when the system cover is removed.
  - — INTRUDER# allowed to go active in any power state, including G3.
- Detection of bad FWH programming
  - — Detects if data on first read is FFh (indicates unprogrammed Firmware Hub (FWH))

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

3.13.1 Theory of Operation

The System Management functions are designed to allow the system to diagnose failing subsystems. The intent of this logic is that some of the system management functionality be provided without the aid of an external microcontroller.

Detecting a System Lockup

When the processor is reset, it is expected to fetch its first instruction. If the processor fails to fetch the first instruction after reset, the TCO timer times out twice and the 82801E C-ICH asserts PCIRST#.

Handling an Intruder

The 82801E C-ICH has an input signal (INTRUDER#) that can be attached to a switch that is activated when the system's case is opened. This input has a two RTC clock debounce. If INTRUDER# goes active (after the debouncer), this will set the INTRD_DET bit in the TCO_STS register. The INTRD_SEL bits in the TCO_CNT register can enable the 82801E C-ICH to cause an SMI# or interrupt. The BIOS or interrupt handler can then cause a transition to the S5 state by writing to the SLP_EN bit.

The software can also directly read the status of the INTRUDER# signal (high or low) by clearing and then reading the INTRD_DET bit. This allows the signal to be used as a GPI if the intruder function is not required.

Note: The INTRD_DET bit resides in the 82801E C-ICH's RTC well, and is set and cleared synchronously with the RTC clock. Thus, when software attempts to clear INTRD_DET (by writing a "1" to the bit location) there may be as much as two RTC clocks (about 65 µs) delay before the bit is actually cleared. Also, the INTRUDER# signal should be asserted for a minimum of 1 ms to guarantee that the INTRD_DET bit will be set.

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

Note: If the INTRUDER# signal goes inactive after the INTRD_DET bit is written to a 1, then INTRD_DET will go to a 0 when INTRUDER# input signal goes inactive.

Detecting Improper FWH Programming

The 82801E C-ICH can detect the case where the FWH is not programmed. This results in the first instruction fetched to have a value of FFh. If this occurs, the 82801E C-ICH sets the BAD_BIOS bit, which can then be reported via the Heartbeat and Event reporting using an external, Alert on LAN* enabled LAN Controller (See Section 3.13.2).

Handling an ECC Error or Other Memory Error

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

3.13.2 Alert on LAN*

Note: The 82801E C-ICH does not support sleep states and suspend mode; it is an "on" or "off" device. Wake-up events, Alert on LAN*, and incoming packets are still processed, since the 82801E C-ICH is always "awake" while the device is powered on.

The 82801E C-ICH integrated LAN controller supports Alert on LAN* functionality when used with the 82562EM Platform LAN Connect component. This allows the integrated LAN controller to report messages to a network management console without the aid of the system processor.

The 82801E C-ICH also features an independent, dedicated SMBus interface, referred to as the SMLINK interface that can be used with an external Alert on LAN* (or Alert on LAN 2*) enabled LAN Controller.

The basic scheme is for the 82801E C-ICH integrated LAN Controller to send a prepared Ethernet message to a network management console. The prepared message is stored in the non-volatile EEPROM that is connected to the 82801E C-ICH.

Messages are sent by the LAN Controller either because a specific event has occurred or they are sent periodically (also known as a heartbeat). The event and heartbeat messages have an identical format. The event messages are sent based on events occurring. The heartbeat messages are sent every 30 to 32 seconds. When an event occurs, the 82801E C-ICH sends a new message and increments the SEQ[3:0] field. For heartbeat messages, the sequence number does not increment.

If the policy is for the 82801E C-ICH to reboot the system after a hardware lockup, the 82801E C-ICH does not immediately send an Alert on LAN* message. It first attempts to reboot the processor and let the BIOS perform the appropriate recovery (and potentially send the message). However, if the boot fails, the 82801E C-ICH sends the message.

If the policy is for the 82801E C-ICH to not reboot after a hardware lockup, the 82801E C-ICH sends an Alert on LAN* message with the Watchdog (WD) Event Status bit set. This message is sent as soon as the lockup is detected. The message is sent with the next incremented sequence number. If a system is locked, the 82801E C-ICH continues sending the Alert on LAN* messages every heartbeat period (30–32 seconds) unless one of the following occurs:

- The system is suspended via a PowerButton Override.
- The NO_REBOOT bit (D31:F0, offset D4h, bit 1) is set and the system is reset using PWROK, or the system is reset remotely by SMLINK SMBus Slave write and BIOS clears the SECOND_TO_STS bit before a TCO timeout can occur.
- The NO_REBOOT bit (D31:F0, offset D4h, bit 1) is not set causing the system to reboot automatically.

If another event occurs prior to a power button override, the 82801E C-ICH will send another Alert on LAN* message with the next incremented sequence number and appropriate status bit set.

If a boot is unsuccessful (processor does not fetch the first instruction), then the 82801E C-ICH will send an Alert on LAN* message with the processor event status bit set and the next incremented sequence number. This message will be sent as soon as the lockup is detected (two TCO timer time-outs).

Note: Normally, the 82801E C-ICH does not send heartbeat messages while in the G0 state (except in the case of a lockup). However, if a hardware event (or heartbeat) occurs just as the system is transitioning into a G0 state, the hardware continues to send the message even though the system is in a G0 state (and the status bits may indicate this).

When used with an external Alert on LAN* enabled LAN controller, the 82801E C-ICH sends these messages via the SMLINK signals. When sending messages via these signals, the 82801E C-ICH abides by the SMBus rules associated with collision detection. It delays starting a message until the bus is idle and detects collisions. If a collision is detected, the 82801E C-ICH waits until the bus is idle and tries again. Table 45 shows the data included in the Alert on LAN* messages.

Table 45. Alert on LAN* Message Data

Field	Comment
Cover Tamper Status	1 = This bit will be set if the intruder detect bit is set (INTRD_DET).
Temp Event Status	1 = This bit will be set if the 82801E C-ICHTHERM# input signal is asserted.
Processor Missing Event Status	1 = This bit will be set if the processor failed to fetch the first instruction.
TCO Timer Event Status	1 = This bit is set when the TCO timer expires.
Software Event Status	1 = This bit is set when software writes a 1 to the SEND_NOW bit.
Unprogrammed FWH Event Status	1 = First BIOS fetch returned a value of FFh, indicating that the FWH has not yet been programmed (still erased).
GPIO Status	1 = This bit is set when GPIO[11] signal is high. 0 = This bit is cleared when GPIO[11] signal is low.
SEQ[3:0]	This is a sequence number. It will initially be 0, and will increment each time the 82801E C-ICH sends a new message. Upon reaching 1111, then the sequence number will roll over to 0000. MSB (SEQ3) sent first.
System Power State	00 = G0, 01 = G1, 10 = G2, 11 = Pre-Boot. MSB sent first.
MESSAGE1	Same as the MESSAGE1 Register. MSB sent first.
MESSAGE2	Same as the MESSAGE2 Register. MSB sent first.
WDSTATUS	Same as the WDSTATUS Register. MSB sent first.

3.13.3 IRQ1 and IRQ12 for Legacy Elimination

The new IRQ1 and IRQ12 sources are logically ANDed with the respective IRQ1 and IRQ12 that come from the SERIRQ logic. This is necessary because the SERIRQ logic will report IRQ1 and IRQ12 to be high (active), since there is no Super I/O to drive them low.

Note: In a system that does have a Super I/O, the new bits must be left at 1, since it is ANDed with the Super I/O's IRQ. **Don't attempt to write these bits to 0 in a system that has a keyboard controller (such as in a Super I/O).** It will not be validated, and is highly likely to break software.

The following algorithm assumes the byte is sent from the keyboard. The byte sent from the mouse is equivalent. The "setup" to the area of interest is left at a high level in this description. The area of interest is then described in more detail.

1. An SMI is received and discovered to be a USB interrupt.
2. The interrupt is discovered to be due to a TD associated with a keyboard device.
3. The data is analyzed and it is determined that the interrupt is due to a new key press.
4. The USB key-code is translated into the equivalent scan code set 2 (SS2) PS/2 scan code.
5. The result is queued on a queue of data to be sent from the keyboard to the system.
6. Other USB interrupts are handled.
7. The keyboard controller emulation code is executed at exit. It determines if the conditions are correct to return a byte to the system (e.g., emulated OBF indicates empty, keyboard interface not disabled, etc.). If not, the emulation exits awaiting the next event.
8. The queue of data to be sent from the keyboard to the system is found to contain a byte to be returned.

9. Given the typical keyboard controller configuration, it is translated from SS2 to SS1.

End of Setup.

10. The byte to be returned is stored in the emulated output buffer.
11. The emulation does the out to the port to enable IRQ1.
12. The emulation exits.

– time passes –

13. The system code services the interrupt and reads port 60h.
14. The UHCI traps the read and causes an SMI trap.
15. The trap is determined to be caused by the read from port 60h (TBY60R in LEGSUP).
16. The emulation code clears the interrupt register thus turning off IRQ1 and/or IRQ12.

Note: The emulation code can validly do this each time there is a read from 60h since that is what the keyboard controller would do as well. (There is no time where both IRQ 1 and IRQ 12 should be asserted simultaneously by either the keyboard controller or the emulated keyboard controller. This would be a violation of the keyboard controller/system protocol.)

3.14 General Purpose I/O

Power Wells

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

Some 82801E C-ICH GPIOs may be connected to pins on devices that exist in the core well. If these GPIOs are outputs, there is a danger that a loss of core power (PWROK low) event will result in the 82801E C-ICH driving a pin to a logic 1 to another device that is powered down.

GPIO[13:11,8:6,4:3,1] have “sticky” bits on the input. Refer to the GPE1_STS register. As long as the signal goes active for at least two clocks, the 82801E C-ICH will keep the sticky status bit active. The active level can be selected in the GP_LVL register.

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

SMI# and SCI Routing

The routing bits for GPIO[13:11,8:6,4:3,1:0] allow an input to be routed to SMI# or SCI, or neither. Note that a bit can be routed to either an SMI# or to an SCI, but not both.

3.15 IDE Controller (D31:F1)

The 82801E C-ICH IDE controller features two sets of interface signals (Primary and Secondary) that can be independently enabled, tri-stated or driven low.

The IDE interfaces of the 82801E C-ICH can support several types of data transfers:

- **Programmed I/O (PIO):** Processor is in control of the data transfer.
- **8237 Style DMA:** DMA protocol that resembles the DMA on the ISA bus, although it does not use the 8237 in the 82801E C-ICH. This protocol off loads the processor from moving data. This allows higher transfer rate of up to 16 Mbytes/s.
- **Ultra ATA/33:** DMA protocol that redefines signals on the IDE cable to allow both host and target throttling of data and transfer rates of up to 33 Mbytes/s.
- **Ultra ATA/66:** DMA protocol that redefines signals on the IDE cable to allow both host and target throttling of data and transfer rates of up to 66 Mbytes/s.
- **Ultra ATA/100:** DMA protocol that redefines signals on the IDE cable to allow both host and target throttling of data and transfer rates of up to 100 Mbytes/s.

3.15.1 PIO Transfers

The 82801E C-ICH IDE controller includes both compatible and fast timing modes. The fast timing modes can be enabled only for the IDE data ports. All other transactions to the IDE registers are run in single transaction mode with compatible timings.

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

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

PIO IDE Timing Modes

IDE data port transaction latency consists of startup latency, cycle latency, and shutdown latency:

- Startup latency is incurred when a PCI master cycle targeting the IDE data port is decoded and the DA[2:0] and CSxx# lines are not set up. Startup latency provides the setup time for the DA[2:0] and CSxx# lines prior to assertion of the read and write strobes (DIOR# and DIOW#).
- Cycle latency consists of the I/O command strobe assertion length and recovery time. Recovery time is provided so that transactions may occur back-to-back on the IDE interface (without incurring startup and shutdown latency) without violating minimum cycle periods for the IDE interface. The command strobe assertion width for the enhanced timing mode is selected by the IDETIM Register and may be set to 2, 3, 4, or 5 PCI clocks. The recovery time is selected by the IDETIM Register and may be set to 1, 2, 3, or 4 PCI clocks.

  If IORDY is asserted when the initial sample point is reached, no wait states are added to the command strobe assertion length. If IORDY is negated when the initial sample point is reached, additional wait states are added. Since the rising edge of IORDY must be synchronized, at least two additional PCI clocks are added.
- Shutdown latency is incurred after outstanding scheduled IDE data port transactions (either a non-empty write post buffer or an outstanding read prefetch cycles) have completed and before other transactions can proceed. It provides hold time on the DA[2:0] and CS*nn*# lines with respect to the read and write strobes (DIOR# and DIOW#). Shutdown latency is two PCI clocks in duration.

The IDE timings for various transaction types are shown in Table 46. Note that bit 2 (16 bit I/O recovery enable) of the ISA I/O Recovery Timer Register does not add wait states to IDE data port read accesses when any of the fast timing modes are enabled.

Table 46. IDE Transaction Timings (PCI Clocks)

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

IORDY Masking

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

PIO 32 Bit IDE Data Port Accesses

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

PIO IDE Data Port Prefetching and Posting

The 82801E C-ICH can be programmed via the IDETIM registers to allow data to be posted to and prefetched from the IDE data ports.

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

Data posting is performed for writes to the IDE data ports. The transaction is completed on the PCI bus after the data is received by the 82801E C-ICH. The 82801E C-ICH then runs the IDE cycle to transfer the data to the drive. If the 82801E C-ICH write buffer is non-empty and an unrelated (non-data or opposite channel) IDE transaction occurs, that transaction is stalled until all current data in the write buffer is transferred to the drive.

3.15.2 Bus Master Function

The 82801E C-ICH can act as a PCI Bus master on behalf of an IDE slave device. Two PCI Bus master channels are provided, one channel for each IDE connector (primary and secondary). By performing the IDE data transfer as a PCI Bus master, the 82801E C-ICH off-loads the processor and improves system performance in multitasking environments. Both devices attached to a connector can be programmed for bus master transfers, but only one device per connector can be active at a time.

Physical Region Descriptor Format

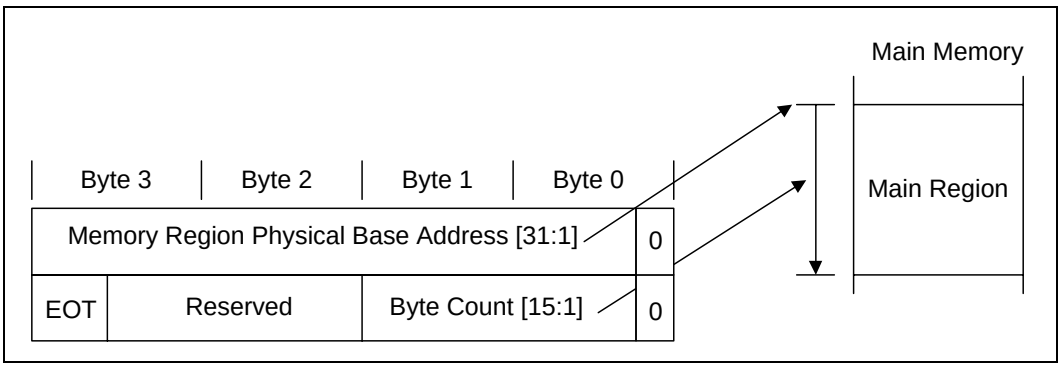
The physical memory region to be transferred is described by a Physical Region Descriptor (PRD). The PRDs are stored sequentially in a Descriptor Table in memory. The data transfer proceeds until all regions described by the PRDs in the table have been transferred. Note that the 82801E C-ICH bus master IDE function does not support memory regions or Descriptor tables located on ISA.

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

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

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

Figure 18. Physical Region Descriptor Table Entry



Line Buffer

A single line buffer exists for the 82801E C-ICH Bus master IDE interface. This buffer is not shared with any other function. The buffer is maintained in either the read state or the write state. Memory writes are typically 4-DWord bursts and invalid DWords have C/BE[3:0]#=0Fh. The line buffer allows burst data transfers to proceed at peak transfer rates.

The Bus Master IDE Active bit in Bus Master IDE Status register is reset automatically when the controller has transferred all data associated with a Descriptor Table (as determined by EOT bit in last PRD). The IDE Interrupt Status bit is set when the IDE device generates an interrupt. These events may occur prior to line buffer emptying for memory writes. If either of these conditions exist, all PCI Master non-memory read accesses to 82801E C-ICH are retried until all data in the line buffers has been transferred to memory.

Bus Master IDE Timings

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

Interrupts

The 82801E C-ICH is connected to IRQ14 for the primary interrupt and IRQ15 for the secondary interrupt. This connection is done from the ISA pin, before any mask registers. This implies the following:

- Bus Master IDE is operating under an interrupt-based driver. Therefore, it does not operate under environments where the IDE device drives an interrupt but the interrupt is masked in the system.
- Bus Master IDE devices are connected directly off of 82801E C-ICH. IDE interrupts cannot be communicated through PCI devices or the serial stream.

Bus Master IDE Operation

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

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

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

The buffer is flushed (when in the write state) or invalidated (when in the read state) when a terminal count condition exists (i.e., the current region descriptor has the EOL bit set and that region has been exhausted). The buffer is also flushed (write state) or invalidated (read state) when the Interrupt bit in the Bus Master IDE Status register is set. Software that reads the status register and finds the Error bit reset, and either the Active bit reset or the Interrupt bit set, can be assured that all data destined for system memory has been transferred and that data is valid in system memory. Table 47 describes how to interpret the Interrupt and Active bits in the Status Register after a DMA transfer has started.

During concurrent DMA or Ultra ATA transfers, the 82801E C-ICH IDE interface arbitrates between the primary and secondary IDE cables when a PRD expires.

Table 47. Interrupt/Active Bit Interaction Definition

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

Error Conditions

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

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

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

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

8237-Like Protocol

The 8237 mode DMA is similar in form to DMA used on the ISA bus. This mode uses a DMA Request, a DMA Acknowledge, and I/O read/write strobes. These signals have similar characteristics to their ISA counterparts in terms of when data is valid relative to strobe edges, and the polarity of the strobes, however the 82801E C-ICH does not use the 8237 for this mode.

3.15.3 Ultra ATA/33 Protocol

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

Ultra ATA/33 is a physical protocol used to transfer data between a Ultra ATA/33 capable IDE controller such as the 82801E C-ICH and one or more Ultra ATA/33 capable IDE devices. It utilizes the standard Bus Master IDE functionality and interface to initiate and control the transfer. Ultra ATA/33 utilizes a “source synchronous” signaling protocol to transfer data at rates up to 33 Mbytes/s. The Ultra ATA/33 definition also incorporates a Cyclic Redundancy Checking (CRC-16) error checking protocol.

Signal Descriptions

The Ultra ATA/33 protocol requires no extra signal pins on the IDE connector. It does redefine a number of the standard IDE control signals when in Ultra ATA/33 mode. These redefinitions are shown in Table 48. Read cycles are defined as transferring data from the IDE device to the 82801E C-ICH. Write cycles are defined as transferring data from the 82801E C-ICH to an IDE device.

Table 48. UltraATA/33 Control Signal Redefinitions

Standard IDE Signal Definition	Ultra ATA/33 Read Cycle Definition	Ultra ATA/33 Write Cycle Definition	82801E C-ICH Primary Channel Signal	82801E C-ICH Secondary Channel Signal
DIOW#	STOP	STOP	PDIOW#	SDIOW#
DIOR#	DMARDY#	STROBE	PDIOR#	SDIOR#
IORDY	STROBE	DMARDY#	PIORDY	SIORDY

The DIOW# signal is redefined as STOP for both read and write transfers. This is always driven by the 82801E C-ICH and is used to request that a transfer be stopped or as an acknowledgment to stop a request from the IDE device.

The DIOR# signal is redefined as DMARDY# for transferring data from the IDE device to the 82801E C-ICH (read). It is used by the 82801E C-ICH to signal when it is ready to transfer data and to add wait states to the current transaction. The DIOR# signal is redefined as STROBE for transferring data from the 82801E C-ICH to the IDE device (write). It is the data strobe signal driven by the 82801E C-ICH on which data is transferred during each rising and falling edge transition.

The IORDY signal is redefined as STROBE for transferring data from the IDE device to the 82801E C-ICH (read). It is the data strobe signal driven by the IDE device on which data is transferred during each rising and falling edge transition. The IORDY signal is redefined as DMARDY# for transferring data from the 82801E C-ICH to the IDE device (write). It is used by the IDE device to signal when it is ready to transfer data and to add wait states to the current transaction.

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

Operation

Initial setup programming consists of enabling and performing the proper configuration of the 82801E C-ICH and the IDE device for Ultra ATA/33 operation. For the 82801E C-ICH, this consists of enabling Synchronous DMA mode and setting up appropriate Synchronous DMA timings.

When transferring data to or from an IDE device, the Bus Master IDE programming model is followed. Once programmed, the drive and the 82801E C-ICH control the transfer of data via the Ultra ATA/33 protocol. The actual data transfer consists of three phases, a start-up phase, a data transfer phase, and a burst termination phase.

The IDE device begins the start-up phase by asserting DMARQ signal. When ready to begin the transfer, the 82801E C-ICH asserts the DMACK# signal. When DMACK# signal is asserted, the host controller drives CS0# and CS1# inactive, DA0–DA2 low. For write cycles, the 82801E C-ICH deasserts STOP, waits for the IDE device to assert DMARDY#, and then drives the first data word and STROBE signal. For read cycles, the 82801E C-ICH tri-states the DD lines, deasserts STOP, and asserts DMARDY#. The IDE device then sends the first data word and STROBE.

The data transfer phase continues the burst transfers with the data transmitter (82801E C-ICH - writes, IDE device - reads) providing data and toggling STROBE. Data is transferred (latched by receiver) on each rising and falling edge of STROBE. The transmitter can pause the burst by holding STROBE high or low, resuming the burst by again toggling STROBE. The receiver can pause the burst by deasserting DMARDY# and resumes the transfers by asserting DMARDY#. The 82801E C-ICH pauses a burst transaction to prevent an internal line buffer over or under flow condition, resuming once the condition has cleared. It may also pause a transaction if the current PRD byte count has expired, resuming once it has fetched the next PRD.

The current burst can be terminated by either the transmitter or receiver. A burst termination consists of a Stop Request, Stop Acknowledge and transfer of CRC data. The 82801E C-ICH can stop a burst by asserting STOP; the IDE device acknowledges by deasserting DMARQ. The IDE device stops a burst by deasserting DMARQ and the 82801E C-ICH acknowledges by asserting STOP. The transmitter then drives the STROBE signal to a high level. The 82801E C-ICH then drives the CRC value on the DD lines and deasserts DMACK#. The IDE device latches the CRC value on the rising edge of DMACK#. The 82801E C-ICH terminates a burst transfer if it needs to service the opposite IDE channel, if a Programmed I/O (PIO) cycle is executed to the IDE channel currently running the burst, or upon transferring the last data from the final PRD.

CRC Calculation

Cyclic Redundancy Checking (CRC-16) is used for error checking on Ultra ATA/33 transfers. The CRC value is calculated for all data by both the 82801E C-ICH and the IDE device over the duration of the Ultra ATA/33 burst transfer segment. This segment is defined as all data transferred with a valid STROBE edge from DDACK# assertion to DDACK# deassertion. At the end of the transfer burst segment, the 82801E C-ICH drives the CRC value onto the DD[15:0] signals. It is then latched by the IDE device on deassertion of DDACK#. The IDE device compares the 82801E C-ICH CRC value to its own and reports an error if there is a mismatch.

3.15.4 Ultra ATA/66 Protocol

In addition to Ultra ATA/33, the 82801E C-ICH supports the Ultra ATA/66 protocol. The Ultra ATA/66 protocol is enabled via configuration bits 3:0 at offset 54h. The two protocols are similar, and are intended to be device driver compatible. The Ultra ATA/66 logic can achieve transfer rates of up to 66 Mbytes/s.

To achieve the higher data rate, the timings are shortened and the quality of the cable is improved to reduce reflections, noise, and inductive coupling. Note that the improved cable is required and will still plug into the standard IDE connector. The Ultra ATA/66 protocol also supports a 44 Mbytes/s mode.

3.15.5 Ultra ATA/100 Protocol

When the ATA_FAST bit is set for any of the four IDE devices, the timings for the transfers to and from the corresponding device run at a higher rate. The 82801E C-ICH Ultra ATA/100 logic can achieve read transfer rates up to 100 Mbytes/s and write transfer rates up to 88.9 Mbytes/s.

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

3.15.6 Ultra ATA/33/66/100 Timing

The timings for Ultra ATA/33/66/100 modes are programmed via the Synchronous DMA Timing Register and the IDE Configuration Register. Different timings can be programmed for each drive in the system. The Base Clock frequency for each drive is selected in the IDE Configuration Register. The Cycle Time (CT) and Ready to Pause (RP) time (defined as multiples of the Base Clock) are programmed in the Synchronous DMA Timing Register. The Cycle Time represents the minimum pulse width of the data strobe (STROBE) signal. The Ready to Pause time represents the number of Base Clock periods that the 82801E C-ICH will wait from deassertion of DMARDY# to the assertion of STOP when it desires to stop a burst read transaction.

Note: The internal Base Clock for Ultra ATA/100 (Mode 5) runs at 133 MHz, and the Cycle Time (CT) must be set for three Base Clocks. The 82801E C-ICH, thus, toggles the write strobe signal every 22.5 ns, transferring two bytes of data on each strobe edge. This means that the 82801E C-ICH performs Mode 5 write transfers at a maximum rate of 88.9 Mbytes/s. For read transfers, the read strobe is driven by the ATA/100 device; the 82801E C-ICH supports reads at the maximum rate of 100 Mbytes/s.

3.16 USB Controller (Device 31:Function 2)

The 82801E C-ICH contains a USB Host Controller. The Host Controller includes a root hub with two separate USB ports. The 82801E C-ICH Host Controller supports the standard Universal Host Controller Interface (UHCI) Rev 1.1.

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

The 82801E C-ICH's USB controller is arbitrated differently than standard PCI devices to improve arbitration latency.

3.16.1 Data Structures in Main Memory

This section describes the details of the data structures used to communicate control, status, and data between software and the 82801E C-ICH: Frame Lists, Transfer Descriptors, and Queue Heads. Frame Lists are aligned on 4 Kbyte boundaries. Transfer Descriptors and Queue Heads are aligned on 16-byte boundaries.

3.16.1.1 Frame List Pointer

The frame list pointer contains a link pointer to the first data object to be processed in the frame, as well as the control bits defined in Table 49.

Table 49. Frame List Pointer Bit Description

Bit	Description
31:4	**Frame List Pointer (FLP).** This field contains the address of the first data object to be processed in the frame and corresponds to memory address signals [31:4], respectively.
3:2	Reserved. These bits must be written as 0.
1	**QH/TD Select (Q).** This bit indicates to the hardware whether the item referenced by the link pointer is a TD (Transfer Descriptor) or a QH (Queue Head). This allows the 82801E C-ICH to perform the proper type of processing on the item after it is fetched. 1 = QH 0 = TD
0	**Terminate (T).** This bit indicates to the 82801E C-ICH whether the schedule for this frame has valid entries in it. 1 = Empty Frame (pointer is invalid). 0 = Pointer is valid (points to a QH or TD).

3.16.1.2 Transfer Descriptor (TD)

Transfer Descriptors (TDs) express the characteristics of the transaction requested on USB by a client. TDs are always aligned on 16-byte boundaries, and the elements of the TD are shown in Figure 19. The four different USB transfer types are supported by a small number of control bits in the descriptor that the 82801E C-ICH interprets during operation. All Transfer Descriptors have the same basic, 32-byte structure. During operation, the 82801E C-ICH hardware performs consistency checks on some fields of the TD. If a consistency check fails, the 82801E C-ICH halts immediately and issues an interrupt to the system. This interrupt cannot be masked within the 82801E C-ICH.

Figure 19. Transfer Descriptor

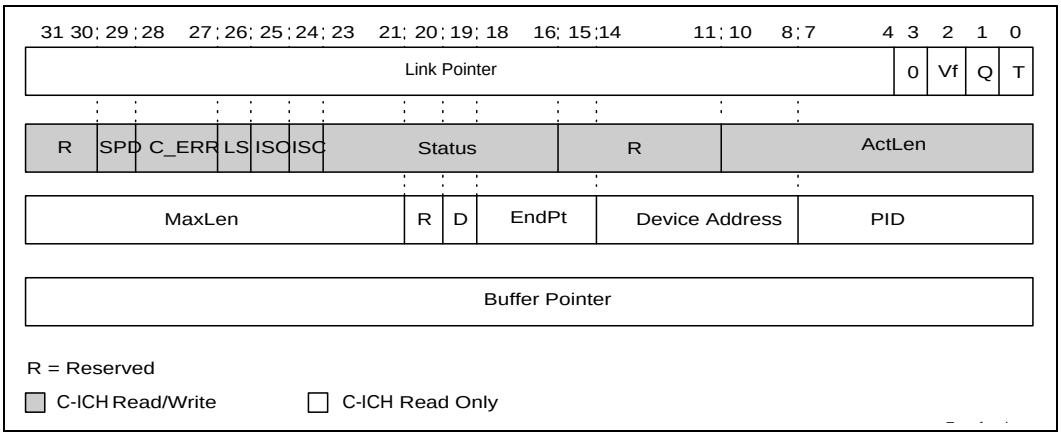


Table 50. TD Link Pointer

Bit	Description
31:4	**Link Pointer (LP).** Bits [31:4] Correspond to memory address signals [31:4], respectively. This field points to another TD or QH.
3	Reserved. Must be 0 when writing this field.
2	**Depth/Breadth Select (VF).** This bit is only valid for queued TDs and indicates to the hardware whether it should process in a depth first or breadth first fashion. When set to depth first, it informs the 82801E C-ICH to process the next transaction in the queue rather than starting a new queue. 1 = Depth first. 0 = Breadth first.
1	**QH/TD Select (Q).** This bit informs the 82801E C-ICH whether the item referenced by the link pointer is another TD or a QH. This allows the 82801E C-ICH to perform the proper type of processing on the item after it is fetched 1 = QH. 0 = TD.
0	**Terminate (T).** This bit informs the 82801E C-ICH that the link pointer in this TD does not point to another valid entry. When encountered in a queue context, this bit indicates to the 82801E C-ICH that there are no more valid entries in the queue. A TD encountered outside of a queue context with the T bit set informs the 82801E C-ICH that this is the last TD in the frame. 1 = Link Pointer field not valid. 0 = Link Pointer field is valid.

Table 51. TD Control and Status (Sheet 1 of 3)

Bit	Description
31:30	Reserved.
29	**Short Packet Detect (SPD).** When a packet has this bit set to 1 and the packet is an input packet, is in a queue; and successfully completes with an actual length less than the maximum length then the TD is marked inactive, the Queue Header is not updated and the USBINT status bit (Status Register) is set at the end of the frame. In addition, if the interrupt is enabled, the interrupt will be sent at the end of the frame. Note that any error (e.g., babble or FIFO error) prevents the short packet from being reported. The behavior is undefined when this bit is set with output packets or packets outside of queues. 1 = Enable. 0 = Disable.
28:27	**Error Counter (C_ERR).** This field is a 2-bit down counter that keeps track of the number of Errors detected while executing this TD. If this field is programmed with a non zero value during setup, the 82801E C-ICH decrements the count and writes it back to the TD if the transaction fails. If the counter counts from one to zero, the 82801E C-ICH marks the TD inactive, sets the “STALLED” and error status bit for the error that caused the transition to zero in the TD. An interrupt will be generated to Host Controller Driver (HCD) if the decrement to zero was caused by Data Buffer error, Bit stuff error, or if enabled, a CRC or Timeout error. If HCD programs this field to zero during setup, the 82801E C-ICH will not count errors for this TD and there will be no limit on the retries of this TD. **Bits[28:27] Interrupt After** 00 No Error Limit 01 1 Error 10 2 Errors 11 3 Errors **Error — Decrement Counter** CRC Error — Yes Timeout Error — Yes NAK Received — No Babble Detected — No† Data Buffer Error — Yes Stalled — No† Bit stuff Error — Yes †Detection of Babble or Stall automatically deactivates the TD. Thus, count is not decremented. †Detection of Babble or Stall automatically deactivates the TD. Thus, count is not decremented.
26	**Low Speed Device (LS).** This bit indicates that the target device (USB data source or sink) is a low speed device, running at 1.5 Mbit/s, instead of at full speed (12 Mbit/sec). There are special restrictions on schedule placement for low speed TDs. If an 82801E C-ICH root hub port is connected to a full speed device and this bit is set to a 1 for a low speed transaction, the 82801E C-ICH sends out a low speed preamble on that port before sending the PID. No preamble is sent if a 82801E C-ICH root hub port is connected to a low speed device. 1 = Low Speed Device 0 = Full Speed Device
25	**Isochronous Select (IOS).** The field specifies the type of the data structure. If this bit is set to a 1, then the TD is an isochronous transfer. Isochronous TDs are always marked inactive by the hardware after execution, regardless of the results of the transaction. 1 = Isochronous Transfer Descriptor 0 = Non-isochronous Transfer Descriptor
24	**Interrupt on Complete (IOC).** This specifies that the 82801E C-ICH should issue an interrupt on completion of the frame in which this Transfer Descriptor is executed. Even if the Active bit in the TD is already cleared when the TD is fetched (no transaction will occur on USB), an IOC interrupt is generated at the end of the frame. 1 = Issue IOC

Table 51. TD Control and Status (Sheet 2 of 3)

Bit	Description
23	**Active.** For 82801E C-ICH schedule execution operations, see the Data Transfers To/From Main Memory section. 1 = Set to 1 by software to enable the execution of a message transaction by the 82801E C-ICH. 0 = When the transaction associated with this descriptor is completed, the 82801E C-ICH sets this bit to 0 indicating that the descriptor should not be executed when it is next encountered in the schedule. The Active bit is also set to 0 if a stall handshake is received from the endpoint.
22	**Stalled.** 1 = Set to a 1 by the 82801E C-ICH during status updates to indicate that a serious error has occurred at the device/endpoint addressed by this TD. This can be caused by babble, the error counter counting down to zero, or reception of the STALL handshake from the device during the transaction. Any time that a transaction results in the Stalled bit being set, the Active bit is also cleared (set to 0). If a STALL handshake is received from a SETUP transaction, a Time Out Error will also be reported.
21	**Data Buffer Error (DBE).** 1 = Set to a 1 by the 82801E C-ICH during status update to indicate that the 82801E C-ICH is unable to keep up with the reception of incoming data (overrun) or is unable to supply data fast enough during transmission (underrun). When this occurs, the actual length and Max Length field of the TD will not match. In the case of an underrun, the 82801E C-ICH transmits an incorrect CRC (thus invalidating the data at the endpoint) and leaves the TD active (unless error count reached zero). If a overrun condition occurs, the 82801E C-ICH forces a timeout condition on the USB, invalidating the transaction at the source.
20	**Babble Detected (BABD).** 1 = Set to a 1 by the 82801E C-ICH during status update when “babble” is detected during the transaction generated by this descriptor. Babble is unexpected bus activity for more than a preset amount of time. In addition to setting this bit, the 82801E C-ICH also sets the” STALLED” bit (bit 22) to a 1. Since “babble” is considered a fatal error for that transfer, setting the” STALLED” bit to a 1 insures that no more transactions occur as a result of this descriptor. Detection of babble causes immediate termination of the current frame. No further TDs in the frame are executed. Execution resumes with the next frame list index.
19	**Negative Acknowledgment (NAK) Received (NAKR).** 1 = Set to a 1 by the 82801E C-ICH during status update when the 82801E C-ICH receives a “NAK” packet during the transaction generated by this descriptor. If a NAK handshake is received from a SETUP transaction, a Time Out Error is also be reported.
18	**CRC/Time Out Error (CRC_TOUT).** 1 = Set to a 1 by the 82801E C-ICH as follows: • During a status update in the case that no response is received from the target device/endpoint within the time specified by the protocol chapter of the USB specification. • During a status update when a Cycli Redundancy Check (CRC) error is detected during the transaction associated with this transfer descriptor. • In the transmit case (OUT or SETUP Command), this is in response to the 82801E C-ICH detecting a timeout from the target device/endpoint. • In the receive case (IN Command), this is in response to the 82801E C-ICH’s CRC checker circuitry detecting an error on the data received from the device/endpoint or a NAK or STALL handshake being received in response to a SETUP transaction.
17	**Bit stuff Error (BSE).** 1 = This bit is set to a 1 by the 82801E C-ICH during status update to indicate that the receive data stream contained a sequence of more than six ones in a row.

Table 51. TD Control and Status (Sheet 3 of 3)

Bit	Description
16	**Bus Turn Around Time-out (BTTO).** 1 = This bit is set to a 1 by the 82801E C-ICH during status updates to indicate that a bus time-out condition was detected for this USB transaction. This time-out is specially defined as not detecting an IDLE-to 'K' state Start of Packet (SOP) transition from 16 to 18 bit times after the SE0-to IDE transition of previous End of Packet (EOP).
15:11	Reserved.
10:0	**Actual Length (ACTLEN).** The Actual Length field is written by the 82801E C-ICH at the conclusion of a USB transaction to indicate the actual number of bytes that were transferred. It can be used by the software to maintain data integrity. The value programmed in this register is encoded as n-1 (see Maximum Length field description in the TD Token).

Table 52. TD Token

Bit	Description
31:21	**Maximum Length (MAXLEN).** The Maximum Length field specifies the maximum number of data bytes allowed for the transfer. The Maximum Length value does not include protocol bytes, such as Packet ID (PID) and CRC. The maximum data packet is 1280 bytes. The 1280 packet length is the longest packet theoretically guaranteed to fit into a frame. Actual packet maximum lengths are set by HCD according to the type and speed of the transfer. Note that the maximum length allowed by the USB specification is 1023 bytes. The valid encodings for this field are: 0x000 = 1 byte 0x001 = 2 bytes .... 0x3FE = 1023 bytes 0x3FF = 1024 bytes .... 0x4FF = 1280 bytes 0x7FF = 0 bytes (null data packet) Note that values from 500h to 7FEh are illegal and cause a consistency check failure. In the transmit case, the 82801E C-ICH uses this value as a terminal count for the number of bytes it fetches from host memory. In most cases, this is the number of bytes it will actually transmit. In rare cases, the 82801E C-ICH may be unable to access memory (e.g., due to excessive latency) in time to avoid underrunning the transmitter. In this instance the 82801E C-ICH would transmit fewer bytes than specified in the Maximum Length field. In the receive case, this field represents the maximum number of bytes that the device should send to the 82801E C-ICH. If the device continues to send after the 82801E C-ICH has received Max
20	Reserved.
19	**Data Toggle (D).** This bit is used to synchronize data transfers between a USB endpoint and the host. This bit determines which data PID is sent or expected (0=DATA0 and 1=DATA1). The Data Toggle bit provides a 1-bit sequence number to check whether the previous packet completed. This bit must always be 0 for Isochronous TDs.
18:15	**Endpoint (ENDPT).** This 4-bit field extends the addressing internal to a particular device by providing 16 endpoints. This permits more flexible addressing of devices in which more than one sub-channel is required.
14:8	**Device Address.** This field identifies the specific device serving as the data source or sink.
7:0	**Packet Identification (PID).** This field contains the Packet ID to be used for this transaction. Only the IN (69h), OUT (E1h), and SETUP (2Dh) tokens are allowed. Any other value in this field causes a consistency check failure resulting in an immediate halt of the 82801E C-ICH. Bits [3:0] are complements of bits [7:4].

Table 53. TD Buffer Pointer

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

3.16.1.3 Queue Head (QH)

Queue heads are special structures used to support the requirements of Control, Bulk and Interrupt transfers. Since these TDs are not automatically retired after each use, their maintenance requirements can be reduced by putting them into a queue. Queue heads must be aligned on a 16-byte boundary; the elements are shown in Table 54.

Table 54. Queue Head Block

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

Table 55. Queue Head Link Pointer

Bit	Description
31:4	**Queue Head Link Pointer (QHLP).** This field contains the address of the next data object to be processed in the horizontal list and corresponds to memory address signals [31:4], respectively.
3:2	Reserved. These bits must be written as 0s.
1	**QH/TD Select (Q).** This bit indicates to the hardware whether the item referenced by the link pointer is another TD or a QH. 1=QH 0=TD
0	**Terminate (T).** This bit indicates to the 82801E C-ICH that this is the last QH in the schedule. If there are active TDs in this queue, they are the last to be executed in this frame. 1 = Last QH (pointer is invalid). 0 = Pointer is valid (points to a QH or TD).

Table 56. Queue Element Link Pointer

Bit	Description
31:4	**Queue Element Link Pointer (QELP).** This field contains the address of the next TD or QH to be processed in this queue and corresponds to memory address signals [31:4], respectively.
3:2	Reserved.
1	**QH/TD Select (Q).** This bit indicates to the hardware whether the item referenced by the link pointer is another TD or a QH. For entries in a queue, this bit is typically set to 0. 1 = QH 0 = TD
0	**Terminate (T).** This bit indicates to the 82801E C-ICH that there are no valid TDs in this queue. When HCD has new queue entries it overwrites this value with a new TD pointer to the queue entry. 1 = Terminate (No valid queue entries). 0 = Pointer is valid.

3.16.2 Data Transfers To/From Main Memory

The following sections describe the details on how the Host Controller Driver (HCD) and the 82801E C-ICH communicate via the schedule data structures. The discussion is organized in a top-down manner, beginning with the basics of walking the Frame List, followed by a description of generic processing steps common to all transfer descriptors, and finally a discussion on Transfer Queuing.

3.16.2.1 Executing the Schedule

Software programs the 82801E C-ICH with the starting address of the Frame List and the Frame List index, then causes the 82801E C-ICH to execute the schedule by setting the Run/Stop bit in the Control register to Run. The 82801E C-ICH processes the schedule one entry at a time. The next element in the frame list is not fetched until the current element in the frame list is retired.

Schedule execution proceeds in the following fashion:

- The 82801E C-ICH first fetches an entry from the Frame List. This entry has three fields. Bit 0 indicates whether the address pointer field is valid. Bit 1 indicates whether the address points to a Transfer Descriptor or to a queue head. The third field is the pointer itself.
- If isochronous traffic is to be moved in a given frame, the Frame List entry points to a Transfer Descriptor. If no isochronous data is to be moved in that frame, the entry points to a queue head or the entry is marked invalid and no transfers are initiated in that frame.
- If the Frame List entry indicates that it points to a Transfer Descriptor, the 82801E C-ICH fetches the entry and begins the operations necessary to initiate a transaction on USB. Each TD contains a link field that points to the next entry, as well as indicating whether it is a TD or a QH.
- If the Frame List entry contains a pointer to a QH, the 82801E C-ICH processes the information from the QH to determine the address of the next data object that it should process.
- The TD/QH process continues until the millisecond allotted to the current frame expires. At this point, the 82801E C-ICH fetches the next entry from the Frame List. If the 82801E C-ICH is not able to process all of the transfer descriptors during a given frame, those descriptors are retired by software without having been executed.

3.16.2.2 Processing Transfer Descriptors

The 82801E C-ICH executes a TD using the following generalized algorithm. These basic steps are common across all modes of TDs. Subsequent sections present processing steps unique to each TD mode.

1. 82801E C-ICH Fetches TD or QH from the current Link Pointer.
2. If a QH, go to 1 to fetch from the Queue Element Link Pointer. If inactive, go to 12
3. Build Token, actual bits are in TD Token.
4. If (Host-to-Function) then
   [*PCI Access*] issue request for data, (referenced through TD.BufferPointer)
   wait for first chunk data arrival
   end if
5. [*Begin USB Transaction*] Issue Token (from token built in 2, above) and begin data transfer.
   if (Host-to-Function) then Go to 6
   else Go to 7
   end if
6. Fetch data from memory (via TD BufferPointer) and transfer over USB until TD Max-Length bytes have been read and transferred. [*Concurrent system memory and USB Accesses*]. Go to 8.
7. Wait for data to arrive (from USB). Write incoming bytes into memory beginning at TD BufferPointer. Internal HC buffer should signal end of data packet. Number of bytes received must be (TD Max-Length; The length of the memory area referenced by TD BufferPointer must be (TD Max-Length. [*Concurrent system memory and USB Accesses*].
8. Issue handshake based on status of data received (Ack or Time-out). Go to 10.
9. Wait for handshake, if required [*End of USB Transaction*].
10. Update Status [*PCI Access*] (TD.Status and TD.ActualLength).
    If the TD was an isochronous TD, mark the TD inactive. Go to 12.
    If not an isochronous TD, and the TD completed successfully, mark the TD inactive. Go to 11.
    If not successful, and the error count has not been reached, leave the TD active. If the error count has been reached, mark the TD inactive. Go to 12.
11. Write the link pointer from the current TD into the element pointer field of the QH structure. If the Vf bit is set in the TD link pointer, go to 2.
12. Proceed to next entry.

3.16.2.3 Command Register, Status Register, and TD Status Bit Interaction

Table 57. Command Register, Status Register and TD Status Bit Interaction

Condition	82801E C-ICH USB Status Register Actions	TD Status Register Actions
CRC/Time Out Error	Set USB Error Int bit[1], Clear HC Halted bit	Clear Active bit[1] and set Stall bit[1]
Illegal PID, PID Error, Max Length (illegal)	Clear Run/Stop bit in command register Set HC Process Error and HC Halted bits	
PCI Master/Target Abort	Clear Run/Stop bit in command register Set Host System Error and HC Halted bits	
Resume Received and Suspend Mode[2] = 1	Set Resume received bit[2] **NOTE:** Not supported; must be set to 0.	
Run/Stop = 0	Clear Run/Stop bit in command register Set HC Halted bit	
Configuration Flag Set	Set configuration Flag in command register	
HC Reset/Global Reset	Clear Run/Stop and configuration Flag in command register Clear USB Int, USB Error Int, Resume received, Host System Error, HC Process Error, and HC Halted bits	
IOC = 1 in TD Status	Set USB Int bit	
Stall	Set USB Error Int bit	Clear Active bit[1] and set Stall bit
Bit Stuff/Data Buffer Error	Set USB Error Int bit[1]	Clear Active bit[1] and set Stall bit[1]
Short Packet Detect	Set USB Int bit	Clear Active bit

NOTES:
1. Only If error counter counted down from 1 to 0
2. The 82801E C-ICH does not support USB power management.

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

3.16.2.4 Transfer Queuing

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

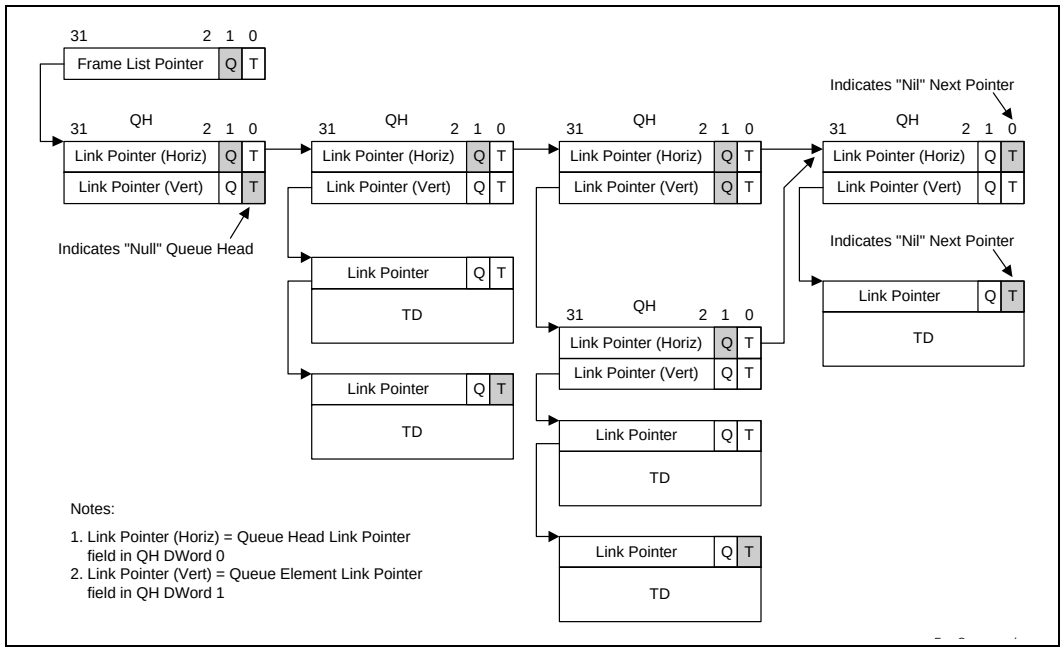
The QH contains two link pointers and is organized as two contiguous DWords. The first DWord is a horizontal pointer (Queue Head Link Pointer), used to link a single transfer queue with either another transfer queue, or a TD (target data structure depends on Q bit). If the T bit is set, this QH represents the last data structure in the current Frame. The T bit informs the 82801E C-ICH that no further processing is required until the beginning of the next frame. The second DWord is a vertical pointer (Queue Element Link Pointer) to the first data structure (TD or QH) being managed by this QH. If the T bit is set, the queue is empty. This pointer may reference a TD or another QH.

Figure 20 illustrates four example queue conditions. The first QH (on far left) is an example of an “empty” queue; the termination bit (T Bit), in the vertical link pointer field, is set to 1. The horizontal link pointer references another QH. The next queue is the expected typical configuration. The horizontal link pointer references another QH, and the vertical link pointer references a valid TD.

Typically, the vertical pointer in a QH points to a TD. However, as shown in Figure 20 (third example from left side of figure) the vertical pointer could point to another QH. When this occurs, a new Q Context is entered and the Q Context just exited is NULL (82801E C-ICH does not update the vertical pointer field).

The far right QH is an example of a frame ‘termination’ node. Since its horizontal link pointer has its termination bit set, the 82801E C-ICH assumes there is no more work to complete for the current Frame.

Figure 20. Example Queue Conditions



Transfer Queues are based on the following characteristics:

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

In the simplest model, the 82801E C-ICH follows vertical link point to a queue element, then executes the element. If the completion status of the TD satisfies the advance criteria as shown in Table 58, the 82801E C-ICH advances the queue by writing the just-executed TD’s link pointer back into the QH’s Queue Element link pointer. The next time the queue head is traversed, the next queue element will be the Top element.

The traversal has two options: Breadth first, or Depth first. A flag bit in each TD (Vf - Vertical Traversal Flag) controls whether traversal is Breadth or Depth first. The default mode of traversal is Breadth-First. For Breadth-First, the 82801E C-ICH only executes the top element from each queue. The execution path is shown below:

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

Breadth-First is also performed for every transaction execution that fails the advance criteria. This means that if a queued TD fails, the queue does not advance, and the 82801E C-ICH traverses the QH's Queue Head Link Pointer.

In a Depth-first traversal, the top queue element must complete successfully to satisfy the *advance criteria* for the queue. If the 82801E C-ICH is currently processing a queue, and the advance criteria are met, and the Vf bit is set, the 82801E C-ICH follows the TD's link pointer to the next schedule work item.

Note that regardless of traversal model, when the advance criteria are met, the successful TD's link pointer is written back to the QH's Queue Element link pointer.

When the 82801E C-ICH encounters a QH, it caches the QH internally, and sets internal state to indicate it is in a Q-context. It needs this state to update the correct QH (for auto advancement) and also to make the correct decisions on how to traverse the Frame List.

Restricting the advancement of queues to advancement criteria implements a guaranteed data delivery stream.

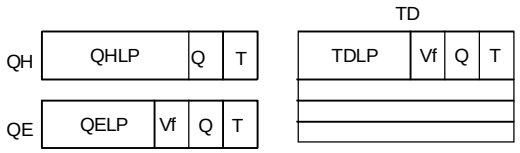
A queue is **never** advanced on an error completion status (even in the event the error count was exhausted).

Table 58 lists the general queue advance criteria, which are based on the execution status of the TD at the "Top" of a currently "active" queue.

Table 58. Queue Advance Criteria

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

Table 59 is a decision table illustrating the valid combinations of link pointer bits and the valid actions taken when advancement criteria for a queued transfer descriptor are met. The column headings for the link pointer fields are encoded, based on the following list:



Legends:

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

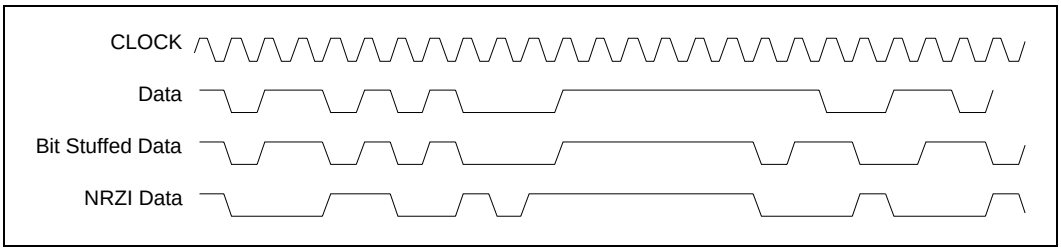
Table 59. USB Schedule List Traversal Decision Table

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

3.16.3 Data Encoding and Bit Stuffing

The USB employs NRZI data encoding (Non-Return to Zero Inverted) when transmitting packets. In NRZI encoding, a 1 is represented by no change in level and a 0 is represented by a change in level. A string of zeros causes the NRZI data to toggle each bit time. A string of ones causes long periods with no transitions in the data. To ensure adequate signal transitions, bit stuffing is employed by the transmitting device when sending a packet on the USB. A 0 is inserted after every six consecutive 1s in the data stream before the data is NRZI encoded to force a transition in the NRZI data stream. This gives the receiver logic a data transition at least once every seven bit times to guarantee the data and clock lock. A waveform of the data encoding is shown in Figure 21.

Figure 21. USB Data Encoding



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

3.16.4 Bus Protocol

3.16.4.1 Bit Ordering

Bits are sent out onto the bus least significant bit (LSb) first, followed by next LSb, through to the most significant bit (MSb) last.

3.16.4.2 SYNC Field

All packets begin with a synchronization (SYNC) field, which is a coded sequence that generates a maximum edge transition density. The SYNC field appears on the bus as IDLE followed by the binary string "KJKJKJKK," in its NRZI encoding. It is used by the input circuitry to align incoming data with the local clock and is defined to be eight bits in length. SYNC serves only as a synchronization mechanism and is not shown in the following packet diagrams. The last two bits in the SYNC field are a marker that is used to identify the first bit of the PID. All subsequent bits in the packet must be indexed from this point.

3.16.4.3 Packet Field Formats

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

Table 60. PID Format

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

Packet Identifier Field

A packet identifier (PID) immediately follows the SYNC field of every USB packet. A PID consists of a four bit packet type field followed by a four-bit check field as shown in Table 60. The PID indicates the type of packet and, by inference, the format of the packet and the type of error detection applied to the packet. The four-bit check field of the PID insures reliable decoding of the PID so that the remainder of the packet is interpreted correctly. The PID check field is generated by performing a ones complement of the packet type field.

Any PID received with a failed check field or that decodes to a non-defined value is assumed to be corrupted and the remainder of the packet is assumed to be corrupted and is ignored by the receiver. PID types, codes, and descriptions are listed in Table 61.

Table 61. PID Types

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

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

3.16.4.4 Address Fields

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

Table 62. Address Field

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

Bit	Data Sent
4	ADDR 4
5	ADDR 5
6	ADDR 6

Address Field

The function address (ADDR) field specifies the function, via its address, that is either the source or destination of a data packet, depending on the value of the token PID. As shown in Table 62, a total of 128 addresses are specified as ADDR[6:0]. The ADDR field is specified for IN, SETUP, and OUT tokens.

Endpoint Field

An additional four-bit endpoint (ENDP) field, shown in Table 63, permits more flexible addressing of functions in which more than one sub-channel is required. Endpoint numbers are function specific. The endpoint field is defined for IN, SETUP, and OUT token PIDs only.

Table 63. Endpoint Field

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

3.16.4.5 Frame Number Field

The frame number field is an 11-bit field that is incremented by the host on a per frame basis. The frame number field rolls over upon reaching its maximum value of x7FFh and is sent only for SOF tokens at the start of each frame.

3.16.4.6 Data Field

The data field may range from 0 to 1023 bytes and must be an integral numbers of bytes. Data bits within each byte are shifted out LSB first.

3.16.4.7 Cyclic Redundancy Check (CRC)

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

3.16.5 Packet Formats

3.16.5.1 Token Packets

Table 64 shows the field formats for a token packet. A token consists of a PID, specifying either IN, OUT, or SETUP packet type, and ADDR and ENDP fields. For OUT and SETUP transactions, the address and endpoint fields uniquely identify the endpoint that will receive the subsequent data packet. For IN transactions, these fields uniquely identify which endpoint should transmit a data packet. Only the 82801E C-ICH can issue token packets. IN PIDs define a data transaction from a function to the 82801E C-ICH. OUT and SETUP PIDs define data transactions from the 82801E C-ICH to a function.

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

Table 64. Token Format

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

3.16.5.2 Start of Frame Packets

Table 65 shows a start of frame (SOF) packet. SOF packets are issued by the host at a nominal rate of once every 1.00 ms. SOF packets consist of a PID indicating packet type followed by an 11-bit frame number field.

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

Table 65. SOF Packet

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

3.16.5.3 Data Packets

A data packet consists of a PID, a data field, and a CRC as shown in Table 66. There are two types of data packets identified by differing PIDs: DATA0 and DATA1. Two data packet PIDs are defined to support data toggle synchronization.

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

Table 66. Data Packet Format

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

3.16.5.4 Handshake Packets

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

There are three types of handshake packets:

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

3.16.5.5 Handshake Responses

IN Transaction

A function may respond to an IN transaction with a STALL or NAK. If the token received was corrupted, the function issues no response. If the function can transmit data, it issues the data packet. The 82801E C-ICH, as the USB host, can return only one type of handshake on an IN transaction, an ACK. If it receives a corrupted data or cannot accept data due to a condition such as an internal buffer overrun, it discards the data and issues no response.

OUT Transaction

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

SETUP Transaction

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

3.16.6 USB Interrupts

There are two general groups of USB interrupt sources, those resulting from execution of transactions in the schedule, and those resulting from an 82801E C-ICH operation error. All transaction-based sources can be masked by software through the 82801E C-ICH's Interrupt Enable register. Additionally, individual transfer descriptors can be marked to generate an interrupt on completion.

When the 82801E C-ICH drives an interrupt for USB, it drives the PIRQD# pin active for interrupts occurring due to ports 0 and 1 until all sources of the interrupt are cleared.

3.16.6.1 Transaction Based Interrupts

These interrupts are not signaled until after the status for the last complete transaction in the frame has been written back to host memory. This guarantees that software can safely process through (Frame List Current Index -1) when it is servicing an interrupt.

CRC Error/Time-out

A CRC/Time-out error occurs when a packet transmitted from the 82801E C-ICH to a USB device or a packet transmitted from a USB device to the 82801E C-ICH generates a CRC error. The 82801E C-ICH is informed of this event by a time-out from the USB device or by the 82801E C-ICH's CRC checker generating an error on reception of the packet. Additionally, a USB bus time-out occurs when USB devices do not respond to a transaction phase within 19 bit times of an EOP. Either of these conditions will cause the C_ERR field of the TD to decrement. When the C_ERR field decrements to zero, the following occurs:

- The Active bit in the TD is cleared
- The Stalled bit in the TD is set
- The CRC/Time-out bit in the TD is set
- At the end of the frame, the USB Error Interrupt bit is set in the HC status register

If the CRC/Time out interrupt is enabled in the Interrupt Enable register, a hardware interrupt is signaled to the system.

Interrupt on Completion

Transfer Descriptors contain a bit that can be set to cause an interrupt upon their completion. The completion of the transaction associated with that block causes the USB Interrupt bit in the HC Status Register to be set at the end of the frame in which the transfer completed. When a TD is encountered with the IOC bit set to 1, the IOC bit in the HC Status register is set to 1 at the end of the frame if the active bit in the TD is set to 0 (even if it was set to zero when initially read).

If the IOC Enable bit of Interrupt Enable register (bit 2 of I/O offset 04h) is set, a hardware interrupt is signaled to the system. The USB Interrupt bit in the HC Status register is set either when the TD completes successfully or because of errors. If the completion is because of errors, the USB Error bit in the HC Status register is also set.

Short Packet Detect

A transfer set is a collection of data that requires more than one USB transaction to completely move the data across the USB. An example might be a large print file that requires numerous TDs in multiple frames to completely transfer the data. Reception of a data packet that is less than the endpoint's Max Packet size during Control, Bulk or Interrupt transfers signals the completion of the transfer set, even if there are active TDs remaining for this transfer set. Setting the SPD bit in a TD indicates to the HC to set the USB Interrupt bit in the HC Status register at the end of the frame in which this event occurs. This feature streamlines the processing of input on these transfer types. If the Short Packet Interrupt Enable bit in the Interrupt Enable register is set, a hardware interrupt is signaled to the system at the end of the frame where the event occurred.

Serial Bus Babble

When a device transmits on the USB for a time greater than its assigned Max Length, it is said to be babbling. Since isochrony can be destroyed by a babbling device, this error results in the Active bit in the TD being cleared to 0 and the Stalled and Babble bits being set to one. The C_ERR field is not decremented for a babble. The USB Error Interrupt bit in the HC Status register is set to 1 at the end of the frame. A hardware interrupt is signaled to the system.

If an EOF babble was caused by the 82801E C-ICH (due to incorrect schedule for instance), the 82801E C-ICH forces a bit stuff error followed by an EOP and the start of the next frame.

Stalled

This event indicates that a device/endpoint returned a STALL handshake during a transaction or that the transaction ended in an error condition. The TDs Stalled bit is set and the Active bit is cleared. Reception of a STALL does not decrement the error counter. A hardware interrupt is signaled to the system.

Data Buffer Error

This event indicates that an overrun of incoming data or a under-run of outgoing data has occurred for this transaction. This would generally be caused by the 82801E C-ICH not being able to access required data buffers in memory within necessary latency requirements. Either of these conditions causes the C_ERR field of the TD to be decremented.

When C_ERR decrements to zero, the Active bit in the TD is cleared, the Stalled bit is set, the USB Error Interrupt bit in the HC Status register is set to 1 at the end of the frame and a hardware interrupt is signaled to the system.

Bit Stuff Error

A bit stuff error results from the detection of a sequence of more that 6 ones in a row within the incoming data stream. This will cause the C_ERR field of the TD to be decremented. When the C_ERR field decrements to zero, the Active bit in the TD is cleared to 0, the Stalled bit is set to 1, the USB Error Interrupt bit in the HC Status register is set to 1 at the end of the frame and a hardware interrupt is signaled to the system.

3.16.6.2 Non-Transaction Based Interrupts

If an 82801E C-ICH process error or system error occur, the 82801E C-ICH halts and immediately issues a hardware interrupt to the system.

Resume Received

This event indicates that the 82801E C-ICH received a RESUME signal from a device on the USB bus during a global suspend. If this interrupt is enabled in the Interrupt Enable register, a hardware interrupt will be signaled to the system that allows the USB to be brought out of the suspend state and returned to normal operation.

82801E C-ICH Process Error

The HC monitors certain critical fields during operation to ensure that it does not process corrupted data structures. These include checking for a valid PID and verifying that the MaxLength field is less than 1280. If it detects a condition that would indicate that it is processing corrupted data structures, it immediately halts processing, sets the HC Process Error bit in the HC Status Register, and signals a hardware interrupt to the system.

This interrupt cannot be disabled through the Interrupt Enable Register.

Host System Error

The 82801E C-ICH sets this bit to 1 when a PCI Parity error, PCI Master Abort, or PCI Target Abort occurs. When this error occurs, the 82801E C-ICH clears the Run/Stop bit in the Command Register to prevent further execution of the scheduled TDs. This interrupt cannot be disabled through the Interrupt Enable Register.

3.16.7 USB Power Management

The 82801E C-ICH does not support USB power management. The 82801E C-ICH is an on or off device.

The settings of the following bits in I/O space are maintained since the 82801E C-ICH is always on.

Table 67. Bits Maintained

Register	Offset	Bit	Description
Command	00h	3	Enter Global Suspend Mode (EGSM)
Status	02h	2	Resume Detect
Port Status and Control	10h & 12h	2	Port Enabled/Disabled
		6	Resume Detect
		8	Low Speed Device Attached
		12	Suspend

3.16.8 USB Legacy Keyboard Operation

When a USB keyboard is plugged into the system and a standard keyboard is not, the system may not boot and DOS legacy software will not run; this is because the keyboard is not identified. The 82801E C-ICH implements a series of trapping operations that snoop accesses that go to the keyboard controller and put the expected data from the USB keyboard into the keyboard controller.

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

This legacy operation is performed through SMM space.

Figure 22 shows the Enable and Status path. The latched SMI source (60R, 60W, 64R, 64W) is available in the Status Register. Because the enable is after the latch, it is possible to check for other events that didn't necessarily cause an SMI. It is the software's responsibility to logically AND the value with the appropriate enable bits.

Note also that the SMI is generated before the PCI cycle completes (e.g., before TRDY# goes active) to ensure that the processor does not complete the cycle before the SMI is observed. This method is used on MPIIX and has been validated.

The logic will also need to block the accesses to the 8042. If there is an external 8042, this is accomplished by not activating the 8042 CS. This is done by logically ANDing the 4 enables (60R, 60W, 64R, 64W) with the 4 types of accesses to determine if the 8042CS should go active. An additional term is required for the "Pass-through" case. The state table for the diagram is shown in Table 68.

Figure 22. USB Legacy Keyboard Flow Diagram

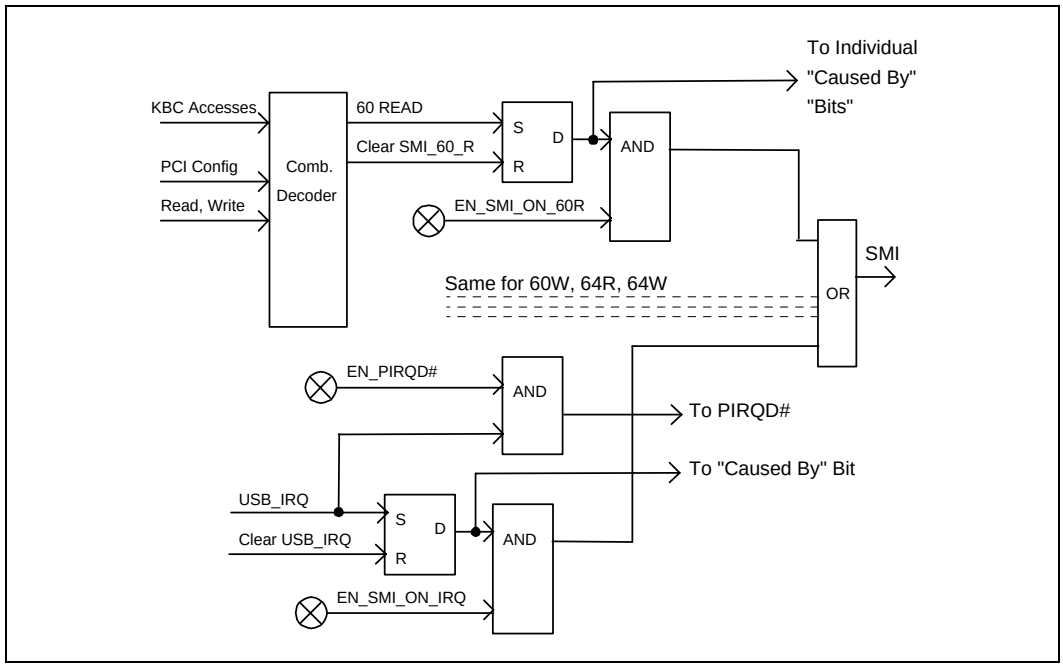


Table 68. USB Legacy Keyboard State Transitions (Sheet 1 of 2)

Current State	Action	Data Value	Next State	Comment
IDLE	64h/Write	D1h	GateState1	Standard D1 command. The cycle is passed through to the 8042. SMI# does not go active. PSTATE goes to 1.
IDLE	64h/Write	Not D1h	IDLE	Bit 3 in the configuration register determines if the cycle is passed through to the 8042 and if an SMI# is generated.
IDLE	64h/Read	N/A	IDLE	Bit 2 in the configuration register determines if the cycle is passed through to the 8042 and if an SMI# is generated.
IDLE	60h/Write	Don't Care	IDLE	Bit 1 in the configuration register determines if the cycle is passed through to the 8042 and if an SMI# is generated.
IDLE	60h/Read	N/A	IDLE	Bit 0 in the configuration register determines if the cycle is passed through to the 8042 and if an SMI# is generated.
GateState1	60h/Write	XXh	GateState2	The cycle is passed through to the 8042, even if a trap is enabled in Bit 1 in the configuration register. No SMI# is generated. PSTATE remains 1. If the data value is not DFh or DDh, the 8042 may choose to ignore it.
GateState1	64h/Write	D1h	GateState1	The cycle is passed through to the 8042, even if a trap is enabled via Bit 3 in the configuration register. No SMI# is generated. PSTATE remains 1. Stay in GateState1 because this is part of the double-trigger sequence.
GateState1	64h/Write	Not D1h	ILDE	Bit 3 in configuration space determines if the cycle is passed through to the 8042 and if an SMI# is generated. PSTATE goes to 0. If Bit 7 in configuration Register is set, then SMI# should be generated.

Table 68. USB Legacy Keyboard State Transitions (Sheet 2 of 2)

Current State	Action	Data Value	Next State	Comment
GateState1	60h/Read	N/A	IDLE	This is an invalid sequence. Bit 0 in the configuration register determines if a cycle is passed through to the 8042 and if an SMI# is generated. PSTATE goes to 0. If Bit 7 in the configuration register is set, then an SMI# is generated.
GateState1	64h/Read	N/A	GateState1	Remain in the same state. Generate an SMI# if enabled in Bit 2 of the configuration register. PSTATE remains 1.
GateState2	64/Write	FFh	IDLE	Standard end of sequence. The cycle is passed through to the 8042. PSTATE goes to 0. Bit 7 in configuration space determines if an SMI# is generated.
GateState2	64h/Write	Not FFh	IDLE	Improper end of sequence. Bit 3 in the configuration register determines if the cycle is passed through to the 8042 and if an SMI# is generated. PSTATE goes to 0. If Bit 7 in the configuration register is set, then an SMI# is generated.
GateState2	64h/Read	N/A	GateState2	Remain in the same state. Generate an SMI# if enabled in Bit 2 of the configuration register. PSTATE remains 1.
GateState2	60h/Write	XXh	IDLE	Improper end of sequence. Bit 1 in the configuration register determines if the cycle is passed through to the 8042 and if an SMI# is generated. PSTATE goes to 0. If Bit 7 in the configuration register is set, then an SMI# is generated.
GateState2	60h/Read	N/A	IDLE	Improper end of sequence. Bit 0 in the configuration register determines if the cycle is passed through to the 8042 and if an SMI# is generated. PSTATE goes to 0. If Bit 7 in the configuration register is set, an SMI# is generated.

3.17 SMBus Controller Functional Description (D31:F3)

The 82801E C-ICH provides an SMBus Host Controller as well as an SMBus Slave Interface.

The Host Controller provides a mechanism for the processor to initiate communications with SMBus peripherals (slaves). The 82801E C-ICH is also capable of operating in a mode in which it can communicate with I^2C compatible devices.

The Slave Interface allows an external master to read from or write to the 82801E C-ICH. Write cycles can be used to cause certain events or pass messages and read cycles can be used to determine the state of various status bits. The 82801E C-ICH's internal Host Controller cannot access the 82801E C-ICH's internal Slave Interface.

The 82801E C-ICH SMBus logic exists in Device 31:Function 3 configuration space and consists of a transmit data path and host controller. The transmit data path provides the data flow logic needed to implement the seven different SMBus command protocols and is controlled by the host controller. The 82801E C-ICH SMBus controller logic is clocked by RTC clock.

The programming model of the host controller is combined into two portions: a PCI configuration portion and a system I/O mapped portion. All static configuration (e.g., the I/O base address) is done via the PCI configuration space. Real-time programming of the Host interface is done in system I/O space.

3.17.1 Host Controller

The SMBus Host Controller is used to send commands to other SMBus slave devices. Software sets up the host controller with an address, command, and, for writes, data, and then tells the controller to start. When the controller has finished transmitting data on writes, or receiving data on reads, it generates an SMI# or interrupt, if enabled.

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

The SMBus Host Controller requires that the various data and command fields be setup for the type of command to be sent. When software sets the START bit, the SMBus Host Controller performs the requested transaction and interrupts the processor (or generate an SMI#) when the transaction is completed. Once a START command has been issued, the values of the “active registers” (Host Control, Host Command, Transmit Slave Address, Data 0, Data 1) should not be changed or read until the interrupt status bit (INTR) has been set (indicating the completion of the command). Any register values needed for computation purposes should be saved prior to issuing of a new command, as the SMBus Host Controller will update all registers while completing the new command.

Using the SMB Host Controller to send commands to the 82801E C-ICH's SMB slave port is not supported.

3.17.1.1 Command Protocols

In all of the following commands, the Host Status Register (offset 00h) is used to determine the progress of the command. While the command is in operation, the HOST_BUSY bit is set. If the command completes successfully, the INTR bit is set in the Host Status Register. If the device does not respond with an acknowledge and the transaction times out, the DEV_ERR bit is set. If software sets the KILL bit in the Host Control Register while the command is running, the transaction will stop and the FAILED bit will be set.

Quick Command

When programmed for a Quick Command, the Transmit Slave Address Register is sent. The format of the protocol is shown in Table 69.

Table 69. Quick Protocol

Bit	Description
1	Start Condition
2:8	Slave Address - 7 bits
9	Read/Write Direction
10	Acknowledge from slave
11	Stop

Send Byte/Receive Byte

For the Send Byte command, the Transmit Slave Address and Device Command Registers are sent For the Receive Byte command, the Transmit Slave Address Register is sent. The data received is stored in the DATA0 register.

The Receive Byte is similar to a Send Byte; the only difference is the direction of data transfer. The format of the protocol is shown in Table 70.

Table 70. Send/Receive Byte Protocol

Send Byte Protocol		Receive Byte Protocol	
Bit	**Description**	**Bit**	**Description**
1	Start	1	Start
2:8	Slave Address - 7 bits	2:8	Slave Address - 7 bits
9	Write	9	Read
10	Acknowledge from slave	10	Acknowledge from slave
11:18	Command code - 8 bits	11:18	Data byte from slave
19	Acknowledge from slave	19	NOT Acknowledge
20	Stop	20	Stop

Write Byte/Word

The first byte of a Write Byte/Word access is the command code. The next 1 or 2 bytes are the data to be written. When programmed for a write byte/word command, the Transmit Slave Address, Device Command and Data0 Registers are sent. In addition, the Data1 Register is sent on a write word command. The format of the protocol is shown in Table 71.

Table 71. Write Byte/Word Protocol

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

Read Byte/Word

Reading data is slightly more complicated than writing data. First the 82801E C-ICH must write a command to the slave device. Then it must follow that command with a repeated start condition to denote a read from that device's address. The slave then returns 1 or 2 bytes of data.

When programmed for the read byte/word command, the Transmit Slave Address and Device Command Registers are sent. Data is received into the DATA0 on the read byte, and the DAT0 and DATA1 registers on the read word. The format of the protocol is shown in Table 72.

Table 72. Read Byte/Word Protocol

Read Byte Protocol		Read Word Protocol	
Bit	**Description**	**Bit**	**Description**
1	Start	1	Start
2:8	Slave Address - 7 bits	2:8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11:18	Command code - 8 bits	11:18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave
20	Repeated Start	20	Repeated Start
21:27	Slave Address - 7 bits	21:27	Slave Address - 7 bits
28	Read	28	Read
29	Acknowledge from slave	29	Acknowledge from slave
30:37	Data from slave - 8 bits	30:37	Data Byte Low from slave - 8 bits
38	NOT acknowledge	38	Acknowledge
39	Stop	39:46	Data Byte High from slave - 8 bits
		47	NOT acknowledge
		48	Stop

Process Call

The process call is so named because a command sends data and waits for the slave to return a value dependent on that data. The protocol is simply a Write Word followed by a Read Word, but without a second command or stop condition.

When programmed for the Process Call command, the 82801E C-ICH transmits the Transmit Slave Address, Host Command, DATA0 and DATA1 registers. Data received from the device is stored in the DATA0 and DATA1 registers. The format of the protocol is shown in Table 73.

Note: For process call command, the value written into bit 0 of the Transmit Slave Address Register (SMB I/O register, offset 04h) needs to be 0.

Table 73. Process Call Protocol

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

Block Read/Write

The Block Write begins with a slave address and a write condition. After the command code, the 82801E C-ICH issues a byte count that describes how many more bytes will follow in the message. If a slave has 20 bytes to send, the first byte is the number 20 (14h), followed by the 20 bytes of data. The byte count may not be 0.

Note that, unlike the PIIX4, which implements 32-byte buffer for Block Read/Write command, the 82801E C-ICH implements the Block Data Byte register (D31:F3, I/O offset 07h) for Block Read/Write command.

When programmed for a block write command, the Transmit Slave Address, Host Command, and Data0 (count) registers are sent. Data is then sent from the Block Data Byte register. After the byte has been sent, the 82801E C-ICH sets the BYTE_DONE_STS bit in the Host Status register. If there are more bytes to send, software writes the next byte to the Block Data Byte register and clears the BYTE_DONE_STS bit. The 82801E C-ICH then sends the next byte. When doing a block write, first poll the BYTE_DONE_STS register until it is set, then write the next byte, then clear the BYTE_DONE_STS register.

On block read commands, after the byte count is stored in the DATA 0 register, the first data byte goes in the Block Data Byte Register; the 82801E C-ICH will then set the BYTE_DONE_STS bit and generate an SMI# or interrupt. The SMI# or interrupt handler reads the byte and then clears the BYTE_DONE_STS bit to allow the next byte to be read into the Block Data Byte register. Note

that after receiving data byte N-1 of the block, the software needs to set the LAST_BYTE bit in the Host Control Register; this allows the 82801E C-ICH to send a NOT ACK (instead of an ACK) after receiving the last data byte (byte N) of the block.

After each byte of a block message the 82801E C-ICH sets the BYTE_DONE_STS bit and generates an interrupt or SMI#. Software clears the BYTE_DONE_STS bit before the next transfer occurs. When the interrupt handler clears the BYTE_DONE_STS bit after the last byte has been transferred, the 82801E C-ICH sets the INTR bit and generates another interrupt to signal the end of the block transfer. Thus, for a block message of n bytes, the 82801E C-ICH generates n+1 interrupts. The interrupt handler needs to be implemented to handle all of these interrupts

The format of the Block Read/Write protocol is shown in Table 74.

Note: For Block Write, if the I^2C_EN bit is set, the format of the command changes slightly. The 82801E C-ICH still sends the number of bytes indicated in the DATA0 register. However, it does not send the contents of the Data 0 register as part of the message.

Table 74. Block Read/Write Protocol

Block Write Protocol		Block Read Protocol	
Bit	**Description**	**Bit**	**Description**
1	Start	1	Start
2:8	Slave Address - 7 bits	2:8	Slave Address - 7 bits
9	Write	9	Write
10	Acknowledge from slave	10	Acknowledge from slave
11:18	Command code - 8 bits	11:18	Command code - 8 bits
19	Acknowledge from slave	19	Acknowledge from slave
20:27	Byte Count - 8 bits (Skip this step if I^2C_En bit set)	20	Repeated Start
28	Acknowledge from Slave (Skip this step if I2C_EN bit set)	21:27	Slave Address - 7 bits
29:36	Data Byte 1 - 8 bits	28	Read
37	Acknowledge from Slave	29	Acknowledge from slave
38:45	Data Byte 2–8 bits	30:37	Byte Count from slave - 8 bits
46	Acknowledge from slave	38	Acknowledge
...	Data Bytes/Slave Acknowledges...	39:46	Data Byte 1 from slave - 8 bits
...	Data Byte N - 8 bits	47	Acknowledge
...	Acknowledge from Slave	48:55	Data Byte 2 from slave - 8 bits
...	Stop	56	Acknowledge
		...	Data Bytes from slave/Acknowledge
		...	Data Byte N from slave - 8 bits
		...	NOT Acknowledge
		...	Stop

3.17.1.2 I²C Read

This command allows the 82801E C-ICH to perform block reads to certain I²C devices (e.g., serial E²PROMs). The SMBus Block Read sends the 7-bit address and the Command field. This command field could be used as the extended 10-bit address for accessing I²C devices that use 10-bit addressing.

However, this does not allow access to devices using the I²C "Combined Format" that has data bytes after the address. Typically, these data bytes correspond to an offset (address) within the serial memory chips.

Note: This new command is supported independent of the setting of the I²C_EN bit.

For I²C Read command, the value written into bit 0 of the Transmit Slave Address Register (SMB I/O register, offset 04h) needs to be 0. The format that is used for the new command is shown in Table 75:

Table 75. I²C Block Read

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

The 82801E C-ICH continues reading data from the peripheral until the NAK is received.

Note: The 82801E C-ICH uses the HST_D0 register (Dev 31, Func 3, Offset 05h) as the byte count register instead of depending on the LAST_BYTE bit in the Host Control register (Dev 31, Func 3, Offset 02h:[bit-5]) to end the transaction. The transaction will stop prematurely if HST_D0 contains a number smaller than the intended transaction. There is no workaround for 10-bit addressing of I²C devices. Designers can use the SM Bus read command for 7-bit addressing of I²C devices. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.

3.17.1.3 I²C Behavior

When the I²C_EN bit is set, the 82801E C-ICH SMBus logic is instead set to communicate with I²C devices. This forces the following changes:

1. The Process Call command skips the Command code (and its associated acknowledge)
2. The Block Write command does not send the Byte Count (DATA0)

In addition, the 82801E C-ICH supports the new I²C Read command. This is independent of the I²C_EN bit.

3.17.1.4 Heartbeat for Use With the External LAN Controller

This method allows the 82801E C-ICH to send messages to an *external* LAN Controller when the processor is otherwise unable to do so. It uses the SMLINK I/F between the 82801E C-ICH and the external LAN Controller. The actual Heartbeat message is a Block Write. Only 8 bytes are sent.

3.17.2 Bus Arbitration

Several masters may attempt to get on the bus at the same time by driving the SMBDATA line low to signal a start condition. The 82801E C-ICH continuously monitors the SMBDATA line. When the 82801E C-ICH attempts to drive the bus to a 1 by letting go of the SMBDATA line and it samples SMBDATA low, then some other master is driving the bus and the 82801E C-ICH stops transferring data.

When the 82801E C-ICH loses arbitration, the condition is called a collision. The 82801E C-ICH sets the BUS_ERR bit in the Host Status Register, and, if enabled, generates an interrupt or SMI#. The processor is responsible for restarting the transaction.

When the 82801E C-ICH is a SMBus master, it drives the clock. When the 82801E C-ICH is sending address or command as an SMBus master or data bytes as a master on writes, it drives data relative to the clock it is also driving. It does not start toggling the clock until the start or stop condition meets proper setup and hold time. The 82801E C-ICH also guarantees minimum time between SMBus transactions as a master.

The 82801E C-ICH supports the same arbitration protocol for both the SMBus and the System Management (SMLINK) interfaces.

Note: The 82801E C-ICH 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 82801E C-ICH may not set the Bus Error bit. A master attempting a transfer that had actually "lost" may think that its transaction was completed. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.

Clock Stretching

Some devices may not be able to handle their clock toggling at the rate that the 82801E C-ICH, as an SMBus master, would like. They have the capability of stretching the low time of the clock. When the 82801E C-ICH attempts to release the clock (allowing the clock to go high), the clock will remain low for an extended period of time.

The 82801E C-ICH monitors the SMBus clock line after it releases the bus to determine whether to enable the counter for the high time of the clock. While the bus is still low, the high time counter must not be enabled. Similarly, the low period of the clock can be stretched by an SMBus master if it is not ready to send or receive data.

The 82801E C-ICH SMBus Host Controller will never stretch the low period of the clock (SMBCLK). It always has the data to transfer on writes and it always has a spot for the data on reads.

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

Bus Time Out (82801E C-ICH as SMBus Master)

If there is an error in the transaction, such that an SMBus device does not signal an acknowledge or holds the clock lower than the allowed time-out time, the transaction times out. The 82801E C-ICH discards the cycle and sets the DEV_ERR bit. The time-out minimum is 25 ms. The time-out counter inside the 82801E C-ICH starts after the last bit of data is transferred by the 82801E C-ICH and it is waiting for a response. The 25 ms is a count of 800 RTC clocks.

3.17.3 Interrupts/SMI#

The 82801E C-ICH SMBus controller uses PIRQB# as its interrupt pin. However, the system can alternatively be set up to generate SMI# instead of an interrupt, by setting the SMBUS_SMI_EN bit.

3.17.4 SMBALERT#

SMBALERT# is multiplexed with GPIO[11]. When enabled and the signal is asserted, the 82801E C-ICH can generate an interrupt, an SMI#. To resume using SMBALERT#, the SMB_SMI_EN bit must be enabled to generate an SMI (see “HOSTC—Host Configuration Register (SMBUS—D31:F3)” on page 323).

Note: As long as SMBALERT# is enabled and asserted, the 82801E C-ICH will continue to assert PIRQ[B]# or SMI# (depending on the state of the SMB_SMI_EN bit). To avoid continuous SMIs or interrupts, the interrupt or SMI handler should:

1. Disable SMBALERT# by setting GPIO_USE_SEL[11] (GPIOBase + 00h, bit 11)
2. Use the SMBus Host Controller to service the peripheral that is asserting SMBALERT# (causing the device to deassert the signal)
3. Re-enable SMBALERT# by clearing GPIO_USE_SEL[11].

3.17.5 SMBus Slave Interface

The 82801E C-ICH’s SMBus Slave interface is accessed via the SMLINK[1:0] signals. The slave interface allows the 82801E C-ICH to decode cycles and allows an external microcontroller to perform specific actions. Key features and capabilities include:

- Supports decode of two messages type: Write and Read

- Receive Slave Address register: This is the address that the 82801E C-ICH decodes. A default value is provided so that the slave interface can be used without the processor having to program this register.
- Receive Slave Data register in the SMBus I/O space that includes the data written by the external microcontroller
- Registers that the external microcontroller can read to get the state of the 82801E C-ICH. See Table 80 on page 155
- Status bit to indicate that the SMBus logic caused an SMI# due to the reception of a message that matched the slave address. See "SMI_STS—SMI Status Register" on page 272.

Format of Slave Write Cycle

The external master performs Byte Write commands to the 82801E C-ICH SMBus Slave I/F. The "Command" field (bits 11-18) indicate which register is being accessed. The Data field (bits 20-27) indicate the value that should be written to that register.

The Write Cycle format is shown in Table 76. Table 77 lists the values associated with the registers.

Table 76. Slave Write Cycle Format

Bits	Description	Driven by	Comment
1	Start Condition	External Microcontroller	
2:8	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
9	Write	External Microcontroller	Always 0
10	ACK	82801E C-ICH	
11:18	Command	External Microcontroller	This field indicates which register will be accessed. See Table 77 below for the register definitions
19	ACK	82801E C-ICH	
20:27	Register Data	External Microcontroller	See Table 77 below for the register definitions
28	ACK	82801E C-ICH	
29	Stop	External Microcontroller	

Table 77. Slave Write Registers

Register	Function
0	Command Register. See Table 78 below for legal values written to this register.
1–3	Reserved
4	Data Message Byte 0
5	Data Message Byte 1
6–7	Reserved
8	Frequency Straps will be written on bits 3:0. Bits 7:4 should be 0, but will be ignored.
9–FFh	Reserved

NOTE: The external microcontroller is responsible to make sure that it does not update the contents of the data byte registers until they have been read by the system processor. The 82801E C-ICH overwrites the old value with any new value received. A race condition is possible where the new value is being written to the register just at the time it is being read. 82801E C-ICH will not attempt to cover this race condition (i.e., unpredictable results in this case).

Table 78. Command Types

Command Type	Description
0	Reserved.
1	**NOTE: The 82801E C-ICH is always "awake" since sleep states are not supported.** **WAKE/SMI#:** Wake system if it is not already awake. If the system is already awake, an SMI# is generated. The SMB_WAK_STS bit will be set by this command, even if the system is already awake. The SMI handler should then clear this bit.
2	**Unconditional Powerdown:** This command sets the PWRBTNOR_STS bit and has the same effect as the Powerbutton Override occurring. This functionality depends upon the BIOS having cleared the PWRBTN_STS bit.
3	**Hard Reset without Cycling:** This causes a hard reset of the system (does not include cycling of the power supply). This is equivalent to a write to the CF9h register with bits 2:1 set to 1, but bit 3 set to 0.
4	**Hard Reset System:** This causes a hard reset of the system (including cycling of the power supply). This is equivalent to a write to the CF9h register with bits 3:1 set to 1.
5	**Disable the TCO Messages.** This command disables the 82801E C-ICH from sending Heartbeat and Event messages (as described in "Alert on LAN*" on page 110). Once this command has been executed, Heartbeat and Event message reporting can only be re-enabled by assertion and deassertion of the RSMRST# signal.
6	**WD RELOAD:** Reload watchdog timer.
7–FFh	Reserved.

Format of Read Command

The external master performs Byte Read commands to the 82801E C-ICH SMBus Slave interface. The "Command" field (bits 11:18) indicates which register is being accessed. The Data field (bits 30:37) contain the value that should be read from that register. Table 79 shows the Read Cycle format. Table 80 shows the register mapping for the data byte.

Table 79. Read Cycle Format

Bit	Description	Driven by	Comment
1	Start	External Microcontroller	
2:8	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
9	Write	External Microcontroller	Always 0
10	ACK	82801E C-ICH	
11:18	Command code – 8 bits	External Microcontroller	Indicates which register is being accessed. See Table 80.
19	ACK	82801E C-ICH	
20	Repeated Start	External Microcontroller	
21:27	Slave Address - 7 bits	External Microcontroller	Must match value in Receive Slave Address register
28	Read	External Microcontroller	Always 1
29	ACK	82801E C-ICH	
30:37	Data Byte	82801E C-ICH	Value depends on register being accessed. See Table 80.
38	NOT ACK	External Microcontroller	
39	Stop	82801E C-ICH	

Table 80. Data Values for Slave Read Registers

Register	Bits	Description
0	7:0	Reserved.
1	2:0	System Power State 000 = S0 001 = Reserved 010 = Reserved 011 = Reserved 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved
1	7:3	Reserved.
2	3:0	Frequency Strap Register
2	7:4	Reserved.
3	5:0	Watchdog Timer current value
3	7:6	Reserved.
4	0	1 = The Intruder Detect (INTRD_DET) bit is set. This indicates that the system cover has probably been opened.
4	1	1 = BTI Temperature Event occurred. This bit is set if the 82801E C-ICH's THRM# input signal is active. Need to take after polarity control.
4	2	DOA processor status. This bit is 1 to indicate that the processor is dead.
4	3	1 = Watchdog timer expired. This bit is set if the 82801E C-ICH's TCO timers have timed out.
4	6:4	Reserved.
4	7	Will reflect the state of the 82801E C-ICH's GPIO[11].
5	0	Unprogrammed FWH bit. This bit will be 1 to indicate that the first BIOS fetch returned FFh, which indicates that the FWH is probably blank.
5	7:1	Reserved.
6	7:0	Contents of the Message 1 register. See Section 8.9.10.
7	7:0	Contents of the Message 2 register. See Section 8.9.10.
8	7:0	Contents of the WDSTATUS register. See Section 8.9.11.
9–FFh	7:0	Reserved.

Behavioral Notes

According to the SMBus protocol, Read and Write messages always begin with a Start bit - Address - Write bit sequence. When the 82801E C-ICH detects that the address matches the value in the Receive Slave Address register, it assumes that the protocol is always followed and ignores the Write bit (bit 9) and signals an Acknowledge during bit 10 (See Table 76 and Table 79). In other words, if a Start - Address - Read occurs (which is illegal for SMBus Read or Write protocol), and the address matches the 82801E C-ICH's Slave Address, the 82801E C-ICH will still perform the cycle.

Also according to SMBus protocol, a Read cycle contains a Repeated Start - Address - Read sequence beginning at bit 20 (See Table 79). Once again, if the Address matches the 82801E C-ICH's Receive Slave Address, it will assume that the protocol is followed, ignore bit 28, and proceed with the Slave Read cycle.

Note: An external microcontroller must not attempt to access the 82801E C-ICH's SMBus Slave logic until at least one second after both RTCRST# and RSMRST# are deasserted (high).

3.18 Firmware Hub Interface

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

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

3.18.1 Field Definitions

3.18.1.1 START

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

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

3.18.1.2 IDSEL (Device Select)

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

3.18.1.3 MSIZE (Memory Size)

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

3.18.1.4 MADDR (Memory Address)

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

3.18.1.5 SYNC

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

3.18.1.6 TAR

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

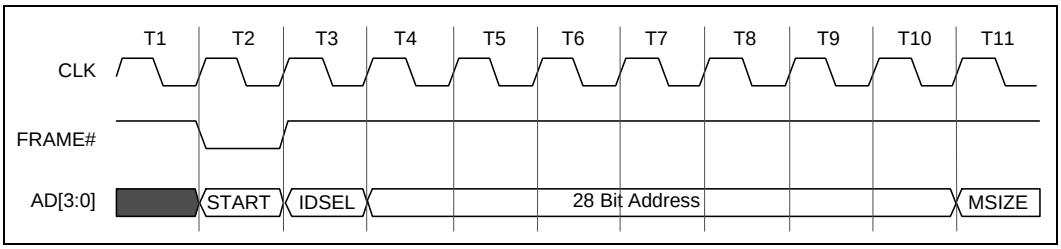
3.18.2 Protocol

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

3.18.2.1 Preamble

The initiation of the FWH Memory cycles is shown in Figure 23. The FWH Memory transaction begins with LFRAME# going low and a START field driven on AD[3:0]. For FWH Memory Read cycles, the START field must be '1101b'; for FWH Memory Write cycles, the START field must be '1110b'. Following the START field is the IDSEL field. This field acts like a chip select in that it indicates which device should respond to the current transaction. The next seven clocks are the 28-bit address from where to begin reading in the selected device. Next, an MSIZE value of 0 indicates the master is requesting a single byte.

Figure 23. FWH Memory Cycle Preamble



Read Cycle (Single Byte)

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

Figure 24. Single Byte Read

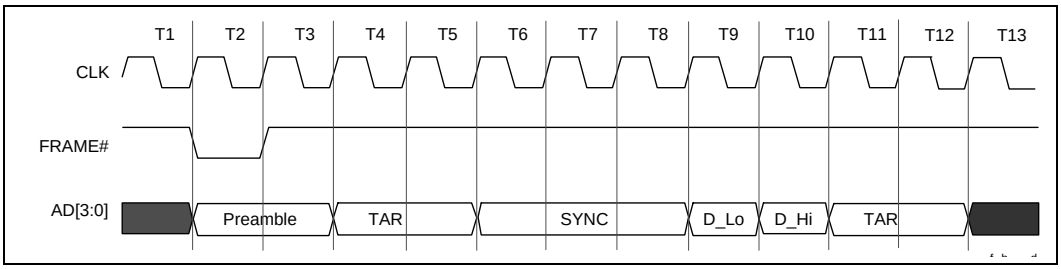


Figure 24 shows a device that requires three SYNC clocks to access data. Since the access time can begin once the address phase has been completed, the two clocks of the TAR phase can be considered as part of the access time of the part. For example, a device with a 120 ns access time could assert '0101b' for clocks 1 and 2 of the SYNC phase and '0000b' for the last clock of the

SYNC phase. This would be equivalent to five clocks worth of access time if the device started that access at the conclusion of the Preamble phase. Once SYNC is achieved, the device returns the data in two clocks and gives ownership of the bus back to the host with a TAR phase.

Write Cycles (Single Byte)

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

Figure 25. Single Byte Write

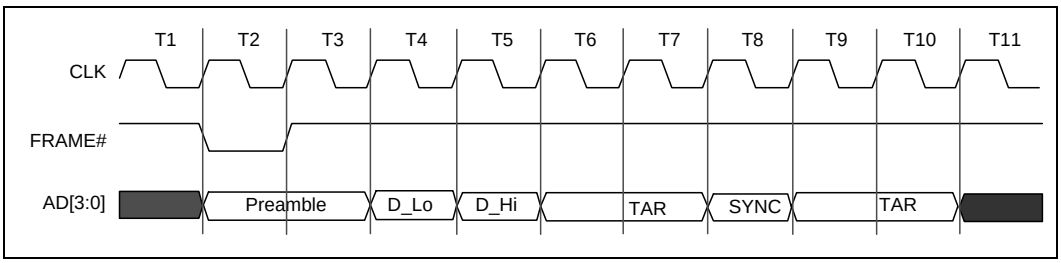


Figure 25 shows an FWH memory write cycle where a single byte is transferred. The master asserts an MSIZE value of 0. After the address has been transferred, the two clock data phase begins. Following the data phase, bus ownership is transferred to the FWH component with a TAR cycle. Following the TAR phase, the device must assert a SYNC value of '0000b' (ready) or '1010b' (error) indicating the data has been received. Bus ownership is then given back to the master with another TAR phase.

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

Error Reporting

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

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

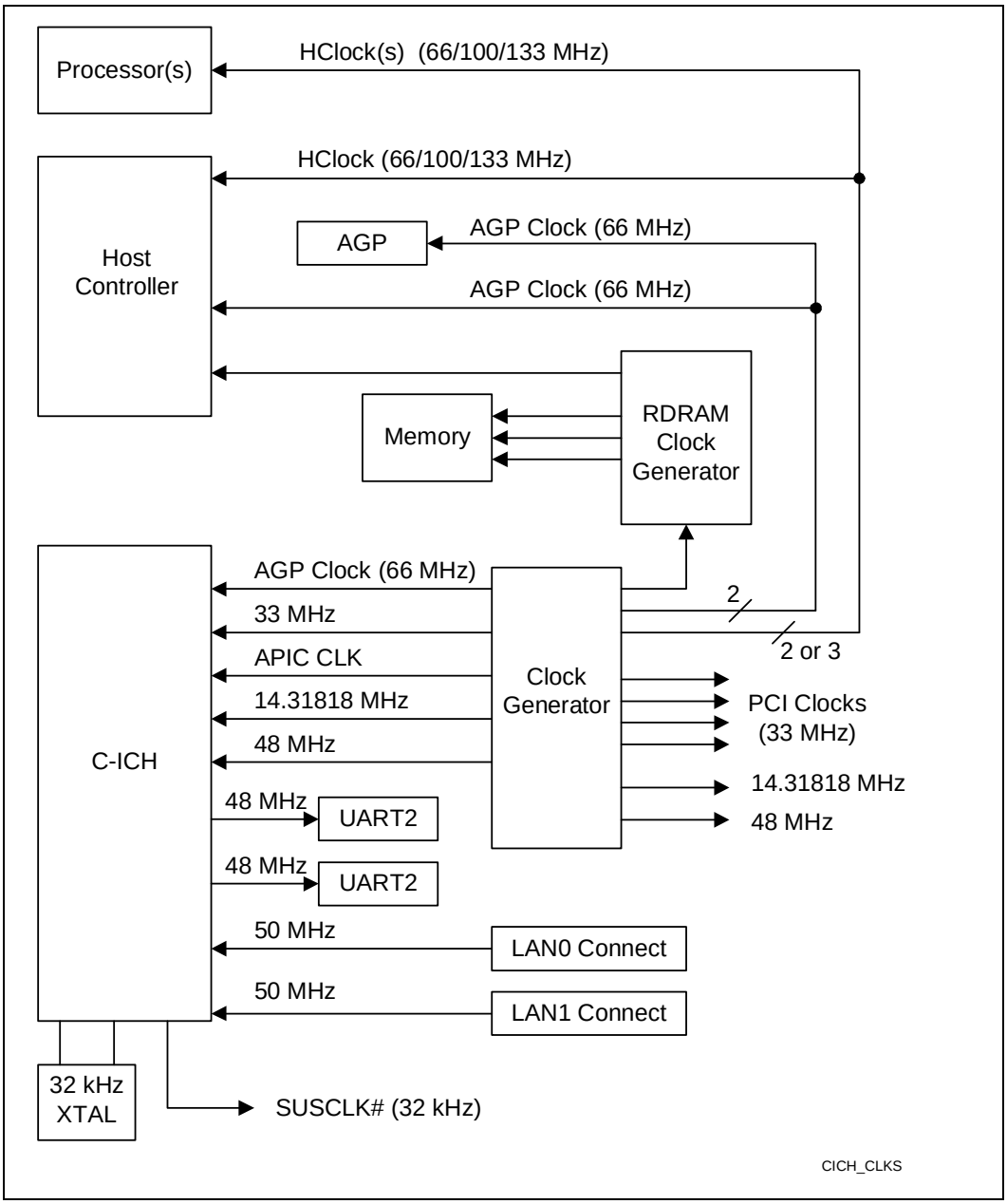
System Clock Domains 4

Table 26 shows the system clock domains. Figure 27 shows the assumed connection of the various system components, including the clock generator. For complete details of the system clocking solution, refer to the system's clock generator component specification.

Figure 26. System Clock Domains

Clock Domain	Frequency	Source	Usage
CLK66 (HLCLK)	66 MHz	Main Clock Generator	Hub interface, processor interface. AGP.
PCICLK	33 MHz	Main Clock Generator	Free-running PCI Clock to 82801E C-ICH.
System PCI	33 MHz	Main Clock Generator	PCI Bus, LPC I/F. These only go to external PCI and LPC devices.
CLK48	48 MHz	Main Clock Generator	Super I/O, USB Controller, Serial I/O units.
CLK14	14.31818 MHz	Main Clock Generator	Used for ACPI timer.
RTC	32.768 kHz	82801E C-ICH	RTC, Power Management. 82801E C-ICH has its own oscillator. Always running, even in G3 state.
APICCLK	33.33 MHz	Main Clock Generator	Used for 82801E C-ICH processor interrupt messages. Runs at 33.33 MHz.
LAN0_CLK LAN1_CLK	0.8 to 50 MHz	LAN Connect Component	Generated by the LAN Connect components.

Figure 27. Conceptual System Clock Diagram



Register and Memory Mapping 5

The 82801E C-ICH contains registers that are located in the processor's I/O space and memory space and sets of PCI configuration registers that are located in PCI configuration space. This chapter describes the 82801E C-ICH I/O and memory maps at the register-set level. Register access is also described. Register-level address maps and Individual register bit descriptions are provided in the following chapters. The following notations and definitions are used in the register/instruction description chapters.

RO Read Only. In some cases, if a register is read only, writes to this register location have no effect. However, in other cases, two separate registers are located at the same location where a read accesses one of the registers and a write accesses the other register. See the I/O and memory map tables for details.

WO Write Only. In some cases, if a register is write only, reads to this register location have no effect. However, in other cases, two separate registers are located at the same location where a read accesses one of the registers and a write accesses the other register. See the I/O and memory map tables for details.

R/W Read/Write. A register with this attribute can be read and written.

R/WC Read/Write Clear. A register bit with this attribute can be read and written. However, a write of 1 clears (sets to 0) the corresponding bit and a write of 0 has no effect.

Default When 82801E C-ICH is reset, it sets its registers to predetermined default states. The default state represents the minimum functionality feature set required to successfully bring up the system. Hence, it does not represent the optimal system configuration. It is the responsibility of the system initialization software to determine configuration, operating parameters, and optional system features that are applicable, and to program the 82801E C-ICH registers accordingly.

Bold Register bits that are highlighted in bold text indicate that the bit is implemented in the 82801E C-ICH. Register bits that are not implemented or are rewired are in plain text.

5.1 PCI Devices and Functions

The 82801E C-ICH incorporates a variety of PCI functions as shown in Table 81. These functions are divided into three logical devices (B0:D30, B0:D31 and B1:D8). D30 is the hub interface-to-PCI bridge, D31 contains the PCI-to-LPC Bridge, IDE Controller, USB Controller, and SMBus Controller functions. B1:D8/D9 are the integrated LAN Controllers.

Note: From a software perspective, the integrated LAN Controllers reside on the 82801E C-ICH's external PCI bus (See "PCI-to-PCI Bridge Model" on page 36). This is typically Bus 1, but may be assigned a different number depending on system configuration.

If a particular system platform does not want to support any one of Device 31's Functions 1–6, they can individually be disabled. Each integrated LAN Controller will be disabled if no Platform LAN Connect component is detected (See "LAN Controllers (B1:D8/D9:F0)" on page 40). When a function is disabled, it does not appear at all to the software. A disabled function will not respond to any register reads or writes. This is intended to prevent software from thinking that a function is present (and reporting it to the end-user).

Table 81. PCI Devices and Functions

Bus:Device:Function	Function Description
Bus 0:Device 30:Function 0	Hub Interface to PCI Bridge
Bus 0:Device 31:Function 0	PCI to LPC Bridge – The PCI to LPC bridge contains registers that control LPC, Power Management, System Management, GPIO, processor interface, RTC, Interrupts, Timers, DMA.
Bus 0:Device 31:Function 1	IDE Controller
Bus 0:Device 31:Function 2	USB Controller
Bus 0:Device 31:Function 3	SMBus Controller
Bus 1:Device 8:Function 0	LAN0 Controller
Bus 1:Device 9:Function 0	LAN1 Controller

5.2 PCI Configuration Map

Each PCI function on the 82801E C-ICH has a set of PCI configuration registers. The register address map tables for these register sets are included at the beginning of the chapter for the particular function.

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

Some of the PCI registers contain reserved bits. Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back. Note the software does not need to perform read, merge, write operation for the configuration address register.

In addition to reserved bits within a register, the configuration space contains reserved locations. Software should not write to reserved PCI configuration locations in the device-specific region (above address offset 3Fh).

5.3 I/O Map

The I/O map is divided into Fixed and Variable address ranges. Fixed ranges cannot be moved. In some cases they can be disabled. Variable ranges can be moved and can also be disabled.

5.3.1 Fixed I/O Address Ranges

Table 82 shows the fixed I/O decode ranges from the processor perspective. Note that for each I/O range, there may be a separate behavior for reads and writes. The hub interface cycles that go to target ranges that are marked as “Reserved” are not decoded by the 82801E C-ICH; they are passed to PCI. If a PCI master targets one of the fixed I/O target ranges, it will be positively decoded by the 82801E C-ICH in medium speed.

Refer to Table 147 for a complete list of all fixed I/O registers. Address ranges that are not listed or marked “Reserved” are NOT decoded by the 82801E C-ICH (unless they are assigned to one of the variable ranges).

Table 82. Fixed I/O Ranges Decoded by 82801E C-ICH (Sheet 1 of 2)

I/O Address	Read Target	Write Target	Internal Unit
00h–08h	DMA Controller	DMA Controller	DMA
09h–0Eh	RESERVED	DMA Controller	DMA
0Fh	DMA Controller	DMA Controller	DMA
10h–18h	DMA Controller	DMA Controller	DMA
19h–1Eh	RESERVED	DMA Controller	DMA
1Fh	DMA Controller	DMA Controller	DMA
20h–21h	Interrupt Controller	Interrupt Controller	Interrupt
24h–25h	Interrupt Controller	Interrupt Controller	Interrupt
28h–29h	Interrupt Controller	Interrupt Controller	Interrupt
2Ch–2Dh	Interrupt Controller	Interrupt Controller	Interrupt
2Eh–2Fh	LPC SIO	LPC SIO	Forwarded to LPC
30h–31h	Interrupt Controller	Interrupt Controller	Interrupt
34h–35h	Interrupt Controller	Interrupt Controller	Interrupt
38h–39h	Interrupt Controller	Interrupt Controller	Interrupt
3Ch–3Dh	Interrupt Controller	Interrupt Controller	Interrupt
40h–42h	Timer/Counter	Timer/Counter	PIT (8254)
43h	RESERVED	Timer/Counter	PIT
4E–4F	LPC SIO	LPC SIO	Forwarded to LPC
50h–52h	Timer/Counter	Timer/Counter	PIT
53h	RESERVED	Timer/Counter	PIT
60h	Microcontroller	Microcontroller	Forwarded to LPC
61h	NMI Controller	NMI Controller	processor I/F
62h	Microcontroller	Microcontroller	Forwarded to LPC
63h	NMI Controller	NMI Controller	processor I/F
64h	Microcontroller	Microcontroller	Forwarded to LPC
65h	NMI Controller	NMI Controller	processor I/F
66h	Microcontroller	Microcontroller	Forwarded to LPC
67h	NMI Controller	NMI Controller	processor I/F
70h	RESERVED[5]	NMI and RTC Controller	RTC
71h	RTC Controller	RTC Controller	RTC
72h	RTC Controller	NMI and RTC Controller	RTC

Table 82. Fixed I/O Ranges Decoded by 82801E C-ICH (Sheet 2 of 2)

I/O Address	Read Target	Write Target	Internal Unit
73h	RTC Controller	RTC Controller	RTC
74h	RTC Controller	NMI and RTC Controller	RTC
75h	RTC Controller	RTC Controller	RTC
76h	RTC Controller	NMI and RTC Controller	RTC
77h	RTC Controller	RTC Controller	RTC
80h	DMA Controller	DMA Controller and LPC or PCI	DMA
81h–83h	DMA Controller	DMA Controller	DMA
84h–86h	DMA Controller	DMA Controller and LPC or PCI	DMA
87h	DMA Controller	DMA Controller	DMA
88h	DMA Controller	DMA Controller and LPC or PCI	DMA
89h–8Bh	DMA Controller	DMA Controller	DMA
8Ch–8Eh	DMA Controller	DMA Controller and LPC or PCI	DMA
08Fh	DMA Controller	DMA Controller	DMA
90h–91h	DMA Controller	DMA Controller	DMA
92h	Reset Generator	Reset Generator	processor interface
93h–9Fh	DMA Controller	DMA Controller	DMA
A0h–A1h	Interrupt Controller	Interrupt Controller	Interrupt
A4h–A5h	Interrupt Controller	Interrupt Controller	Interrupt
A8h–A9h	Interrupt Controller	Interrupt Controller	Interrupt
ACh–ADh	Interrupt Controller	Interrupt Controller	Interrupt
B0h–B1h	Interrupt Controller	Interrupt Controller	Interrupt
B2h–B3h	Power Management	Power Management	Power Management
B4h–B5h	Interrupt Controller	Interrupt Controller	Interrupt
B8h–B9h	Interrupt Controller	Interrupt Controller	Interrupt
BCh–BDh	Interrupt Controller	Interrupt Controller	Interrupt
C0h–D1h	DMA Controller	DMA Controller	DMA
D2h–DDh	RESERVED	DMA Controller	DMA
DEh–DFh	DMA Controller	DMA Controller	DMA
F0h	See Note 3	FERR#/IGNNE#/Interrupt Controller	processor interface
170h–177h	IDE Controller[2]	IDE Controller[2]	Forwarded to IDE
1F0h–1F7h	IDE Controller[1]	IDE Controller[1]	Forwarded to IDE
376h	IDE Controller[2]	IDE Controller[2]	Forwarded to IDE
3F6h	IDE Controller[1]	IDE Controller[1]	Forwarded to IDE
4D0h–4D1h	Interrupt Controller	Interrupt Controller	Interrupt
CF9h	Reset Generator	Reset Generator	processor interface

NOTES:
1. Only if IDE Standard I/O space is enabled for Primary Drive. Otherwise, the target is PCI.
2. Only if IDE Standard I/O space is enabled for Secondary Drive. Otherwise, the target is PCI.
3. If POS_DEC_EN bit is enabled, reads from F0h will not be decoded by the 82801E C-ICH. If POS_DEC_EN is not enabled, reads from F0h will forward to LPC.

5.3.2 Variable I/O Decode Ranges

Table 83 shows the Variable I/O Decode Ranges. They are set using Base Address Registers (BARs) or other configuration bits in the various PCI configuration spaces. The PNP software (PCI or ACPI) can use their configuration mechanisms to set and adjust these values.

When a cycle is detected on the hub interface, the 82801E C-ICH positively decodes the cycle. If the response is on the behalf of an LPC device, 82801E C-ICH will forward the cycle to the LPC interface.

Refer to Table 148 for a complete list of all variable I/O registers.

Warning: The Variable I/O Ranges should not be set to conflict with the Fixed I/O Ranges. Results are unpredictable if the configuration software allows conflicts to occur. The 82801E C-ICH does not perform any checks for conflicts.

Table 83. Variable I/O Decode Ranges

Range Name	Mappable	Size (Bytes)	Target
ACPI	Anywhere in 64 Kbyte I/O Space	64	Power Management
IDE	Anywhere in 64 Kbyte I/O Space	16	IDE Unit
USB #1	Anywhere in 64 Kbyte I/O Space	32	USB Unit 1
SMBus	Anywhere in 64 Kbyte I/O Space	16	SMB Unit
TCO	96 Bytes above ACPI Base	32	TCO Unit
GPIO	Anywhere in 64 Kbyte I/O Space	64	GPIO Unit
Parallel Port	3 ranges in 64 Kbyte I/O Space	8	LPC Peripheral
Serial Port 1	8 Ranges in 64 Kbyte I/O Space	8	LPC Peripheral
Serial Port 2	8 Ranges in 64 Kbyte I/O Space	8	LPC Peripheral
Floppy Disk Controller	2 Ranges in 64 Kbyte I/O Space	8	LPC Peripheral
MIDI	4 Ranges in 64 Kbyte I/O Space	2	LPC Peripheral
MSS	4 Ranges in 64 Kbyte I/O Space	8	LPC Peripheral
SoundBlaster	2 Ranges in 64 Kbyte I/O Space	32	LPC Peripheral
LAN0/LAN1	Anywhere in 64 Kbyte I/O Space	64	LAN Unit
LPC Generic 1	Anywhere in 64 Kbyte I/O Space	128	LPC Peripheral
LPC Generic 2	Anywhere in 64 Kbyte I/O Space	16	LPC Peripheral
Monitors 4:7	Anywhere in 64 Kbyte I/O Space	16	LPC Peripheral or Trap on PCI

5.4 Memory Map

Table 84 shows (from the processor perspective) the memory ranges that the 82801E C-ICH decodes. Cycles that arrive from the MCH will first be driven out on PCI. The 82801E C-ICH may then claim the cycle for it to be forwarded to LPC or claimed by the internal APIC. If subtractive decode is enabled, the cycle can be forwarded to LPC.

PCI cycles generated by an external PCI master will be positively decoded unless it falls in the PCI-PCI bridge forwarding range (those addresses are reserved for PCI peer-to-peer traffic). If the cycle is not in the I/O APIC or LPC ranges, it will be forwarded up the hub interface to the Host Controller.

Table 84. Memory Decode Ranges from Processor Perspective (Sheet 1 of 2)

Memory Range	Target	Dependency/Comments
0000 0000h–000D FFFFh 0010 0000–TOM (Top of Memory)	Main Memory	TOM registers in Host Controller
000E 0000h–000F FFFFh	FWH	Bit 7 in FWH Decode Enable Register is set
FEC0 0000h–FEC0 0100h	I/O APIC inside 82801E C-ICH	
FFC0 0000h–FFC7 FFFFh FF80 0000h–FF87 FFFFh	FWH	Bit 0 in FWH Decode Enable Register
FFC8 0000h–FFCF FFFFh FF88 0000h–FF8F FFFFh	FWH	Bit 1 in FWH Decode Enable Register
FFD0 0000h–FFD7 FFFFh FF90 0000h–FF97 FFFFh	FWH	Bit 2 in FWH Decode Enable Register is set
FFD8 0000h–FFDF FFFFh FF98 0000h–FF9F FFFFh	FWH	Bit 3 in FWH Decode Enable Register is set
FFE0 000h–FFE7 FFFFh FFA0 0000h–FFA7 FFFFh	FWH	Bit 4 in FWH Decode Enable Register is set
FFE8 0000h–FFEF FFFFh FFA8 0000h–FFAF FFFFh	FWH	Bit 5 in FWH Decode Enable Register is set
FFF0 0000h–FFF7 FFFFh FFB0 0000h–FFB7 FFFFh	FWH	Bit 6 in FWH Decode Enable Register is set.
FFF8 0000h–FFFF FFFFh FFB8 0000h–FFBF FFFFh	FWH	Always enabled. The top two 64 Kbyte blocks of this range can be swapped as described in “Boot-Block Update Scheme” on page 167.
FF70 0000h–FF7F FFFFh FF30 0000h–FF3F FFFFh	FWH	Bit 3 in FWH Decode Enable 2 Register is set
FF60 0000h–FF6F FFFFh FF20 0000h–FF2F FFFFh	FWH	Bit 2 in FWH Decode Enable 2 Register is set
FF50 0000h–FF5F FFFFh FF10 0000h–FF1F FFFFh	FWH	Bit 1 in FWH Decode Enable 2 Register is set
FF40 0000h–FF4F FFFFh FF00 0000h–FF0F FFFFh	FWH	Bit 0 in FWH Decode Enable 2 Register is set

Table 84. Memory Decode Ranges from Processor Perspective (Sheet 2 of 2)

Memory Range	Target	Dependency/Comments
Anywhere in 4 Gbyte range	LAN0 Controller	Enable via BAR in Bus1:Device 8:Function 0 (LAN0 Controller)
Anywhere in 4 Gbyte range	LAN1 Controller	Enable via BAR in Bus1:Device 9:Function 0 (LAN1 Controller)
All other	PCI	None

5.4.1 Boot-Block Update Scheme

The 82801E C-ICH supports a "Top-Block Swap" mode that has the 82801E C-ICH swap the top block in the FWH (the boot block) with another location. This allows for safe update of the Boot Block (even if a power failure occurs). When the "top-swap" enable bit is set, the 82801E C-ICH will invert A16 for cycles targeting FWH BIOS space. When this bit is 0, the 82801E C-ICH will not invert A16. This bit is automatically set to 0 by RTCRST#, but not by PCIRST#.

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

The algorithm is:

1. Software copies the top block to the block immediately below the top
2. Software checks that the copied block is correct. This could be done by performing a checksum calculation.
3. Software sets the "Top-Block Swap" bit. This inverts A16 for cycles going to the FWH. Processor access to FFFF_0000 through FFFF_FFFF are directed to FFFE_0000 through FFFE_FFFF in the FWH. Processor accesses to FFFE_0000 through FFFE_FFFF are directed to FFFF_0000 through FFFF_FFFF.
4. Software erases the top block
5. Software writes the new top block
6. Software checks the new top block
7. Software clears the top-block swap bit

Note: The Top-Block Swap mode may be forced by an external strapping option (see the datasheet for details on strapping options). When Top-Block Swap mode is forced in this manner, the Top-Swap bit cannot be cleared by software. A re-boot with the strap removed will be required to exit a forced Top-Block Swap mode.

Note: Top-Block Swap mode only affects accesses to the FWH BIOS space, not feature space.

Note: The Top Block Swap mode has no effect on accesses below FFFE_0000.

Note: The Top Block Swap feature of the 82801E C-ICH does not work correctly if the PCI Delayed Transaction feature is enabled. Re-programming the Boot Block of the FWH will fail if the Top Block Swap feature is used with Delayed Transactions enabled. When updating BIOS, disable Delayed Transactions by clearing the DTE bit (D31:F0, offset D0h, bit 1) before setting the TOP_SWAP bit (D31:F0, offset D4h, bit 13). After the update is completed and verified, re-enable Delayed Transactions by setting DTE after clearing TOP_SWAP. See the *Intel® 82801E*

Communications I/O Controller Hub (C-ICH) Specification Update for information about steppings affected by this issue.

LAN Controller Registers (B1:D8/D9:F0) 6

Each LAN Controller acts as both a master and a slave on the PCI bus. As a master, the LAN Controller interacts with the system main memory to access data for transmission or deposit received data. As a slave, some LAN Controller control structures are accessed by the host CPU to read or write information to the on-chip registers. The CPU also provides the LAN Controller with the necessary commands and pointers that allow it to process receive and transmit data.

The LAN Controller appears in the 82801E C-ICH as Bus1: Device 8: Function 0, the other as Bus1:Device 9: Function 0. The LAN Controller PCI configuration space is configured as 16 Dwords of Type 0 Configuration Space Header, as defined in the PCI Specification, Revision 2.2. A small section is also configured according to its device specific configuration space.

6.1 PCI Configuration Registers (B1:D8/D9:F0)

Some of the PCI registers contain reserved bits. Software must deal correctly with fields that are reserved. On read, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions, and then written back. Note that the software does not need to perform read, merge, write operation for the configuration address register.

In addition to reserved bits within a register, the configuration space contains reserved location. Software should not write to reserved PCI configuration locations in the device-specific region (above address offset 3Fh).

Note: Registers that are not shown should be treated as Reserved (See “PCI Configuration Map” on page 162 for details).

Table 85. PCI Configuration Map (LAN Controller—B1:D8/D9:F0)

Offset	Mnemonic	Register Name/Function	Default	Type
00–01h	VID	Vendor ID	8086h	RO
02–03h	DID	Device 8 ID Device 9 ID	2459h 245Dh	RO
04–05h	PCICMD	PCI Device Command Register	0000h	R/W
06–07h	PCISTS	PCI Device Status Register	0290h	R/W
08h	REVID	Revision ID	Note 1	RO
0Ah	SCC	Sub Class Code	00h	RO
0Bh	BCC	Base Class Code	02h	RO
0Ch	CLS	Cache Line Size	00h	R/W
0Dh	PMLT	PCI Master Latency Timer	00h	R/W
0Eh	HEADTYP	Header Type	00h	RO
10–13h	CSR_MEM_BASE	CSR Memory-mapped Base Address	0008h	R/W
14–17h	CSR_IO_BASE	CSR I/O-mapped Base Address	0001h	R/W
2C–2Dh	SVID	Subsystem Vendor ID	0000h	RO
2E–2Fh	SID	Subsystem ID	0000h	RO
34h	CAP_PTR	Capabilities Pointer	DCh	RO
3Ch	INT_LN	Interrupt Line	00h	R/W
3Dh	INT_PN	Interrupt Pin	01h	RO
3Eh	MIN_GNT	Minimum Grant	08h	RO
3Fh	MAX_LAT	Maximum Latency	38h	RO
DCh	CAP_ID	Capability ID	01h	RO
DDh	NXT_PTR	Next Item Pointer	00h	RO
DE–DFh	PM_CAP	Power Management Capabilities	FE22h	RO
E0–E1h	PMCSR	Power Management Control/Status	0000h	R/W
E3h	DATA	Data	00h	RO

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

6.1.1 VID—Vendor ID Register (LAN Controller—B1:D8/D9:F0)

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

Bit	Description
15:0	**Vendor Identification Number.** This is a 16-bit value assigned to Intel.

6.1.2 DID—Device ID Register (LAN Controller—B1:D8/D9:F0)

Offset Address: 02–03h
Default Value: 2459h (Device 8)
245Dh (Device 9)
Attribute: RO
Size: 16 bits

Bit	Description
15:0	**Device Identification Number.** This is a 16 bit value assigned to the 82801E C-ICH integrated LAN Controller.

6.1.3 PCICMD—PCI Command Register (LAN Controller—B1:D8/D9:F0)

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

Bit	Description
15:10	Reserved.
9	**Fast Back to Back Enable (FBE)**—RO. Hardwired to 0. The integrated LAN Controller will not run fast back-to-back PCI cycles.
8	**SERR# Enable (SERR_EN)—**R/W. 1 = Enable. Allow SERR# to be generated. 0 = Disable.
7	**Wait Cycle Control (WCC)**—RO. Hardwired to 0. Not implemented.
6	**Parity Error Response (PER)**—R/W 1 = The integrated LAN Controller will take normal action when a PCI parity error is detected. The generation of parity is also enabled on the hub interface. 0 = The LAN Controller will ignore PCI parity errors.
5	**VGA Palette Snoop (VPS)**—RO. Hardwired to 0. Not Implemented.
4	**Memory Write and Invalidate Enable (MWIE)**—R/W. 0 = Disable. The LAN Controller will not use the Memory Write and Invalidate command. 1 = Enable.
3	**Special Cycle Enable (SCE)**—RO. Hardwired to 0. The LAN Controller ignores special cycles.
2	**Bus Master Enable (BME)**—R/W. 1 = Enable. The 82801E C-ICH's integrated LAN Controller may function as a PCI bus master. 0 = Disable.
1	**Memory Space Enable (MSE)**—R/W. 1 = Enable. The 82801E C-ICH's integrated LAN Controller will respond to the memory space accesses. 0 = Disable.
0	**I/O Space Enable (IOE)**—R/W. 1 = Enable. The 82801E C-ICH's integrated LAN Controller will respond to the I/O space accesses. 0 = Disable.

6.1.4 PCISTS—PCI Status Register (LAN Controller—B1:D8/D9:F0)

Offset Address: 06–07h
Default Value: 0290h
Attribute: RO, R/WC
Size: 16 bits

Bit	Description
15	**Detected Parity Error (DPE)**—R/WC. 1 = The 82801E C-ICH's integrated LAN Controller has detected a parity error on the PCI bus (will be set even if Parity Error Response is disabled in the PCI Command register). 0 = This bit is cleared by writing a 1 to the bit location.
14	**Signaled System Error (SSE)**—R/WC. 1 = The 82801E C-ICH's integrated LAN Controller has asserted SERR#. (SERR# can be routed to cause NMI, SMI# or interrupt. 0 = This bit is cleared by writing a 1 to the bit location.
13	**Master Abort Status (RMA)**—R/WC. 1 = The 82801E C-ICH's integrated LAN Controller (as a PCI master) has generated a master abort. 0 = This bit is cleared by writing a 1 to the bit location.
12	**Received Target Abort (RTA)**—R/WC. 1 = The 82801E C-ICH's integrated LAN Controller (as a PCI master) has received a target abort. 0 = This bit is cleared by writing a 1 to the bit location.
11	**Signaled Target Abort (STA)**—RO. Hardwired to 0. The device will never signal Target Abort.
10:9	**DEVSEL# Timing Status (DEV_STS)**—RO. 01h = Medium timing.
8	**Data Parity Error Detected (DPED)**—R/WC. 1 = All of the following three conditions have been met: 1. The LAN Controller is acting as bus master 2. The LAN Controller has asserted PERR# (for reads) or detected PERR# asserted (for writes) 3. The Parity Error Response bit in the LAN Controller's PCI Command Register is set. 0 = This bit is cleared by writing a 1 to the bit location.
7	**Fast Back to Back (FB2B)**—RO. Hardwired to 1. The device can accept fast back-to-back transactions.
6	**User Definable Features (UDF)**—RO. Hardwired to 0. Not implemented.
5	**66 MHz Capable (66MHZ_CAP)**—RO. Hardwired to 0. The device does not support 66 MHz PCI.
4	**Capabilities List (CAP_LIST)**—RO. 1 = The EEPROM indicates that the integrated LAN controller supports PCI Power Management. 0 = The EEPROM indicates that the integrated LAN controller does not support PCI Power Management.
3:0	Reserved.

6.1.5 REVID—Revision ID Register (LAN Controller—B1:D8/D9:F0)

Offset Address: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

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

6.1.6 SCC—Sub-Class Code Register (LAN Controller—B1:D8/D9:F0)

Offset Address: 0Ah Attribute: RO
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**Sub Class Code.** 8-bit value that specifies the sub-class of the device as an Ethernet controller.

6.1.7 BCC—Base-Class Code Register (LAN Controller—B1:D8/D9:F0)

Offset Address: 0Bh Attribute: RO
Default Value: 02h Size: 8 bits

Bit	Description
7:0	**Base Class Code.** 8-bit value that specifies the base class of the device as a network controller.

6.1.8 CLS—Cache Line Size Register (LAN Controller—B1:D8/D9:F0)

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

Bit	Description
7:5	Reserved.
4:3	**Cache Line Size (CLS)**—RW. 00 = Memory Write and Invalidate (MWI) command will not be used by the integrated LAN Controller. 01 = MWI command will be used with Cache Line Size set to 8 DWords (only set if a value of 08h is written to this register). 10 = MWI command will be used with Cache Line Size set to 16 DWords (only set if a value of 10h is written to this register). 11 = Invalid. MWI command will not be used.
2:0	Reserved.

6.1.9 PMLT—PCI Master Latency Timer Register (LAN Controller—B1:D8/D9:F0)

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

Bit	Description
7:3	**Master Latency Timer Count (MLTC)**—RW. Defines the number of PCI clock cycles that the integrated LAN Controller may own the bus while acting as bus master.
2:0	Reserved.

6.1.10 HEADTYP—Header Type Register (LAN Controller—B1:D8/D9:F0)

Offset Address: 0Eh Attribute: RO
Default Value: 00h Size: 8 bits

Bit	Description
7	**Multi-function Device**—RO. Hardwired to 0 to indicate a single function device.
6:0	**Header Type**—RO. 7-bit field identifies the header layout of the configuration space as an Ethernet controller.

6.1.11 CSR_MEM_BASE CSR—Memory-Mapped Base Address Register (LAN Controller—B1:D8/D9:F0)

Offset Address: 10–13h Attribute: R/W, RO
Default Value: 0000 0008h Size: 32 bits

Note: The 82801E C-ICH's integrated LAN Controller requires one BAR for memory mapping. Software determines which BAR (memory or I/O) is used to access the Lan Controller's CSR registers.

Bit	Description
31:12	**Base Address**—R/W. Upper 20 bits of the base address provides 4 Kbytes of memory-mapped space for the LAN Controller's Control/Status Registers.
11:4	Reserved.
3	**Pre-fetchable**—RO. Hardwired to 0 to indicate that this is not a pre-fetchable memory-mapped address range.
2:1	**Type**—RO. Hardwired to 00b to indicate the memory-mapped address range may be located anywhere in 32-bit address space.
0	**Memory-Space Indicator**—RO. Hardwired to 0 to indicate that this base address maps to memory space.

6.1.12 CSR_IO_BASE—CSR I/O-Mapped Base Address Register (LAN Controller—B1:D8/D9:F0)

Offset Address:	14–17h	Attribute:	R/W
Default Value:	0000 0001h	Size:	32 bits

Note: The 82801E C-ICH's integrated LAN Controller requires one BAR for memory mapping. Software determines which BAR (memory or I/O) is used to access the Lan Controller's CSR registers.

Bit	Description
31:16	Reserved.
15:6	**Base Address**—R/W. Provides 64 bytes of I/O-mapped address space for the LAN Controller's Control/Status Registers.
5:1	Reserved.
0	**I/O Space Indicator**—RO. Hardwired to 1 to indicate that this base address maps to I/O space.

6.1.13 SVID—Subsystem Vendor ID (LAN Controller—B1:D8/D9:F0)

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

Bit	Description
15:0	**Subsystem Vendor ID**—RO.

6.1.14 SID—Subsystem ID (LAN Controller—B1:D8/D9:F0)

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

Bit	Description
15:0	**Subsystem ID**—RO.

Note: The 82801E C-ICH's integrated LAN Controller provides support for configureable Subsystem ID and Subsystem Vendor ID fields. After reset, the LAN Controller automatically reads addresses Ah through Ch of the EEPROM. The LAN Controller checks bits 15:13 in the EEPROM word Ah, and functions according to Table 86.

Table 86. Configuration of Subsystem ID and Subsystem Vendor ID via EEPROM

Bits 15:14	Bit 13	Device ID	Vendor ID	Revision ID	Subsystem ID	Subsystem Vendor ID
11b, 10b, 00b	X	2459h (Device 8) 245Dh (Device 9)	8086h	00h	0000h	0000h
01b	0b	2459h (Device 8) 245Dh (Device 9)	8086h	00h	Word Bh	Word Ch
01b	1b	Word Bh	Word Ch	Word Ah, bits 10:8	Word Bh	Word Ch

6.1.15 CAP_PTR—Capabilities Pointer (LAN Controller—B1:D8/D9:F0)

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

Bit	Description
7:0	**Capabilities Pointer (CAP_PTR)**—RO. Hardwired to DCh to indicate the offset within configuration space for the location of the Power Management registers.

6.1.16 INT_LN—Interrupt Line Register (LAN Controller—B1:D8/D9:F0)

Offset Address: 3Ch Attribute: R/W
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**Interrupt Line (INT_LN)**—R/W. Identifies the system interrupt line to which the LAN Controller's PCI interrupt request pin (as defined in the Interrupt Pin Register) is routed.

6.1.17 INT_PN—Interrupt Pin Register (LAN Controller—B1:D8/D9:F0)

Offset Address: 3Dh Attribute: RO
Default Value: 01h Size: 8 bits

Bit	Description
7:0	**Interrupt Pin (INT_PN)**—RO. Hardwired to 01h to indicate that the LAN Controller's interrupt request is connected to PIRQA#. However, in the 82801E C-ICH implementation, when the LAN Controller interrupt is generated PIRQ[E]# will go active, not PIRQ[A]#.

6.1.18 MIN_GNT—Minimum Grant Register (LAN Controller—B1:D8/D9:F0)

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

Bit	Description
7:0	**Minimum Grant (MIN_GNT)**—RO. Indicates the amount of time (in increments of 0.25 µs) that the LAN Controller needs to retain ownership of the PCI bus when it initiates a transaction.

6.1.19 MAX_LAT—Maximum Latency Register (LAN Controller—B1:D8/D9:F0)

Offset Address: 3Fh Attribute: RO
Default Value: 38h Size: 8 bits

Bit	Description
7:0	**Maximum Latency (MAX_LAT)**—RO. Defines how often (in increments of 0.25 µs) the LAN Controller needs to access the PCI bus.

6.1.20 CAP_ID—Capability ID Register (LAN Controller—B1:D8/D9:F0)

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

Bit	Description
7:0	**Capability ID (CAP_ID)**—RO. Hardwired to 01h to indicate that the 82801E C-ICH's integrated LAN Controller supports PCI Power Management.

6.1.21 NXT_PTR—Next Item Pointer (LAN Controller—B1:D8/D9:F0)

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

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

6.1.22 PM_CAP—Power Management Capabilities (LAN Controller—B1:D8/D9:F0)

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

This register is a word read only register; it reports a value of FE22h.

6.1.23 PMCSR—Power Management Control/Status Register (LAN Controller—B1:D8/D9:F0)

Offset Address:	E0–E1h	Attribute:	RO, R/W, R/WC
Default Value:	0000h	Size:	16 bits

Bit	Description
15	Reserved.
14:13	**Data Scale—**RO. This field indicates the data register scaling factor. It equals 10b for registers zero through eight and 00b for registers nine through fifteen, as selected by the “Data Select” field.
12:9	**Data Select—**R/W. This field is used to select which data is reported through the Data register and Data Scale field.
8	**PME Enable**—R/W. Not implemented. Must be set to 0.
7:5	Reserved.
4	**Dynamic Data—**RO. Hardwired to 0 to indicate that the device does not support the ability to monitor the power consumption dynamically.
3:0	Reserved. Should be set to 0000b.

6.1.24 DATA—Data Register (LAN Controller—B1:D8/D9:F0)

Offset Address:	E3h	Attribute:	RO
Default Value:	00h	Size:	8 bits

Bit	Description
7:0	**Data Value**. State dependent power consumption and heat dissipation data.

Note: The data register is an 8-bit read only register that provides a mechanism for the 82801E C-ICH’s integrated LAN Controller to report state dependent maximum power consumption and heat dissipation. The value reported in this register depends on the value written to the Data Select field in the PMCSR register. The power measurements defined in this register have a dynamic range of 0 to 2.55 W with 0.01 W resolution, scaled according to the Data Scale field in the PMCSR. The structure of the Data Register is given in Table 87.

Table 87. Data Register Structure

Data Select	Data Scale	Data Reported
0	2	D0 Power Consumption
1	2	D1 Power Consumption
2	2	D2 Power Consumption
3	2	D3 Power Consumption
4	2	D0 Power Dissipated
5	2	D1 Power Dissipated
6	2	D2 Power Dissipated
7	2	D3 Power Dissipated
8	2	Common Function Power Dissipated
9–15	0	Reserved

6.2 LAN Control/Status Registers (CSR)

Table 88. Intel® 82801E C-ICH Integrated LAN Controller CSR Space

Offset	Register Name/Function	Default	Type
00h–01h	SCB Status Word	0000h	R/WC
02h–03h	SCB Command Word	0000h	R/W
04h–07h	SCB General Pointer	0000 0000h	R/W
08h–0Bh	PORT	0000 0000h	R/W (special)
0Ch–0Dh	Reserved	—	—
0Eh	EEPROM Control Register	00h	R/W
0Fh	Reserved	—	—
10h–13h	MDI Control Register	0000 0000h	R/W (special)
14h–17h	Receive DMA Byte Count	0000 0000h	RO
18h	Early Receive Interrupt	00h	R/W
19h–1Ah	Flow Control Register	0000h	R/W
1Bh	PMDR	00h	R/WC
1Ch	General Control	00h	R/W
1Dh	General Status	N/A	RO
1Ch–3Eh	Reserved	—	—

6.2.1 System Control Block Status Word Register

Offset Address: 00–01h
Default Value: 0000h
Attribute: R/WC, RO
Size: 16 bits

The 82801E C-ICH's integrated LAN Controller places the status of its Command and Receive units and interrupt indications in this register for the processor to read.

Bit	Description
15	**Command Unit (CU) Executed (CX)**—R/WC. 1 = Interrupt signaled because the CU has completed executing a command with its interrupt bit set. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.
14	**Frame Received (FR)**—R/WC. 1 = Interrupt signaled because the Receive Unit (RU) has finished receiving a frame 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.
13	**CU Not Active (CNA)**—R/WC. 1 = The Command Unit left the Active state or entered the Idle state. There are 2 distinct states of the CU. When configured to generate CNA interrupt, the interrupt will be activated when the CU leaves the Active state and enters either the Idle or the Suspended state. When configured to generate CI interrupt, an interrupt will be generated only when the CU enters the Idle state. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.
12	**Receive Not Ready (RNR)**—R/WC. 1 = Interrupt signaled because the Receive Unit left the Ready state. This may be caused by an RU Abort command, a no resources situation, or set suspend bit due to a filled Receive Frame Descriptor. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.
11	**Management Data Interrupt (MDI)**—R/WC. 1 = Set when a Management Data Interface read or write cycle has completed. The management data interrupt is enabled through the interrupt enable bit (bit 29 in the Management Data Interface Control register in the CSR). 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.
10	**Software Interrupt (SWI)**—R/WC. 1 = Set when software generates an interrupt. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.
9	**Early Receive (ER)**—R/WC. 1 = Indicates the occurrence of an Early Receive Interrupt. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.
8	**Flow control Pause (FCP)**—R/WC. 1 = Indicates Flow Control Pause interrupt. 0 = Software acknowledges the interrupt and clears this bit by writing a 1 to the bit position.

Bit	Description
7:6	**Command Unit Status (CUS)**—RO. 00 = Idle 01 = Suspended 10 = LPQ (Low Priority Queue) active 11 = HPQ (High Priority Queue) active
5:2	**Receive Unit Status (RUS)**—RO. 0000 = Idle 0001 = Suspended 0010 = No Resources 0011 = Reserved 0100 = Ready 0101 = Reserved 0110 = Reserved 0111 = Reserved 1000 = Reserved 1001 = Suspended with no more RBDs 1010 = No resources due to no more RBDs 1011 = Reserved 1100 = Ready with no RBDs present 1101 = Reserved 1110 = Reserved 1111 = Reserved
1:0	Reserved.

6.2.2 System Control Block Command Word Register

Offset Address: 02–03h
Default Value: 0000h
Attribute: R/W
Size: 16 bits

The processor places commands for the Command and Receive units in this register. Interrupts are also acknowledged in this register.

Bit	Description
15	**CX Mask**—R/W. 0 = Interrupt not masked. 1 = Disable the generation of a CX interrupt.
14	**FR Mask**—R/W. 0 = Interrupt not masked. 1 = Disable the generation of an FR interrupt.
13	**CNA Mask**—R/W. 0 = Interrupt not masked. 1 = Disable the generation of a CNA interrupt.
12	**RNR Mask**—R/W. 0 = Interrupt not masked. 1 = Disable the generation of an RNR interrupt.
11	**ER Mask**—R/W. 0 = Interrupt not masked. 1 = Disable the generation of an ER interrupt.
10	**FCP Mask**—R/W. 0 = Interrupt not masked. 1 = Disable the generation of an FCP interrupt.
9	**Software Generated Interrupt (SI)**—WO. 0 = No Effect. 1 = Setting this bit causes the LAN Controller to generate an interrupt.
8	**Interrupt Mask (IM)**—R/W. This bit enables or disables the LAN Controller's assertion of the INTA# signal. This bit has higher precedence that the Specific Interrupt Mask bits and the SI bit. 0 = Enable the assertion of INTA#. 1 = Disable the assertion of INTA#.

Bit	Description
7:4	**Command Unit Command (CUC).** Valid values are listed below. All other values are Reserved. 0000 = **NOP:** Does not affect the current state of the unit. 0001 = **CU Start**: Start execution of the first command on the CBL. A pointer to the first CB of the CBL should be placed in the SCB General Pointer before issuing this command. The CU Start command should only be issued when the CU is in the Idle or Suspended states (never when the CU is in the active state), and all of the previously issued Command Blocks have been processed and completed by the CU. Sometimes it is only possible to determine that all Command Blocks are completed by checking that the Complete bit is set in all previously issued Command Blocks. 0010 = **CU Resume:** Resume operation of the Command unit by executing the next command. This command will be ignored if the CU is idle. 0011 = **CU HPQ Start:** Start execution of the first command on the high priority CBL. A pointer to the first CB of the HPQ CBL should be placed in the SCB General POinter before issuing this command. 0100 = **Load Dump Counters Address:** Tells the device where to write dump data when using the Dump Statistical Counters or Dump and Reset Statistical Counters commands. This command must be executed at least once before any usage of the Dump Statistical Counters or Dump and Reset Statistical Counters commands. The address of the dump area must be placed in the General Pointer register. 0101 = **Dump Statistical Counters:** Tells the device to dump its statistical counters to the area designated by the Load Dump Counters Address command. 0110 = **Load CU Base:** The device's internal CU Base Register is loaded with the value in the CSB General Pointer. 0111 = **Dump and Reset Statistical Counters:** Tells the device to dump its statistical counters to the area designated by the Load Dump Counters Address command, and then to clear these counters. 1010 = **CU Static Resume:** Resume operation of the Command unit by executing the next command. This command will be ignored if the CU is idle. This command should be used only when the CU is in the Suspended state and has no pending CU Resume commands. 1011 = **CU HPQ Resume:** Resume execution of the first command on the HPQ CBL. This command will be ignored if the HPQ was never started.
3	Reserved.
2:0	**Receive Unit Command (RUC).** Valid values are: 000 = **NOP:** Does not affect the current state of the unit. 001 = **RU Start:** Enables the receive unit. The pointer to the RFA must be placed in the SCB General POinter before using this command. The device pre-fetches the first RFD and the first RBD (if in flexible mode) in preparation to receive incoming frames that pass its address filtering. 010 = **RU Resume:** Resume frame reception (only when in suspended state). 011 = **RCV DMA Redirect:** Resume the RCV DMA when configured to "Direct DMA Mode." The buffers are indicated by an RBD chain that is pointed to by an offset stored in the General Pointer Register (this offset will be added to the RU Base). 100 = **RU Abort:** Abort RU receive operation immediately. 101 = **Load Header Data Size (HDS):** This value defines the size of the Header portion of the RFDs or Receive buffers. The HDS value is defined by the lower 14 bits of the SCB General Pointer, so bits 31:15 should always be set to zeros when using this command. Once a Load HDS command is issued, the device expects only to find Header RFDs, or be used in "RCV Direct DMA mode" until it is reset. Note that the value of HDS should be an even, non-zero number. 110 = **Load RU Base:** The device's internal RU Base Register is loaded with the value in the SCB General Pointer. 111 = **RBD Resume:** Resume frame reception into the RFA. This command should only be used when the RU is already in the "No Resources due to no RBDs" state or the "Suspended with no more RBDs" state.

6.2.3 System Control Block General Pointer Register

Offset Address: 04–07h Attribute: R/W
Default Value: 0000 0000h Size: 32 bits

Bit	Description
15:0	**SCB General Pointer.** The SCB General Pointer register is programmed by software to point to various data structures in main memory depending on the current SCB Command word.

6.2.4 PORT Register

Offset Address: 08–0Bh Attribute: R/W (special)
Default Value: 0000 0000h Size: 32 bits

The PORT interface allows the processor to reset the 82801E C-ICH's internal LAN Controller or perform an internal self test. The PORT DWord may be written as a 32-bit entity, two 16-bit entities, or four 8-bit entities. The LAN Controller will only accept the command after the high byte (offset 0Bh) is written; therefore, the high byte must be written last.

Bit	Description
31:4	**Pointer Field.** A 16-byte aligned address must be written to this field when issuing a Self-Test command to the PORT interface.The results of the Self Test will be written to the address specified by this field.
3:0	**PORT Function Selection.** Valid values are listed below. All other values are Reserved. 0000 = **PORT Software Reset:** Completely resets the LAN Controller (all CSR and PCI registers). This command should not be used when the device is active. If a PORT Software Reset is desired, software should do a Selective Reset (described below), wait for the PORT register to be cleared (completion of the Selective Reset) and then issue the PORT Software Reset command. Software should wait approximately 10 µs after issuing this command before attempting to access the LAN Controller's registers again. 0001 = **Self Test:** The Self-Test begins by issuing an internal Selective Reset followed by a general internal self-test of the LAN Controller. The results of the self-test are written to memory at the address specified in the Pointer field of this register. The format of the self-test result is shown in Table 89. After completing the self-test and writing the results to memory, the LAN Controller will execute a full internal reset and will re-initialize to the default configuration. Self-Test does not generate an interrupt of similar indicator to the host processor upon completion. 0010 = **Selective Reset:** Sets the CU and RU to the Idle state, but otherwise maintains the current configuration parameters (RU and CU Base, HDSSize, Error Counters, Configure information and Individual/Multicast Addresses are preserved). Software should wait approximately 10 µs after issuing this command before attempting to access the LAN Controller's registers again.

Table 89. Self-Test Results Format

Bit	Description
31:13	Reserved.
12	**General Self-Test Result.** 0 = Pass 1 = Fail
11:6	Reserved.
5	**Diagnose Result.** This bit provides the result of an internal diagnostic test of the Serial Subsystem. 0 = Pass 1 = Fail
4	Reserved.
3	**Register Result.** This bit provides the result of a test of the internal Parallel Subsystem registers. 0 = Pass 1 = Fail
2	**ROM Content Result.** This bit provides the result of a test of the internal microcode ROM. 0 = Pass 1 = Fail
1:0	Reserved.

6.2.5 EEPROM Control Register

Offset Address: 0Eh Attribute: RO, R/W
Default Value: 00h Size: 8 bits

The EEPROM Control Register is a 16-bit field that enables a read from and a write to the external EEPROM.

Bit	Description
7:4	Reserved.
3	**EEPROM Serial Data Out (EEDO)**—RO. Note that this bit represents “Data Out” from the perspective of the EEPROM device. This bit contains the value read from the EEPROM when performing read operations.
2	**EEPROM Serial Data In (EEDI)**—WO. Note that this bit represents "Data In" from the perspective of the EEPROM device. The value of this bit is written to the EEPROM when performing write operations.
1	**EEPROM Chip Select (EECS)**—R/W. 0 = Drives the 82801E C-ICH’s EE_CS signal low, to disable the EEPROM. this bit must be set to 0 for a minimum of 1µs between consecutive instruction cycles. 1 = Drives the 82801E C-ICH’s EE_CS signal high, to enable the EEPROM.
0	**EEPROM Serial Clock (EESK)**—R/W. Toggling this bit clocks data into or out of the EEPROM. Software must ensure that this bit is toggled at a rate that meets the EEPROM component’s minimum clock frequency specification. 0 = Drives the 82801E C-ICH’s EE_SHCLK signal low. 1 = Drives the 82801E C-ICH’s EE_SHCLK signal high.

6.2.6 Management Data Interface (MDI) Control Register

Offset Address:	10–13h	Attribute:	R/W (special)
Default Value:	0000 0000h	Size:	32 bits

The Management Data Interface (MDI) Control register is a 32-bit field and is used to read and write bits from the LAN Connect component. This register may be written as a 32-bit entity, two 16-bit entities, or four 8-bit entities. The LAN Controller will only accept the command after the high byte (offset 13h) is written; therefore, the high byte must be written last.

Bit	Description
31:30	Reserved. These bits are reserved and should be set to 00b.
29	**Interrupt Enable.** 1 = Enables the LAN Controller to assert an interrupt to indicate the end of an MDI cycle. 0 = Disable.
28	**Ready.** 1 = Set by the LAN Controller at the end of an MDI transaction. 0 = Expected to be reset by software at the same time the command is written.
27:26	**Opcode.** These bits define the opcode: 00 = Reserved 01 = MDI write 10 = MDI read 11 = Reserved
25:21	**LAN Connect Address.** This field of bits contains the LAN Connect address.
20:16	**LAN Connect Register Address.** This field of bits contains the LAN Connect Register Address.
15:0	**Data.** In a write command, software places the data bits in this field, and the LAN Controller transfers the data to the external LAN Connect component. During a read command, the LAN Controller reads these bits serially from the LAN Connect, and software reads the data from this location.

6.2.7 Receive DMA Byte Count Register

Offset Address:	14–17h	Attribute:	RO
Default Value:	0000 0000h	Size:	32 bits

Bit	Description
31:0	**Receive DMA Byte Count**—RO. Keeps track of how many bytes of receive data have been passed into host memory via DMA.

6.2.8 Early Receive Interrupt Register

Offset Address:	18h	Attribute:	R/W
Default Value:	00h	Size:	8 bits

The Early Receive Interrupt register allows the internal LAN Controller to generate an early interrupt depending on the length of the frame. The LAN Controller will generate an interrupt at the end of the frame, regardless of whether or not Early Receive Interrupts are enabled.

Note: It is recommended that software NOT utilize this register unless receive interrupt latency is a critical performance issue in that particular software environment. Using this feature may reduce receive interrupt latency, but will also result in the generation of more interrupts, which can degrade system efficiency and performance in some environments.

Bit	Description
7:0	**Early Receive Count**—R/W. When some non-zero value ***x*** is programmed into this register, the LAN controller sets the ER bit in the SCB Status Word Register and assert INTA# when the byte count indicates that there are ***x*** quadwords remaining to be received in the current frame (based on the Type/Length field of the received frame). No Early Receive interrupt will be generated if a value of 00h (the default value) is programmed into this register.

6.2.9 Flow Control Register

Offset Address:	19–1Ah	Attribute:	RO, R/W (special)
Default Value:	0000h	Size:	16 bits

Bit	Description
15:13	Reserved.
12	**FC Paused Low**—RO. 1 = Set when the LAN Controller receives a Pause Low command with a value greater than zero. 0 = Cleared when the FC timer reaches zero, or a Pause frame is received.
11	**FC Paused**—RO. 1 = Set when the LAN Controller receives a Pause command regardless of its cause (FIFO reaching Flow Control Threshold, fetching a Receive Frame Descriptor with its Flow Control Pause bit set, or software writing a 1 to the Xoff bit). 0 = Cleared when the FC timer reaches zero.
10	**FC Full**—RO. 1 = Set when the LAN Controller sends a Pause command with a value greater than zero. 0 = Cleared when the FC timer reaches zero.
9	**Xoff**—R/W (special). This bit should only be used if the LAN Controller is configured to operate with IEEE frame-based flow control. 1 = Writing a 1 to this bit forces the Xoff request to 1 and causes the LAN Controller to behave as if the FIFO extender is full. This bit will also be set to 1 when an Xoff request due to an "RFD Xoff" bit. 0 = This bit can only be cleared by writing a 1 to the Xon bit (bit 8 in this register).

Bit	Description
8	**Xon**—WO. This bit should only be used if the LAN Controller is configured to operate with IEEE frame-based flow control. 1 = Writing a 1 to this bit resets the Xoff request to the LAN Controller, clearing bit 9 in this register. 0 = This bit always returns 0 on reads.
7:3	Reserved.
2:0	**Flow Control Threshold**—R/W. The LAN Controller can generate a Flow Control Pause frame when its Receive FIFO is almost full. The value programmed into this field determines the number of bytes still available in the Receive FIFO when the Pause frame is generated. **Bits 2:0** — **Free Bytes in Receive FIFO** 000 — 0.50 Kbyte — Fast system (recommended default) 001 — 1.00 Kbyte 010 — 1.25 Kbyte 011 — 1.50 Kbyte 100 — 1.75 Kbyte 101 — 2.00 Kbyte 110 — 2.25 Kbyte 111 — 2.50 Kbyte — Slow system

6.2.10 Power Management Driver (PMDR) Register

Offset Address: 1Bh — Attribute: R/WC
Default Value: 00h — Size: 8 bits

The 82801E C-ICH's internal LAN Controller provides an indication in the PMDR that a wake-up event has occurred.

Bit	Description
7	**Link Status Change Indication—**R/WC. 1 = The link status change bit is set following a change in link status. 0 = Software clears this bit by writing a 1 to the bit location.
6	**Magic Packet—**R/WC. 1 = This bit is set when a Magic Packet* is received regardless of the Magic Packet wake-up disable bit in the configuration command and the PME Enable bit in the Power Management Control/ Status Register. 0 = Software clears this bit by writing a 1 to the bit location.
5	**Interesting Packet—**R/WC. 1 = This bit is set when an "interesting" packet is received. Interesting packets are defined by the LAN Controller packet filters. 0 = Software clears this bit by writing a 1 to the bit location.
4:0	Reserved. Should be set to 0.

6.2.11 General Control Register

Offset Address: 1Ch Attribute: R/W
Default Value: 00h Size: 8 bits

Bit	Description
7:4	Reserved. These bits should be set to 0000b.
3	**LAN Connect Software Reset—**R/W. 1 = Software can set this bit to force a reset condition on the LAN Connect interface. 0 = Cleared by software to begin normal LAN Connect operating mode. Software must not attempt to access the LAN Connect interface for at least 1 ms after clearing this bit.
2:0	Reserved. These bits should be set to 0.

6.2.12 General Status Register

Offset Address: 1Dh Attribute: RO
Default Value: N/A Size: 8 bits

Bit	Description
7:3	Reserved.
2	**Duplex Mode.** This bit indicates the wire duplex mode. 1 = Full duplex 0 = Half duplex
1	**Speed.** This bit indicates the wire speed: 1 = 100 Mbps 0 = 10 Mbps
0	**Link Status Indication.** This bit indicates the status of the link: 1 = Valid 0 = Invalid

6.2.13 Statistical Counters

The 82801E C-ICH's integrated LAN Controller provides information for network management statistics by providing on-chip statistical counters that count a variety of events associated with both transmit and receive. The counters are updated by the LAN Controller when it completes the processing of a frame (i.e., when it has completed transmitting a frame on the link or when it has completed receiving a frame). The Statistical Counters are reported to the software on demand by issuing the Dump Statistical Counters command or Dump and Reset Statistical Counters command in the SCB Command Unit Command (CUC) field.

Table 90. Statistical Counters (Sheet 1 of 2)

ID	Counter	Description
0	Transmit Good Frames	This counter contains the number of frames that were transmitted properly on the link. It is updated only after the actual transmission on the link is completed, not when the frame was read from memory as is done for the Transmit Command Block status.
4	Transmit Maximum Collisions (MAXCOL) Errors	This counter contains the number of frames that were not transmitted because they encountered the configured maximum number of collisions.
8	Transmit Late Collisions (LATECOL) Errors	This counter contains the number of frames that were not transmitted since they encountered a collision later than the configured slot time.
12	Transmit Underrun Errors	A transmit underrun occurs because the system bus cannot keep up with the transmission. This counter contains the number of frames that were either not transmitted or retransmitted due to a transmit DMA underrun. If the LAN Controller is configured to retransmit on underrun, this counter may be updated multiple times for a single frame.
16	Transmit Lost Carrier Sense (CRS)	This counter contains the number of frames that were transmitted by the LAN Controller despite the fact that it detected the deassertion of CRS during the transmission.
20	Transmit Deferred	This counter contains the number of frames that were deferred before transmission due to activity on the link.
24	Transmit Single Collisions	This counter contains the number of transmitted frames that encountered one collision.
28	Transmit Multiple Collisions	This counter contains the number of transmitted frames that encountered more than one collision.
32	Transmit Total Collisions	This counter contains the total number of collisions that were encountered while attempting to transmit. This count includes late collisions and frames that encountered MAXCOL.
36	Receive Good Frames	This counter contains the number of frames that were received properly from the link. It is updated only after the actual reception from the link is completed and all the data bytes are stored in memory.
40	Receive CRC Errors	This counter contains the number of aligned frames discarded because of a CRC error. This counter is updated, if needed, regardless of the Receive Unit state. The Receive CRC Errors counter is mutually exclusive of the Receive Alignment Errors and Receive Short Frame Errors counters.
44	Receive Alignment Errors	This counter contains the number of frames that are both misaligned (for example, CRS deasserts on a non-octal boundary) and contain a CRC error. The counter is updated, if needed, regardless of the Receive Unit state. The Receive Alignment Errors counter is mutually exclusive of the Receive CRC Errors and Receive Short Frame Errors counters.
48	Receive Resource Errors	This counter contains the number of good frames discarded due to unavailability of resources. Frames intended for a host whose Receive Unit is in the No Resources state fall into this category. If the LAN Controller is configured to Save Bad Frames and the status of the received frame indicates that it is a bad frame, the Receive Resource Errors counter is not updated.
52	Receive Overrun Errors	This counter contains the number of frames known to be lost because the local system bus was not available. If the traffic problem persists for more than one frame, the frames that follow the first are also lost; however, because there is no lost frame indicator, they are not counted.
56	Receive Collision Detect (CDT)	This counter contains the number of frames that encountered collisions during frame reception.

Table 90. Statistical Counters (Sheet 2 of 2)

ID	Counter	Description
60	Receive Short Frame Errors	This counter contains the number of received frames that are shorter than the minimum frame length. The Receive Short Frame Errors counter is mutually exclusive to the Receive Alignment Errors and Receive CRC Errors counters. A short frame will always increment only the Receive Short Frame Errors counter.
64	Flow Control Transmit Pause	This counter contains the number of Flow Control frames transmitted by the LAN Controller. This count includes both the Xoff frames transmitted and Xon (PAUSE(0)) frames transmitted.
68	Flow Control Receive Pause	This counter contains the number of Flow Control frames received by the LAN Controller. This count includes both the Xoff frames received and Xon (PAUSE(0)) frames received.
72	Flow Control Receive Unsupported	This counter contains the number of MAC Control frames received by the LAN Controller that are not Flow Control Pause frames. These frames are valid MAC control frames that have the predefined MAC control Type value and a valid address but has an unsupported opcode.
76	Receive TCO Frames	This counter contains the number of TCO packets received by the LAN Controller.
78	Transmit TCO Frames	This counter contains the number of TCO packets transmitted.

The Statistical Counters are initially set to zero by the 82801E C-ICH's integrated LAN Controller after reset. They cannot be preset to anything other than zero. The LAN Controller increments the counters by internally reading them, incrementing them and writing them back. This process is invisible to the processor and PCI bus. In addition, the counters adhere to the following rules:

- The counters are wrap-around counters. After reaching FFFFFFFFh the counters wrap around to 0.
- The LAN Controller updates the required counters for each frame. It is possible for more than one counter to be updated as multiple errors can occur in a single frame.
- The counters are 32 bits wide and their behavior is fully compatible with the IEEE 802.1 standard. The LAN Controller supports all mandatory and recommend statistics functions through the status of the receive header and directly through these Statistical Counters.

The processor can access the counters by issuing a Dump Statistical Counters SCB command. This provides a "snapshot", in main memory, of the internal LAN Controller statistical counters. The LAN Controller supports 21 counters. The dump could consist of the either 16, 19, or all 21 counters, depending on the status of the Extended Statistics Counters and TCO Statistics configuration bits in the Configuration command.

Hub Interface to PCI Bridge Registers (D30:F0) 7

The hub interface to PCI Bridge resides in PCI Device 30, Function 0 on bus #0. This portion of the 82801E C-ICH implements the buffering and control logic between PCI and the hub interface. The arbitration for the PCI bus is handled by this PCI device. The PCI decoder in this device must decode the ranges for the hub interface. All register contents are lost when core well power is removed.

7.1 PCI Configuration Registers (D30:F0)

Note: Registers that are not shown should be treated as Reserved (See "PCI Configuration Map" on page 162 for details).

Table 91. PCI Configuration Map (HUB-PCI—D30:F0) (Sheet 1 of 2)

Offset	Mnemonic	Register Name/Function	Default	Type
00h–01h	VID	Vendor ID	8086h	RO
02h–03h	DID	Device ID	245Eh	RO
04h–05h	CMD	PCI Device Command Register	0001h	R/W
06h–07h	PD_STS	PCI Device Status Register	0080h	R/W
08h	REVID	Revision ID	See Note	RO
0Ah	SCC	Sub Class Code	04h	RO
0Bh	BCC	Base Class Code	06h	RO
0Dh	PMLT	Primary Master Latency Timer	00h	RO
0Eh	HEADTYP	Header Type	01h	RO
18h	PBUS_NUM	Primary Bus Number	00h	RO
19h	SBUS_NUM	Secondary Bus Number	00h	R/W
1Ah	SUB_BUS_NUM	Subordinate Bus Number	00h	R/W
1Bh	SMLT	Secondary Master Latency Timer	00h	R/W
1Ch	IOBASE	IO Base Register	F0h	R/W
1Dh	IOLIM	IO Limit Register	00h	R/W
1Eh–1Fh	SECSTS	Secondary Status Register	0280h	R/W
20h–21h	MEMBASE	Memory Base	FFF0h	R/W
22h–23h	MEMLIM	Memory Limit	0000h	R/W
24h–25h	PREF_MEM_BASE	Prefetchable Memory Base	0000h	RO
26h–27h	PREF_MEM_MLT	Prefetchable Memory Limit	0000h	RO
30h–31h	IOBASE_HI	I/O Base Upper 16 Bits	0000h	RO

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

Table 91. PCI Configuration Map (HUB-PCI—D30:F0) (Sheet 2 of 2)

Offset	Mnemonic	Register Name/Function	Default	Type
32h–33h	IOLIMIT_HI	I/O Limit Upper 16 Bits	0000h	RO
3Ch	INT_LINE	Interrupt Line	00h	RO
3Eh–3Fh	BRIDGE_CNT	Bridge Control	0000h	R/W
40h	BRIDGE_CNT2	Bridge Control 2	00h	R/W
50h–51h	CNF	82801E C-ICH Configuration Register	0000h	R/W
70h	MTT	Multi-Transaction Timer	20h	R/W
82h	PCI_MAST_STS	PCI Master Status	00h	R/W
90h	ERR_CMD	Error Command Register	00h	R/W
92h	ERR_STS	Error Status Register	00h	R/W

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

7.1.1 VID—Vendor ID Register (HUB-PCI—D30:F0)

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

Bit	Description
15:0	**Vendor Identification Number—**RO. This is a 16-bit value assigned to Intel. Intel VID = 8086h.

7.1.2 DID—Device ID Register (HUB-PCI—D30:F0)

Offset Address: 02–03h
Default Value: 245Eh
Attribute: RO
Size: 16 bits

Bit	Description
15:0	**Device Identification Number—**RO. This is a 16 bit value assigned to the 82801E C-ICH hub interface to PCI bridge (i.e., Device 30). When this register is read, the 82801E C-ICH returns 245Eh.

7.1.3 CMD—Command Register (HUB-PCI—D30:F0)

Offset Address: 04–05h Attribute: R/W
Default Value: 0001h Size: 16 bits

Bit	Description
15:10	Reserved.
9	**Fast Back to Back Enable (FBE)**—RO. Hardwired to 0. The 82801E C-ICH does not support this capability.
8	**SERR# Enable (SERR_EN)—**R/W. 1 = Enable the 82801E C-ICH to generate an NMI (or SMI# if NMI routed to SMI#) when the D30:F0 SSE bit (offset 06h, bit 14) is set. 0 = Disable.
7	**Wait Cycle Control**—RO. Hardwired to 0
6	**Parity Error Response—**R/W. 1 = The 82801E C-ICH is allowed to report parity errors detected on the hub interface. 0 = The 82801E C-ICH will ignore parity errors on the hub interface. **NOTE:** The 82801E C-ICH does not generate a valid Hub Interface Parity message when Parity Error Response is disabled for the Hub/PCI Bridge function. The system may hang if Hub Interface Parity Error Response is enabled in the MCH when it is disabled in the 82801E C-ICH. This issue only affects platforms in which the MCH supports Hub Interface Parity checking. Software must not disable Hub Interface Parity Error Response in the ICH while it is enabled in the MCH. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.
5	**VGA Palette Snoop**—RO. Hardwired to 0.
4	**Postable Memory Write Enable (PMWE)**—RO. Hardwired to 0.
3	**Special Cycle Enable (SCE)**—RO. Hardwired to 0 by P2P Bridge specification.
2	**Bus Master Enable (BME)—**R/W. Allows the Hub interface-to-PCI bridge to accept cycles from PCI to run on the hub interface. **NOTE:** This bit does not affect the CF8h and CFCh I/O accesses. 0 = Disable
1	**Memory Space Enable (MSE)—**R/W. The 82801E C-ICH provides this bit as read/writable for software only. However, the 82801E C-ICH ignores the programming of this bit, and runs hub interface memory cycles to PCI.
0	**I/O Space Enable (IOE)—**R/W. The 82801E C-ICH provides this bit as read/writable for software only. However, the 82801E C-ICH ignores the programming of this bit and runs hub interface I/O cycles to PCI that are not intended for USB or IDE.

7.1.4 PD_STS—Primary Device Status Register (HUB-PCI—D30:F0)

Offset Address:	06–07h	Attribute:	R/WC
Default Value:	0080h	Size:	16 bits

For the writable bits in this register, writing a 1 will clear the bit. Writing a 0 to the bit will have no effect.

Bit	Description
15	**Detected Parity Error (DPE)—**R/WC. 1 = Indicates that the 82801E C-ICH detected a parity error on the hub interface. This bit gets set even if the Parity Error Response bit (offset 04, bit 6) is not set. 0 = Software clears this bit by writing a 1 to the bit location. **NOTE:** The 82801E C-ICH's Detected Parity Error (DPE) bit in SECSTS register (D30:F0, offset 1Eh: bit-15) and PD_STS register (D31:F0, offset 06h: bit-15) will get set when PERR# is asserted by external PCI devices. The DPE bit of SECSTS and PD_STS may erroneously get set. BIOS needs to clear DPE of SECSTS and PD_STS when those bits are set. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.
14	**Received System Error (SSE)—**R/WC. 1 = An address, or command parity error, or special cycles data parity error has been detected on the PCI bus, and the Parity Error Response bit (D30:F0, Offset 04h, bit 6) is set. If this bit is set because of parity error and the D30:F0 SERR_EN bit (Offset 04h, bit 8) is also set, the 82801E C-ICH will generate an NMI (or SMI# if NMI routed to SMI#) 0 = Software clears this bit by writing a 1 to the bit location.
13	**Received Master Abort (RMA)**—R/WC. 1 = 82801E C-ICH received a master abort from the hub interface device. 0 = Software clears this bit by writing a 1 to the bit location.
12	**Received Target Abort (RTA)—**R/WC. 1 = 82801E C-ICH received a target abort from the hub interface device. The TCO logic can cause an SMI#, NMI, or interrupt based on this bit getting set. 0 = Software clears this bit by writing a 1 to the bit location.
11	**Signaled Target Abort (STA)**—R/WC. 1 = 82801E C-ICH signals a target abort condition on the hub interface. 0 = Software clears this bit by writing a 1 to the bit location.
10:9	**DEVSEL# Timing Status**—RO. 00h = Fast timing. This register applies to the hub interface; therefore, this field does not matter.
8	**Data Parity Error Detected (DPD)—**R/WC. Since this register applies to the hub interface, the 82801E C-ICH must interpret this bit differently than how it is defined in the PCI specification. 1 = 82801E C-ICH detects a parity error on the hub interface and the Parity Error Response bit in the Command Register (offset 04h, bit 6) is set. 0 = Software clears this bit by writing a 1 to the bit location. **NOTE:** 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 two 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 fully functional with regards to this erratum. 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. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.
7	**Fast Back to Back**—RO. Hardwired to 1.

Bit	Description
6	**User Definable Features (UDF)**—RO. Hardwired to 0.
5	**66 MHz Capable**—RO. Hardwired to 0.
4:0	Reserved.

7.1.5 REVID—Revision ID Register (HUB-PCI—D30:F0)

Offset Address: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

Bit	Description
7:0	**Revision Identification Number**—RO. 8-bit value that indicates the revision number for the 82801E C-ICH hub interface to PCI bridge. Refer to the Specification Update for the value of the Revision ID Register.

7.1.6 SCC—Sub-Class Code Register (HUB-PCI—D30:F0)

Offset Address: 0Ah Attribute: RO
Default Value: 04h Size: 8 bits

Bit	Description
7:0	**Sub Class Code**—RO. This 8-bit value indicates the category of bridge for the 82801E C-ICH hub interface to PCI bridge. The code is 04h to indicate a PCI-to-PCI bridge.

7.1.7 BCC—Base-Class Code Register (HUB-PCI—D30:F0)

Offset Address: 0Bh Attribute: RO
Default Value: 06h Size: 8 bits

Bit	Description
7:0	**Base Class Code**—RO. This 8-bit value indicates the type of device for the 82801E C-ICH hub interface to PCI bridge. The code is 06h to indicate a bridge device.

7.1.8 PMLT—Primary Master Latency Timer Register (HUB-PCI—D30:F0)

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

This register does not apply to hub interface.

Bit	Description
7:3	**Master Latency Count**. Not implemented.
2:0	Reserved.

7.1.9 HEADTYP—Header Type Register (HUB-PCI—D30:F0)

Offset Address: 0Eh Attribute: RO
Default Value: 01h Size: 8 bits

Bit	Description
7	**Multi-function Device**—RO. This bit is 0 to indicate a single function device.
6:0	**Header Type**—RO. 7-bit field identifies the header layout of the configuration space, which is a PCI-to-PCI bridge in this case.

7.1.10 PBUS_NUM—Primary Bus Number Register (HUB-PCI—D30:F0)

Offset Address: 18h Attribute: RO
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**Primary Bus Number**—RO. This field indicates the bus number of the hub interface and is hardwired to 00h.

7.1.11 SBUS_NUM—Secondary Bus Number Register (HUB-PCI—D30:F0)

Offset Address: 19h Attribute: R/W
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**Secondary Bus Number**—R/W. This field indicates the bus number of PCI. Note that when this number is equal to the primary bus number (i.e., bus #0), the 82801E C-ICH will run hub interface configuration cycles to this bus number as Type 1 configuration cycles on PCI.

7.1.12 SUB_BUS_NUM—Subordinate Bus Number Register (HUB-PCI—D30:F0)

Offset Address: 1A Attribute: R/W
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**Subordinate Bus Number**—R/W. This field specifies the highest PCI bus number below the hub interface to PCI bridge. If a Type 1 configuration cycle from the hub interface does not fall in the Secondary-to-Subordinate Bus ranges of Device 30, the 82801E C-ICH indicates a master abort back to the hub interface.

7.1.13 SMLT—Secondary Master Latency Timer Register (HUB-PCI—D30:F0)

Offset Address:	1Bh	Attribute:	R/W
Default Value:	00h	Size:	8 bits

This Master Latency Timer (MLT) controls the amount of time that the 82801E C-ICH continues to burst data as a master on the PCI bus. When the 82801E C-ICH starts the cycle after being granted the bus, the counter is loaded and starts counting down from the assertion of FRAME#. If the internal grant to this device is removed, then the expiration of the MLT counter results in the deassertion of FRAME#. If the internal grant has not been removed, the 82801E C-ICH can continue to own the bus.

Bit	Description
7:3	**Master Latency Count**—R/W. This 5-bit value indicates the number of PCI clocks, in 8-clock increments, that the 82801E C-ICH remains as master of the bus.
2:0	Reserved.

7.1.14 IOBASE—I/O Base Register (HUB-PCI—D30:F0)

Offset Address:	1Ch	Attribute:	R/W
Default Value:	F0h	Size:	8 bits

Bit	Description
7:4	**I/O Address Base bits [15:12]—**R/W. I/O Base bits corresponding to address lines 15:12 for 4 Kbyte alignment. Bits 11:0 are assumed to be padded to 000h.
3:0	**I/O Addressing Capability—**RO. This is hardwired to 0h, indicating that the hub interface to PCI bridge does not support 32-bit I/O addressing. This means that the I/O Base Register and I/O Limit Upper Address registers must be read only.

7.1.15 IOLIM—I/O Limit Register (HUB-PCI—D30:F0)

Offset Address:	1Dh	Attribute:	R/W
Default Value:	00h	Size:	8 bits

Bit	Description
7:4	**I/O Address Limit bits [15:12]—**R/W. I/O Base bits corresponding to address lines 15:12 for 4 Kbyte alignment. Bits 11:0 are assumed to be padded to FFFh.
3:0	**I/O Addressing Capability—**RO. This is hardwired to 0h, indicating that the hub interface-to-PCI bridge does not support 32-bit I/O addressing. This means that the I/O Base Register and I/O Limit Upper Address registers must be read only.

7.1.16 SECSTS—Secondary Status Register (HUB-PCI—D30:F0)

Offset Address:	1E–1Fh	Attribute:	R/W
Default Value:	0280h	Size:	16 bits

For the writable bits in this register, writing a 1 will clear the bit. Writing a 0 to the bit will have no effect.

Bit	Description
15	**Detected Parity Error (DPE)**—R/WC. 1 = 82801E C-ICH detected a parity error on the PCI bus. 0 = Software clears this bit by writing a 1 to the bit position. **NOTE:** The 82801E C-ICH's Detected Parity Error (DPE) bit in SECSTS register (D30:F0, offset 1Eh: bit-15) and PD_STS register (D31:F0, offset 06h: bit-15) will get set when PERR# is asserted by external PCI devices. The DPE bit of SECSTS and PD_STS may erroneously get set. BIOS needs to clear DPE of SECSTS and PD_STS when those bits are set. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.
14	**Received System Error (SSE)—**R/WC. 1 = SERR# assertion is received on PCI. 0 = Software clears this bit by writing a 1 to the bit position.
13	**Received Master Abort (RMA)**—R/WC. 1 = Hub interface to PCI cycle is master-aborted on PCI. 0 = Software clears this bit by writing a 1 to the bit position.
12	**Received Target Abort (RTA)**—R/WC. 1 = Hub interface to PCI cycle is target-aborted on PCI. For "completion required" cycles from the hub interface, this event should also set the Signaled Target Abort in the Primary Status Register in this device and the 82801E C-ICH must send the "target abort" status back to the hub interface. 0 = Software clears this bit by writing a 1 to the bit position.
11	**Signaled Target Abort (STA)**—RO. The 82801E C-ICH does not generate target aborts.
10:9	**DEVSEL# Timing Status**—RO. 01h = Medium timing.
8	**Data Parity Error Detected (DPD)**—R/WC. 1 = The 82801E C-ICH sets this bit when all of the following three conditions are met: - The Parity Error Response Enable bit in the Bridge Control Register (bit 0, offset 3Eh) is set - USB or IDE is a Master - PERR# asserts during a write cycle ***OR*** a parity error is detected internally during a read cycle 0 = Software clears this bit by writing a 1 to the bit position. **NOTE:** When enabled, a Parity Error may not be signaled or detected, via either D30:F0;1Eh bit-8 or D31:F0;06h bit-8 (SECSTS:[DPD] or PCISTS:[DPED]), if two 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 fully functional with regards to this issue. 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. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.
7	**Fast Back to Back**—RO. Hardwired to 1 to indicate that the PCI to hub interface target logic is capable of receiving fast back-to-back cycles.

Bit	Description
6	**User Definable Features (UDF)**—RO. Hardwired to 0.
5	**66 MHz Capable**—RO. Hardwired to 0.
4:0	Reserved.

7.1.17 MEMBASE—Memory Base Register (HUB-PCI—D30:F0)

Offset Address: 20–21h Attribute: R/W
Default Value: FFF0h Size: 16 bits

This register defines the base of the hub interface to PCI non-prefetchable memory range. Since the 82801E C-ICH forwards all hub interface memory accesses to PCI, the 82801E C-ICH only uses this information to determine when not to accept cycles as a target.

This register must be initialized by the configuration software. For the purpose of address decode, address bits A[19:0] are assumed to be 0. Thus, the bottom of the defined memory address range will be aligned to a 1 Mbyte boundary.

Bit	Description
15:4	**Memory Address Base**—R/W. Defines the base of the memory range for PCI. These 12 bits correspond to address bits 31:20.
3:0	Reserved.

7.1.18 MEMLIM—Memory Limit Register (HUB-PCI—D30:F0)

Offset Address: 22–23h Attribute: R/W
Default Value: 0000h Size: 16 bits

This register defines the upper limit of the hub interface to PCI non-prefetchable memory range. Since the 82801E C-ICH will forward all hub interface memory accesses to PCI, the 82801E C-ICH will only use this information to determine when not to accept cycles as a target.

This register must be initialized by the configuration software. For the purpose of address decode, address bits A[19:0] are assumed to be FFFFFh. Thus, the top of the defined memory address range will be aligned to a 1 Mbyte boundary.

Bit	Description
15:4	**Memory Address Limit**—R/W. Defines the top of the memory range for PCI. These 12 bits correspond to address bits 31:20.
3:0	Reserved.

7.1.19 PREF_MEM_BASE—Prefetchable Memory Base Register (HUB-PCI—D30:F0)

Offset Address: 24h–25h Attribute: R/W
Default Value: 0000h Size: 16-bit

Bit	Description
15:4	**Prefetchable Memory Address Base—**R/W. Defines the base address of the prefetchable memory address range for PCI. These 12 bits correspond to address bits 31:20.
3:0	Reserved. RO.

7.1.20 PREF_MEM_MLT—Prefetchable Memory Limit Register (HUB-PCI—D30:F0)

Offset Address: 26h–27h Attribute: R/W
Default Value: 0000h Size: 16-bit

Bit	Description
15:4	**Prefetchable Memory Address Limit—**RW. Defines the limit address of the prefetchable memory address range for PCI. These 12 bits correspond to address bits 31:20.
3:0	Reserved. RO

7.1.21 IOBASE_HI—I/O Base Upper 16 Bits Register (HUB-PCI—D30:F0)

Offset Address: 30–31h Attribute: RO
Default Value: 0000h Size: 16 bits

Bit	Description
15:0	**I/O Address Base Upper 16 bits [31:16]**—RO. Not supported; hardwired to 0.

7.1.22 IOLIM_HI—I/O Limit Upper 16 Bits Register (HUB-PCI—D30:F0)

Offset Address: 32–33h Attribute: RO
Default Value: 0000h Size: 16 bits

Bit	Description
15:0	**I/O Address Limit Upper 16 bits [31:16]**—RO. Not supported; hardwired to 0.

7.1.23 INT_LINE—Interrupt Line Register (HUB-PCI—D30:F0)

Offset Address: 3Ch
Default Value: 00h
Attribute: RO
Size: 8 bits

Bit	Description
7:0	**Interrupt Line Routing**—RO. Hardwired to 00h. The bridge does not generate interrupts, and interrupts from downstream devices are routed around the bridge.

7.1.24 BRIDGE_CNT—Bridge Control Register (HUB-PCI—D30:F0)

Offset Address: 3E–3Fh
Default Value: 0000h
Attribute: R/W
Size: 16 bits

Bit	Description
15:8	Reserved.
7	**Fast Back to Back Enable**—RO. Hardwired to 0. The PCI logic will not generate fast back-to-back cycles on the PCI bus.
6	**Secondary Bus Reset**—RO. hardwired to 0. The 82801E C-ICH does not follow the point to point (P2P) bridge reset scheme; Software-controlled resets are implemented in the PCI-LPC device.
5	**Master Abort Mode—**R/W. The 82801E C-ICH ignores this bit. However, this bit is read/write for software compatibility. The 82801E C-ICH must handle master aborts as if this bit is reset to 0.
4	Reserved.
3	**VGA Enable**—R/W. 1 = Enable. Indicates that the VGA device is on PCI. Therefore, the PCI to hub interface decoder will not accept memory cycles in the range A0000h–BFFFFh. Note that the 82801E C-ICH will never take I/O cycles in the VGA range from PCI. 0 = No VGA device on PCI.
2	**ISA Enable—**R/W. The 82801E C-ICH ignores this bit. However, this bit is read/write for software compatibility. Since the 82801E C-ICH forwards all I/O cycles that are not in the USB, or IDE ranges to PCI, this bit would have no effect.
1	**SERR# Enable—**R/W. 1 = Enable. If this bit is set AND bit 8 in CMD register (D30:F0 Offset 04h) is also set, the 82801E C-ICH sets the SSE bit in PD_STS register (D30:F0, offset 06h, bit 14) AND also generate an NMI (or SMI# if NMI routed to SMI) when the SERR# signal is asserted. 0 = Disable
0	**Parity Error Response Enable—**R/W. 1 = Enable the hub interface to PCI bridge for parity error detection and reporting on the PCI bus. 0 = Disable **NOTE:** If the 82801E C-ICH's Parity Error Response Enable (PER) bit in Bridge_CNT register (D30:F0, offset 3Eh: bit-0) is disabled (default), it will block PERR# from being asserted when a data parity error is detected on the PCI bus during LPC or legacy DMA master read cycles, or when the 82801E C-ICH is the target for write cycles to Device 31 Functions 0 and 3. This bit should only block PERR# from being asserted when a PCI data parity error is detected during PCI-to-memory writes or CPU-to-PCI read cycles. PERR# will not be asserted when PCI Parity Error detected during LPC or legacy DMA master read cycles, or when the 82801E C-ICH is the target for write cycles to Device 31 Functions 0 and 3. BIOS needs to set PER of Bridge_CNT when the parity error detection is supported on LPC or legacy DMA. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.

7.1.25 BRIDGE_CNT2—Bridge Control Register 2 (HUB-PCI—D30:F0)

Offset Address: 40h Attribute: R/W
Default Value: 00h Size 8 bits

Bit	Description
7:1	Reserved.
0	**PCI_DAC_EN—**R/W. Allows 82801E C-ICH to recognize external PCI masters performing DAC on PCI. 0 = Disable. 1 = Enable.

7.1.26 CNF—82801E C-ICH Configuration Register (HUB-PCI—D30:F0)

Offset Address: 50–51h Attribute: R/W
Default Value: 0000h Size: 16 bits

Bit	Description
15:10	Reserved.
9	**HP_PCI_EN**—R/W. High Priority PCI Enable. 1 = Enables a mode where the REQ[0]#/GNT[0]# signal pair has a higher arbitration priority. 0 = All PCI REQ#/GNT pairs have the same arbitration priority.
8	**Hole Enable (15 Mbyte–16 Mbyte)—**R/W. 1 = Enables the 15 Mbyte to 16 Mbyte hole in the DRAM. 0 = Disable
7:3	Reserved.
2	**Discard Timer Mode.** This bit shortens all of the Delayed Transaction discard timers to 128 PCI clocks. It controls how long the 82801E C-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 µs) (Default) 1 = 128 PCI clocks (4 µs)
1	**32-Clock Retry Enable—**R/W. System BIOS must set this bit for PCI compliance. 1 = When a PCI device is running a locked memory read cycle, while all other bus masters are waiting to run locked cycles, concurrent with a LPC DMA transfer, this bit, when set allows the 82801E C-ICH to retry the locked memory read cycle. 0 = If this bit is not set, under the same circumstance, the bus will not be released since all other masters see the lock in use.
0	Reserved.

7.1.27 MTT—Multi-Transaction Timer Register (HUB-PCI—D30:F0)

Offset Address:	70h	Attribute:	R/W
Default Value:	20h	Size:	8 bits

MTT is an 8-bit register that controls the amount of time that the 82801E C-ICH's arbiter allows a PCI initiator to perform multiple back-to-back transactions on the PCI bus. The 82801E C-ICH's MTT mechanism is used to guarantee a fair share of the Primary PCI bandwidth to an initiator that performs multiple back-to-back transactions to fragmented memory ranges (and as a consequence it can not use long burst transfers).

The number of clocks programmed in the MTT represents the guaranteed time slice (measured in PCI clocks) allotted to the current agent, after which the arbiter grants another agent that is requesting the bus. The MTT value must be programmed with eight clock granularity in the same manner as MLT. For example, if the MTT is programmed to 18h, the selected value corresponds to the time period of 24 PCI clocks.The default value of MTT is 20h (32 PCI clocks).

Note: Programming the MTT to a value of 00h disables this function, which could cause starvation issues for some PCI master devices. Programming of the MTT to anything less than 16 clocks will not allow the Grant-to-FRAME# latency to be 16 clocks. The MTT timer will time-out before the Grant-to-FRAME# trigger causing a re-arbitration.

Bit	Description
7:3	**Multi-Transaction Timer Count Value—**R/W. This field specifies the amount of time that grant remains asserted to a master continuously asserting its request for multiple transfers. This field specifies the count in an 8-clock (PCI clock) granularity.
2:0	Reserved.

7.1.28 PCI_MAST_STS—PCI Master Status Register (HUB-PCI—D30:F0)

Offset Address:	82h	Attribute:	R/WC
Default Value:	00h	Size:	8 bits

Bit	Description
7	**Internal PCI Master Request Status (INT_MREQ_STS)—**R/WC. 1 = The 82801E C-ICH's internal DMA controller or LPC has requested use of the PCI bus. 0 = Software clears this bit by writing a 1 to the bit position.
6	**Internal LAN Master Request Status (LAN_MREQ_STS)—**R/WC. 1 = The 82801E C-ICH's internal LAN controller has requested use of the PCI bus. 0 = Software clears this bit by writing a 1 to the bit position.
5:0	**PCI Master Request Status (PCI_MREQ_STS)—**R/WC. Allows software to see if a particular bus master has requested use of the PCI bus. For example, bit 0 will be set if 82801E C-ICH has detected REQ[0]# asserted and bit 5 will be set if 82801E C-ICH detected REQ[5]# asserted. 1 = The associated PCI master has requested use of the PCI bus. 0 = Software clears these bits by writing a 1 to the bit position.

7.1.29 ERR_CMD—Error Command Register (HUB-PCI—D30:F0)

Offset Address:	90h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

This register configures the 82801E C-ICH's Device 30 responses to various system errors. The actual assertion of the internal SERR# (routed to cause NMI# or SMI#) is enabled via the PCI Command register.

Bit	Description
7:3	Reserved.
2	**SERR# enable on receiving target abort (SERR_RTA_EN)—**R/W. 1 = Enable. When SERR_EN is set, the 82801E C-ICH will report SERR# when SERR_RTA is set. 0 = Disable
1	**SERR# enable on Delayed Transaction Time-out (SERR_DTT_EN)—**R/W. 1 = Enable. When SERR_EN is set, the 82801E C-ICH will report SERR# when SERR_DTT is set. 0 = Disable.
0	Reserved.

7.1.30 ERR_STS—Error Status Register (HUB-PCI—D30:F0)

Offset Address:	92h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

This register records the cause of system errors in Device 30. The actual assertion of SERR# is enabled via the PCI Command register.

Bit	Description
7:3	Reserved.
2	**SERR# Due to Received Target Abort (SERR_RTA)—**R/W. 1 = The 82801E C-ICH sets this bit when the 82801E C-ICH receives a target abort. If SERR_EN, the 82801E C-ICH will also generate an SERR# when SERR_RTA is set. 0 = This bit is cleared by writing a 1.
1	**SERR# Due to Delayed Transaction Time-out (SERR_DTT)—**R/W. 1 = When a PCI master does not return for the data within 1 ms of the cycle's completion, the 82801E C-ICH clears the delayed transaction, and sets this bit. If both SERR_DTT_EN and SERR_EN are set, then 82801E C-ICH will also generate an SERR# when SERR_DTT is set. 0 = This bit is cleared by writing a 1.
0	Reserved.

LPC Interface Bridge Registers (D31:F0) 8

The LPC Bridge function of the 82801E C-ICH resides in PCI Device 31:Function 0. This function contains many other functional units (e.g., DMA and Interrupt Controllers, Timers, Power Management, System Management., GPIO, RTC, and LPC Configuration Registers).

Registers and functions associated with other functional units (GPIO, USB, IDE, etc.) are described in their respective sections.

8.1 PCI Configuration Registers (D31:F0)

Note: Registers that are not shown should be treated as Reserved (See "PCI Configuration Map" on page 162 for details).

Table 92. PCI Configuration Map (LPC I/F—D31:F0) (Sheet 1 of 2)

Offset	Mnemonic	Register Name	Default	Type
00h–01h	VID	Vendor ID	8086h	RO
02h–03h	DID	Device ID	2450h	RO
04h–05h	PCICMD	PCI Command Register	000Fh	R/W
06h–07h	PCISTS	PCI Device Status Register	0280h	R/W
08h	RID	Revision ID	See Note	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	01h	RO
0Bh	BCC	Base Class Code	06h	RO
0Eh	HEADT	Header Type	80h	RO
40h–43h	PMBASE	ACPI Base Address Register	00000001h	R/W
44h	ACPI_CNTL	ACPI Control	00h	R/W
4Eh–4Fh	BIOS_CNTL	BIOS Control Register	0000h	R/W
54h	TCO_CNTL	TCO Control	00h	R/W
58h–5Bh	GPIO_BASE	GPIO Base Address Register	00000001h	R/W
5Ch	GPIO_CNTL	GPIO Control Register	00h	R/W
60h–63h	PIRQ[*n*]_ROUT	PIRQ[A–D] Routing Control	80808080h	R/W
64h	SIRQ_CNTL	Serial IRQ Control Register	10h	R/W
68h–6Bh	PIRQ[*n*]_ROUT	PIRQ[E–H] Routing Control	80808080h	R/W
88h	D31_ERR_CFG	Device 31 Error configuration Register	00h	R/W
8Ah	D31_ERR_STS	Device 31 Error Status Register	00h	R/W
90h–91h	PCI_DMA_C	PCI DMA Configuration Registers	0000h	R/W

Table 92. PCI Configuration Map (LPC I/F—D31:F0) (Sheet 2 of 2)

Offset	Mnemonic	Register Name	Default	Type
A0h–CFh	Power Management Registers. See “Power Management PCI Configuration Registers (D31:F0)” on page 256			
D0h–D3h	GEN_CNTL	General Control	00000000h	R/W
D4h–D7h	GEN_STS	General Status	00000F00h	R/W
D8h	RTC_CONF	Real Time Clock Configuration	00h	R/W
E0h	COM_DEC	LPC I/F COM Port Decode Ranges	00h	R/W
E1h	LPCFDD_DEC	LPC I/F FDD & LPT Decode Ranges	00h	R/W
E2h	SND_DEC	LPC I/F Sound Decode Ranges	00h	R/W
E3h	FWH_DEC_EN1	FWH Decode Enable 1	FFh	R/W
E4h–E5h	GEN1_DEC	LPC I/F General 1 Decode Range	0000h	R/W
E6h–E7h	LPC_EN	LPC I/F Enables	0000h	R/W
E8h–EBh	FWH_SEL1	FWH Select 1	00112233h	R/W
ECh–EDh	GEN2_DEC	LPC I/F General 2 Decode Range	0000h	R/W
EEh–EFh	FWH_SEL2	FWH Select 2	4567h	R/W
F0h	FWH_DEC_EN2	FWH Decode Enable 2	0Fh	R/W
F2h	FUNC_DIS	Function Disable Register	00h	R/W

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

8.1.1 VID—Vendor ID Register (LPC I/F—D31:F0)

Offset Address: 00–01h
Default Value: 8086h
Lockable: No
Attribute: RO
Size: 16-bit
Power Well: Core

Bit	Description
15:0	**Vendor ID Value.** This is a 16 bit value assigned to Intel. Intel VID = 8086h

8.1.2 DID—Device ID Register (LPC I/F—D31:F0)

Offset Address: 02–03h
Lockable: No
Default Value: 2450h
Attribute: RO
Size: 16-bit
Power Well: Core

Bit	Description
15:0	**Device ID Value.** This is a 16 bit value assigned to the 82801E C-ICH PCI to LPC Bridge (Device 31). When this register is read, the 82801E C-ICH returns 2450h.

8.1.3 PCICMD—PCI COMMAND Register (LPC I/F—D31:F0)

Offset Address:	04–05h	Attribute:	R/W
Default Value:	000Fh	Size:	16-bit
Lockable:	No	Power Well:	Core

Bit	Description
15:10	Reserved.
9	**Fast Back to Back Enable (FBE)**—RO. Hardwired to 0.
8	**SERR# Enable (SERR_EN)**—R/W. 1 **=** Enable. Allow SERR# to be generated. 0 = Disable.
7	**Wait Cycle Control (WCC)**—RO. Hardwired to 0.
6	**Parity Error Response (PER)—**R/W. 1 = The ICH will take normal action when a parity error is detected. 0 = No action is taken when detecting a parity error.
5	**VGA Palette Snoop (VPS)**—RO. Hardwired to 0
4	**Postable Memory Write Enable (PMWE)**—RO. Hardwired to 0
3	**Special Cycle Enable (SCE)**. Hardwired to 1.
2	**Bus Master Enable (BME)**—RO. Hardwired to 1 to indicate that bus mastering can not be disabled for function 0 (DMA/ISA Master).
1	**Memory Space Enable (MSE)**—RO. Hardwired to 1 to indicate that memory space can not be disabled for Function 0 (LPC I/F).
0	**I/O Space Enable (IOE)**—RO. Hardwired to 1 to indicate that the I/O space cannot be disabled for function 0 (LPC I/F).

8.1.4 PCISTS—PCI Device Status (LPC I/F—D31:F0)

Offset Address:	06–07h	Attribute:	R/WC
Default Value:	0280h	Size:16-bit	
Lockable:	No	Power Well:	Core

Bit	Description
15	**Detected Parity Error (DPE)**—R/W. 1 = PERR# signal goes active. Set even if the PER bit is 0. 0 = This bit is cleared by software writing a 1 to the bit position.
14	**Signaled System Error (SSE)**—R/W. 1 = Set by the 82801E C-ICH if the SERR_EN bit is set and the 82801E C-ICH generates an SERR# on function 0. The ERR_STS register can be read to determine the cause of the SERR#. The SERR# can be routed to cause SMI#, NMI, or interrupt. 0 = This bit is cleared by software writing a 1 to the bit position.
13	**Master Abort Status (RMA)**—R/W. 1 = 82801E C-ICH generated a master abort on PCI due to LPC I/F master or DMA cycles. 0 = This bit is cleared by software writing a 1 to the bit position.
12	**Received Target Abort (RTA)**—R/W. 1 = 82801E C-ICH received a target abort during LPC I/F master or DMA cycles to PCI. 0 = This bit is cleared by software writing a 1 to the bit position.
11	**Signaled Target Abort (STA)**—R/W. 1 = 82801E C-ICH generated a target abort condition on PCI cycles claimed by the 82801E C-ICH for internal registers or for going to LPC I/F. 0 = This bit is cleared by software writing a 1 to the bit position. **NOTE:** If there is a downstream I/O 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 82801E C-ICH can erroneously set the STA bit (bit-11) in D31:F0;PCISTS configuration space even though there is no Target Abort on the PCI bus. This has only been observed on DP systems. **NOTE:** 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. **NOTE:** There are two possible workarounds: • 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. • Disable Delayed Transaction (which may induce a performance penalty on PCI)
10:9	**DEVSEL# Timing Status (DEV_STS)**—RO. 01 = Medium Timing.
8	**Data Parity Error Detected (DPED)**—R/WC. 1 = Set when all three of the following conditions are true: - The 82801E C-ICH is the initiator of the cycle, - The 82801E C-ICH asserted PERR# (for reads) or observed PERR# (for writes), and - The PER bit is set. 0 = This bit is cleared by software writing a 1 to the bit position.
7	**Fast Back to Back (FB2B)**—RO. Always 1. Indicates 82801E C-ICH as a target can accept fast back-to-back transactions.
6	**User Definable Features (UDF)**. Hardwired to 0
5	**66 MHz Capable (66MHZ_CAP)**—RO. Hardwired to 0
4:0	Reserved.

8.1.5 REVID—Revision ID Register (LPC I/F—D31:F0)

Offset Address: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

Bit	Description
7:0	**Revision Identification Number.** 8-bit value that indicates the revision number for the LPC bridge. For the A-0 stepping, this value is 00h. Refer to the Specification Update for the value of the Revision ID Register

8.1.6 PI—Programming Interface (LPC I/F—D31:F0)

Offset Address: 09h Attribute: RO
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**Programming Interface Value**.

8.1.7 SCC—Sub-Class Code Register (LPC I/F—D31:F0)

Offset Address: 0Ah Attribute: RO
Default Value: 01h Size: 8 bits

Bit	Description
7:0	**Sub Class Code.** This 8-bit value indicates the category of bridge for the LPC PCI bridge.

8.1.8 BCC—Base-Class Code Register (LPC I/F—D31:F0)

Offset Address: 0Bh Attribute: RO
Default Value: 06h Size: 8 bits

Bit	Description
7:0	**Base Class Code.** This 8-bit value indicates the type of device for the LPC bridge. The code is 06h indicating a bridge device.

8.1.9 HEADTYP—Header Type Register (LPC I/F—D31:F0)

Offset Address: 0Eh Attribute: RO
Default Value: 80h Size: 8 bits

Bit	Description
7	**Multi-function Device—**RO. This bit is 1 to indicate a multi-function device.
6:0	**Header Type—**RO. This 8-bit field identifies the header layout of the configuration space.

8.1.10 PMBASE—ACPI Base Address (LPC I/F—D31:F0)

Offset Address:	40–43h	Attribute:	R/W
Default Value:	00000001h	Size:	32-bit
Lockable:	No	Usage:	ACPI, Legacy
		Power Well:	Core

Sets the base address for ACPI I/O registers, GPIO registers and TCO I/O registers. Can be mapped anywhere in the 64 Kbyte I/O space on 128-byte boundaries.

Bit	Description
31:16	Reserved.
15:7	**Base Address**—R/W. Provides 128 bytes of I/O space for ACPI, GPIO, and TCO logic. This is placed on a 128-byte boundary.
6:1	Reserved.
0	**Resource Indicator**—RO. Tied to 1 to indicate I/O space.

8.1.11 ACPI_CNTL—ACPI Control (LPC I/F—D31:F0)

Offset Address:	44h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Usage:	ACPI, Legacy
		Power Well:	Core

Bit	Description
7:5	Reserved.
4	**ACPI Enable (ACPI_EN)**—R/W. Not implemented, must be set to 0. 0 = Disable.
3	Reserved.
2:0	**SCI IRQ Select (SCI_IRQ_SEL)**—R/W. Specifies on which IRQ the SCI will internally appear.The SCI must be routed to IRQ[9:11], and that interrupt is not sharable with the SERIRQ stream, but is shareable with other PCI interrupts. 000 = IRQ9 001 = IRQ10 010 = IRQ11 011 = Reserved 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved

8.1.12 BIOS_CNTL (LPC I/F—D31:F0)

Offset Address:	4E–4Fh	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	No	Power Well:	Core

Bit	Description
15:2	Reserved.
1	**BIOS Lock Enable (BLE)**—R/W. 1 = Enables setting the BIOSWE bit to cause SMIs. 0 = Setting the BIOSWE will not cause SMIs. Once set, this bit can only be cleared by a PCIRST#.
0	**BIOS Write Enable (BIOSWE)**—R/W. 1 = Access to the BIOS space is enabled for both read and write cycles. When this bit is written from a 0 to a 1 and BIOS lock Enable (BLE) is also set, an SMI# is generated. This ensures that only SMM code can update BIOS. 0 = Only read cycles result in FWH interface cycles.

8.1.13 TCO_CNTL—TCO Control (LPC I/F—D31:F0)

Offset Address:	54h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:4	Reserved.
3	**TCO Interrupt Enable (TCO_INT_EN)**—R/W. This bit enables/disables the TCO interrupt. 1 = Enables TCO Interrupt, as selected by the TCO_INT_SEL field. 0 = Disables TCO interrupt.
2:0	**TCO Interrupt Select (TCO_INT_SEL)**—R/W. Specifies which IRQ the TCO internally appears. If not using the APIC, the TCO interrupt must be routed to IRQ[9:11], and that interrupt is not sharable with the SERIRQ stream, but is shareable with other PCI interrupts. Note that if the TCOSCI_EN bit is set (bit 6 in the GPE0_EN register), then the TCO interrupt will be sent to the same interrupt as the SCI, and the TCO_INT_SEL bits will have no meaning. 000 = IRQ9 001 = IRQ10 010 = IRQ11 011 = Reserved 100 = Reserved 101 = Reserved 110 = Reserved 111 = Reserved

8.1.14 GPIOBASE—GPIO Base Address (LPC I/F—D31:F0)

Offset Address:	58h–5Bh	Attribute:	R/W
Default Value:	00000001h	Size:	32-bit
Lockable:	No	Power Well:	Core

Bit	Description
31:16	Reserved.
15:6	**Base Address—**R/W. Provides the 64 bytes of I/O space for GPIO.
5:1	Reserved.
0	**Resource Indicator**—RO. Tied to 1 to indicate I/O space.

8.1.15 GPIO_CNTL—GPIO Control (LPC I/F—D31:F0)

Offset Address:	5Ch	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:5	Reserved.
4	**GPIO Enable (GPIO_EN)**—R/W. This bit enables/disables decode of the I/O range pointed to by the GPIO base register and enables/disables the GPIO function. 1 = Enable 0 = Disable
3:0	Reserved.

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

Offset Address:	PIRQA–60h, PIRQB–61h, PIRQC–62h, PIRQD–63h	Attribute:	R/W
Default Value:	80808080h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7	**Interrupt Routing Enable (IRQEN)**—R/W. Note that BIOS must program this bit to 0 during POST for any of the PIRQs that are being used. The value of this bit may subsequently be changed by the OS when setting up for I/O APIC interrupt delivery mode. 0 = The corresponding PIRQ is routed to one of the ISA-compatible interrupts specified in bits[3:0]. 1 = The PIRQ is not routed to the 8259.
6:4	Reserved.
3:0	**IRQ Routing**—R/W. (ISA compatible) 0000 = Reserved 0001 = Reserved 0010 = Reserved 0011 = IRQ3 0100 = IRQ4 0101 = IRQ5 0110 = IRQ6 0111 = IRQ7 1000 = Reserved 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11 1100 = IRQ12 1101 = Reserved 1110 = IRQ14 1111 = IRQ15

8.1.17 SERIRQ_CNTL—Serial IRQ Control (LPC I/F—D31:F0)

Offset Address:	64h	Attribute:	R/W
Default Value:	10h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7	**Serial IRQ Enable (SIRQEN)**—R/W. 1 = Serial IRQs will be recognized. The SERIRQ pin will be configured as SERIRQ. 0 = The buffer is input only and internally SERIRQ will be a 1.
6	**Serial IRQ Mode Select (SIRQMD)**—R/W. For systems using Quiet Mode, this bit should be set to 1 (Continuous Mode) for at least one frame after coming out of reset before switching back to Quiet Mode. Failure to do so will result in the **82801E C-ICH** not recognizing SERIRQ interrupts. 1 = The serial IRQ machine will be in continuous mode. 0 = The serial IRQ machine will be in quiet mode.
5:2	**Serial IRQ Frame Size (SIRQSZ)**—R/W. Fixed field that indicates the size of the SERIRQ frame. In the 82801E C-ICH, this field needs to be programmed to 21 frames (0100). This is an offset from a base of 17 which is the smallest data frame size.
1:0	**Start Frame Pulse Width (SFPW)**—R/W. This is the number of PCI clocks that the SERIRQ pin will be driven low by the serial IRQ machine to signal a start frame. In continuous mode, the 82801E C-ICH will drive the start frame for the number of clocks specified. In quiet mode, the 82801E C-ICH will drive the start frame for the number of clocks specified minus one, as the first clock was driven by the peripheral. 00 = 4 clocks 01 = 6 clocks 10 = 8 clocks 11 = Reserved

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

Offset Address:	PIRQE–68h, PIRQF–69h, PIRQG–6Ah, PIRQH–6Bh	Attribute:	R/W
Default Value:	80808080h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7	**Interrupt Routing Enable (IRQEN)**—R/W. Note that BIOS must program this bit to 0 during POST for any of the PIRQs that are being used. The value of this bit may subsequently be changed by the OS when setting up for I/O APIC interrupt delivery mode. 0 = The corresponding PIRQ is routed to one of the ISA-compatible interrupts specified in bits[3:0]. 1 = The PIRQ is not routed to the 8259.
6:4	Reserved.
3:0	**IRQ Routing**—R/W. (ISA compatible) 0000 = Reserved 0001 = Reserved 0010 = Reserved 0011 = IRQ3 0100 = IRQ4 0101 = IRQ5 0110 = IRQ6 0111 = IRQ7 1000 = Reserved 1001 = IRQ9 1010 = IRQ10 1011 = IRQ11 1100 = IRQ12 1101 = Reserved 1110 = IRQ14 1111 = IRQ15

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

Offset Address:	88h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

This register configures the 82801E C-ICH's Device 31 responses to various system errors. The actual assertion of SERR# is enabled via the PCI Command register

.

Bit	Description
7:3	Reserved.
2	**SERR# on Received Target Abort Enable (SERR_RTA_EN)**—R/W. 1 = The 82801E C-ICH will generate SERR# when SERR_RTA is set if SERR_EN is set. 0 = Disable. No SERR# assertion on Received Target Abort.
1	**SERR# on Delayed Transaction Time-out Enable (SERR_DTT_EN)**—R/W. 1 = The 82801E C-ICH will generate SERR# when SERR_DTT bit is set if SERR_EN is set. 0 = Disable. No SERR# assertion on Delayed Transaction Time-out.
0	Reserved.

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

Offset Address:	8Ah	Attribute:	R/WC
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

This register configures the 82801E C-ICH's Device 31 responses to various system errors. The actual assertion of SERR# is enabled via the PCI Command register.

Bit	Description
7:3	Reserved.
2	**SERR# Due to Received Target Abort (SERR_RTA)**—R/WC. 1 = The 82801E C-ICH sets this bit when it receives a target abort. If SERR_EN, the 82801E C-ICH will also generate an SERR# when SERR_RTA is set. 0 = Software clears this bit by writing a 1 to the bit location.
1	**SERR# Due to Delayed Transaction Time-out (SERR_DTT)**—R/WC. 1 = When a PCI master does not return for the data within 1 ms of the cycle's completion, the 82801E C-ICH clears the delayed transaction and sets this bit. If both SERR_DTT_EN and SERR_EN are set, then 82801E C-ICH will also generate an SERR# when SERR_DTT is set. 0 = Software clears this bit by writing a 1 to the bit location.
0	Reserved.

8.1.21 PCI_DMA_CFG—PCI DMA Configuration (LPC I/F—D31:F0)

Offset Address:	90h–91h	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	No	Power Well:	Core

Bit	Description
15:14	**Channel 7 Select**—R/W. 00 = Reserved 01 = PC/PCI DMA 10 = Reserved 11 = LPC I/F DMA
13:12	**Channel 6 Select**—R/W. Same bit decode as for Channel 7
11:10	**Channel 5 Select**—R/W. Same bit decode as for Channel 7
9:8	Reserved.
7:6	**Channel 3 Select**—R/W. Same bit decode as for Channel 7
5:4	**Channel 2 Select**—R/W. Same bit decode as for Channel 7
3:2	**Channel 1 Select**—R/W. Same bit decode as for Channel 7
1:0	**Channel 0 Select**—R/W. Same bit decode as for Channel 7

8.1.22 GEN_CNTL—General Control Register (LPC I/F—D31:F0)

Offset Address:	D0h–D3h	Attribute:	R/W
Default Value:	00000000h	Size:	32-bit
Lockable:	No	Power Well:	Core

Bit	Description
31:26	Reserved.
25	**REQ[5]#/GNT[5]# PC/PCI protocol select (PCPCIB_SEL)**—R/W. 1 = When this bit is set to a 1, the PCI REQ[5]#/GNT[5]# signal pair will use the PC/PCI protocol as REQ[B]#/GNT[B]. The corresponding bits in the GPIO_USE_SEL register must also be set to a 0. If the corresponding bits in the GPIO_USE_SEL register are set to a 1, the signals will be used as a GPI and GPO. 0 = The REQ[5]#/GNT[5]# pins will function as a standard PCI REQ/GNT signal pair.
24	**Hide ISA Bridge (HIDE_ISA)**—R/W. 1 = Software sets this bit to 1 to disable configuration cycle from being claimed by a PCI-to-ISA bridge. This prevents the operating system PCI PnP from getting confused by seeing two ISA bridges. It is required for the 82801E C-ICH PCI address line AD22 to connect to the PCI-to-ISA bridge's IDSEL input. When this bit is 1, the 82801E C-ICH does not assert AD22 during configuration cycles to the PCI-to-ISA bridge. 0 = The 82801E C-ICH does not prevent AD22 from asserting during configuration cycles to the PCI-to-ISA bridge.
23:14	Reserved.
13	**Coprocessor Error Enable (COPR_ERR_EN)**—R/W. 1 = When FERR# is low, 82801E C-ICH generates IRQ13 internally and holds it until an I/O write to port F0h. It will also drive IGNNE# active. 0 = FERR# will not generate IRQ13 nor IGNNE#.
12	**Keyboard IRQ1 Latch Enable (IRQ1LEN)**—R/W. 1 = The active edge of IRQ1 will be latched and held until a port 60h read. 0 = IRQ1 will bypass the latch.

Bit	Description
11	**Mouse IRQ12 Latch Enable (IRQ12LEN)**—R/W. 1 = The active edge of IRQ12 will be latched and held until a port 60h read. 0 = IRQ12 will bypass the latch.
10:9	Reserved.
8[1]	**APIC Enable (APIC_EN)**—R/W. 1 = Enables the internal I/O (x) APIC and its address decode. 0 = Disables internal I/O (x) APIC.
7[1]	**Enables I/O (x) Extension Enable (XAPIC_EN)**—R/W. Note that this bit is only valid if the AIPC_EN bit (bit 8) is also set to 1. 1 = Enables the extra features (beyond standard I/O APIC) associated with the I/O (x) APIC. 0 = The I/O (x) APIC extensions are not supported.
6	**Alternate Access Mode Enable (ALTACC_EN)**—R/W. 1 = Alternate Access Mode Enable 0 = Alternate Access Mode Disabled (default). Alternate Access Mode allows reads to otherwise unreadable registers and writes otherwise unwriteable registers.
5:3	Reserved.
2	**DMA Collection Buffer Enable (DCB_EN)**—R/W. 1 = Enables DMA Collection Buffer (DCB) for LPC I/F and PC/PCI DMA. 0 = DCB disabled.
1	**Delayed Transaction Enable (DTE)**—R/W. 1 = 82801E C-ICH enables delayed transactions for internal register, FWH, and LPC interface accesses. 0 = Delayed transactions disabled. **NOTE:** If there is a downstream I/O 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 82801E C-ICH can erroneously set the STA bit (bit-11) in D31:F0;PCISTS configuration space even though there is no Target Abort on the PCI bus. This has only been observed on DP systems. **NOTE:** 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 that polls this STA bit may incorrectly indicate a Target Abort has occurred. **NOTE:** There are two possible workarounds: • 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. • Disable Delayed Transaction (which may induce a performance penalty on PCI) **NOTE:** See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information on steppings affected by this issue.
0	**Positive Decode Enable (POS_DEC_EN)**—R/W. 1 = Enables 82801E C-ICH to only perform positive decode on the PCI bus. 0 = The 82801E C-ICH performs subtractive decode on the PCI bus and forward the cycles to LPC interface if not to an internal register or other known target on the LPC interface. Accesses to internal registers and to known LPC interface devices are still be positively decoded.

NOTES:

Rule 1: If bit 8 is 0, the 82801E C-ICH does not decode any of the registers associated with the I/O APIC or I/O (x) APIC. The state of bit 7 is a “Don’t Care” in this case.

Rule 2: If bit 8 is 1 and bit 7 is 0, the 82801E C-ICH decodes the memory space associated with the I/O APIC, but not the extra registers associated with the I/O (x) APIC.

Rule 3: If bit 8 is 1 and bit 7 is 1, the 82801E C-ICH decodes the memory space associated with both the I/O APIC and the I/O (x) APIC. This also enables PCI masters to write directly to the register to cause interrupts (PCI Message Interrupt).

Note that there is no separate way to disable PCI Message Interrupts if the I/O (x) APIC is enabled. This is not considered necessary.

8.1.23 GEN_STS—General Status (LPC I/F—D31:F0)

Offset Address:	D4h–D7h	Attribute:	R/W
Default Value:	00000F00h	Size:	8-bit
Lockable:	No	Power Well:	Core(0:7)

Bit	Description
7:3	Reserved.
2	**SAFE_MODE**—RO. 1 = 82801E C-ICH sampled AC_SDOUT high on the rising edge of PWROK. 82801E C-ICH will force FREQ_STRAP[3:0] bits to all 1s (safe mode multiplier). 0 = 82801E C-ICH sampled AC_SDOUT low on the rising edge of PWROK.
1	**NO_REBOOT**—R/W (special). 1 = 82801E C-ICH will disable the TCO Timer system reboot feature. This bit is set either by hardware when SPKR is sampled low on the rising edge of PWROK or by software writing a 1 to the bit. 0 = Normal TCO Timer reboot functionality (reboot after 2nd TCO time-out). Note that this bit cannot be cleared while an external jumper is in place on the SPKR signal.
0	Reserved.

8.1.24 RTC_CONF—RTC Configuration Register (LPC I/F—D31:F0)

Offset Address:	D8h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	Yes	Power Well:	Core

Bit	Description
7:5	Reserved.
4	**Upper 128-byte Lock (U128LOCK)**—R/W (special). 1 = Lock reads and writes to bytes 38h–3Fh in the upper 128 byte bank of the RTC CMOS RAM. Write cycles to this range will have no effect and read cycles will not return any particular guaranteed value. This is a write once register that can only be reset by a hardware reset. 0 = Access to these bytes in the upper CMOS RAM range have not been locked.
3	**Lower 128-byte Lock (L128LOCK)**—R/W (special). 1 = Locks reads and writes to bytes 38h–3Fh in the lower 128 byte bank of the RTC CMOS RAM. Write cycles to this range will have no effect and read cycles will not return any particular guaranteed value. This is a write once register that can only be reset by a hardware reset. 0 = Access to these bytes in the lower CMOS RAM range have not been locked.
2	**Upper 128-byte Enable (U128E)**—R/W. 1 = Enables access to the upper 128 byte bank of RTC CMOS RAM. 0 = Disable.
1:0	Reserved.

8.1.25 COM_DEC—LPC I/F Communication Port Decode Ranges (LPC I/F—D31:F0)

Offset Address:	E0h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7	Reserved.
6:4	**COMB Decode Range**—R/W. This field determines which range to decode for the COMB Port. 000 = 3F8h–3FFh (COM1) 001 = 2F8h–2FFh (COM2) 010 = 220h–227h 011 = 228h–22Fh 100 = 238h–23Fh 101 = 2E8h–2EFh (COM4) 110 = 338h–33Fh 111 = 3E8h–3EFh (COM3)
3	Reserved.
2:0	**COMA Decode Range**—R/W. This field determines which range to decode for the COMA Port. 000 = 3F8h–3FFh (COM1) 001 = 2F8h–2FFh (COM2) 010 = 220h–227h 011 = 228h–22Fh 100 = 238h–23Fh 101 = 2E8h–2EFh (COM4) 110 = 338h–33Fh 111 = 3E8h–3EFh (COM3)

8.1.26 FDD/LPT_DEC—LPC I/F FDD & LPT Decode Ranges (LPC I/F—D31:F0)

Offset Address:	E1h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:5	Reserved.
4	**FDD Decode Range**—R/W. Determines which range to decode for the FDD Port 0 = 3F0h–3F5h, 3F7h (Primary) 1 = 370h–2FFh (Secondary)
3:2	Reserved.
1:0	**LPT Decode Range**—R/W. This field determines which range to decode for the LPT Port. 00 = 378h–37Fh and 778h–77Fh 01 = 278h–27Fh (port 279h is read only) and 678h–67Fh 10 = 3BCh–3BEh and 7BCh–7BEh 11 = Reserved

8.1.27 SND_DEC—LPC I/F Sound Decode Ranges (LPC I/F—D31:F0)

Offset Address:	E2h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:6	Reserved.
5:4	**MSS Decode Range**—R/W. This field determines which range to decode for the Microsoft* Sound System (MSS). 00 = 530h–537h 01 = 604h–60Bh 10 = E80h–E87h 11 = F40h–F47h
3	**MIDI Decode Range**—R/W. This bit determines which range to decode for the Midi Port. 0 = 330h–331h 1 = 300h–301h
2	Reserved.
1:0	**SB16 Decode Range**—R/W. This field determines which range to decode for the Sound Blaster 16 (SB16) Port. 00 = 220h–233h 01 = 240h–253h 10 = 260h–273h 11 = 280h–293h

8.1.28 FWH_DEC_EN1—FWH Decode Enable 1 Register (LPC I/F—D31:F0)

Offset Address:	E3h	Attribute:	R/W
Default Value:	FFh	Size:	8 bits

This register determines which memory ranges will be decoded on the PCI bus and forwarded to the FWH. The 82801E C-ICH will subtractively decode cycles on PCI unless POS_DEC_EN is set to 1.

Bit	Description
7	**FWH Address Range Enable (FWH_F8_EN)**—RO. Enables decoding two 512 Kbyte FWH memory ranges and one 128 Kbyte memory range. 1 = Enable the following ranges for the FWH FFF80000h–FFFFFFFFh FFB80000h–FFBFFFFFh 000E0000h–000FFFFFh
6	**FWH Address Range Enable (FWH_F0_EN)**—R/W. Enables decoding two 512 Kbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH: FFF00000h–FFF7FFFFh FFB00000h–FFB7FFFFh

Bit	Description
5	**FWH Address Range Enable (FWH_E8_EN)**—R/W. Enables decoding two 512 Kbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH: FFE80000h–FFEFFFFFh FFA80000h–FFAFFFFFh
4	**FWH Address Range Enable (FWH_E0_EN)**—R/W. Enables decoding two 512 Kbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH: FFE00000h–FFE7FFFFh FFA00000h–FFA7FFFFh
3	**FWH Address Range Enable (FWH_D8_EN)**—R/W. Enables decoding two 512 Kbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FFD80000h–FFDFFFFFh FF980000h–FF9FFFFFh
2	**FWH Address Range Enable (FWH_D0_EN)**—R/W. Enables decoding two 512 Kbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FFD00000h–FFD7FFFFh FF900000h–FF97FFFFh
1	**FWH Address Range Enable (FWH_C8_EN)**—R/W. Enables decoding two 512 Kbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FFC80000h–FFCFFFFFh FF880000h–FF8FFFFFh
0	**FWH Address Range Enable (FWH_C0_EN)**—R/W. Enables decoding two 512 Kbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FFC00000h–FFC7FFFFh FF800000h–FF87FFFFh

8.1.29 GEN1_DEC—LPC I/F Generic Decode Range 1 (LPC I/F—D31:F0)

Offset Address:	E4h–E5h	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	Yes	Power Well:	Core

Bit	Description
15:7	**Generic I/O Decode Range 1 Base Address (GEN1_BASE)**—R/W. This address is aligned on a 128-byte boundary, and must have address lines 31:16 as 0. Note that this generic decode is for I/O addresses only, not memory addresses. The size of this range is 128 bytes.
6:1	Reserved.
0	**Generic Decode Range 1 Enable (GEN1_EN)**—R/W. 0 = Disable. 1 = Enable the GEN1 I/O range to be forwarded to the LPC I/F

8.1.30 LPC_EN—LPC I/F Enables (LPC I/F—D31:F0)

Offset Address:	E6h–E7h	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	Yes	Power Well:	Core

Bit	Description
15:14	Reserved.
13	**Microcontroller Address Range Enable (CNF2_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 4Eh and 4Fh to the LPC interface. This range is used for a microcontroller.
12	**Super I/O Address Range Enable (CNF1_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 2Eh and 2Fh to the LPC interface. This range is used for Super I/O devices.
11	**Microcontroller Address Range Enable (MC_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 62h and 66h to the LPC interface. This range is used for a microcontroller.
10	**Microcontroller Address Range Enable (KBC_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 60h and 64h to the LPC interface. This range is used for a microcontroller.
9	**Game Port Address Range Enable (GAMEH_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 208h to 20Fh to the LPC interface. This range is used for a gameport.
8	**Game Port Address Range Enable (GAMEL_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 200h to 207h to the LPC interface. This range is used for a gameport.
7	**ADLIB Address Range Enable (ADLIB_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the I/O locations 388h–38Bh to the LPC interface.
6	**MSS Address Range Enable (MSS_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the MSS range to the LPC interface. This range is selected in the LPC_Sound Decode Range Register.
5	**MIDI Address Range Enable (MIDI_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the MIDI range to the LPC interface. This range is selected in the LPC_Sound Decode Range Register.
4	**Sound Blaster Address Range Enable (SB16_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the SB16 range to the LPC interface. This range is selected in the LPC_Sound Decode Range Register.
3	**FDD Address Range Enable (FDD_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the FDD range to the LPC interface. This range is selected in the LPC_FDD/LPT Decode Range Register.

Bit	Description
2	**LPT Address Range Enable (LPT_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the LPT range to the LPC interface. This range is selected in the LPC_FDD/LPT Decode Range Register.
1	**COM B Address Range Enable (COMB_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the COMB range to the LPC interface. This range is selected in the LPC_COM Decode Range Register.
0	**Com A Address Range Enable (COMA_LPC_EN)**—R/W. 0 = Disable. 1 = Enables the decoding of the COMA range to the LPC interface. This range is selected in the LPC_COM Decode Range Register.

8.1.31 FWH_SEL1—FWH Select 1 Register (LPC I/F—D31:F0)

Offset Address: E8h–EBh Attribute: R/W
Default Value: 00112233h Size: 32 bits

Bit	Description
31:28	**FWH Address Range Select (FWH_F8_IDSEL)**—RO. IDSEL for two 512 Kbyte FWH memory ranges and one 128 Kbyte memory range. This field is fixed at 0000. The IDSEL in this field addresses the following memory ranges: FFF8 0000h–FFFF FFFFh FFB8 0000h–FFBF FFFFh 000E 0000h–000F FFFFh
27:24	**FWH Address Range Select (FWH_F0_IDSEL)**—R/W. IDSEL for two 512 Kbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFF0 0000h–FFF7 FFFFh FFB0 0000h–FFB7 FFFFh
23:20	**FWH Address Range Select (FWH_E8_IDSEL)**—R/W. IDSEL for two 512 Kbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFE8 0000h–FFEF FFFFh FFA8 0000h–FFAF FFFFh
19:16	**FWH Address Range Select (FWH_E0_IDSEL)**—R/W. IDSEL for two 512 Kbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFE0 0000h–FFE7 FFFFh FFA0 0000h–FFA7 FFFFh
15:12	**FWH Address Range Select (FWH_D8_IDSEL)**—R/W. IDSEL for two 512 Kbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFD8 0000h–FFDF FFFFh FF98 0000h–FF9F FFFFh
11:8	**FWH Address Range Select (FWH_D0_IDSEL)**—R/W. IDSEL for two 512 Kbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFD0 0000h–FFD7 FFFFh FF90 0000h–FF97 FFFFh
7:4	**FWH Address Range Select (FWH_C8_IDSEL)**—R/W. IDSEL for two 512 Kbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFC8 0000h–FFCF FFFFh FF88 0000h–FF8F FFFFh
3:0	**FWH Address Range Select (FWH_C0_IDSEL)**—R/W. IDSEL for two 512 Kbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FFC0 0000h–FFC7 FFFFh FF80 0000h–FF87 FFFFh

8.1.32 GEN2_DEC—LPC I/F Generic Decode Range 2 (LPC I/F—D31:F0)

Offset Address:	ECh–EDh	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	Yes	Power Well:	Core

Bit	Description
15:4	**Generic I/O Decode Range 2 Base Address (GEN2_BASE)**—R/W. This address is aligned on a 64-byte boundary and must have address lines 31:16 as 0. Note that this generic decode is for I/O addresses only; not memory addresses. The size of this range is 16 bytes.
3:1	Reserved. Read as 0
0	**Generic I/O Decode Range 2 Enable (GEN2_EN)**—R/W. 0 = Disable. 1 = Accesses to the GEN2 I/O range will be forwarded to the LPC interface.

8.1.33 FWH_SEL2—FWH Select 2 Register (LPC I/F—D31:F0)

Offset Address:	EEh–EFh	Attribute:	R/W
Default Value:	4567h	Size:	32 bits

Bit	Description
15:12	**FWH Address Range Select (FWH_70_IDSEL)**—R/W. IDSEL for two 1 Mbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF70 0000h–FF7F FFFFh FF30 0000h–FF3F FFFFh
11:8	**FWH Address Range Select (FWH_60_IDSEL)**—R/W. IDSEL for two 1 Mbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF60 0000h–FF6F FFFFh FF20 0000h–FF2F FFFFh
7:4	**FWH Address Range Select (FWH_50_IDSEL)**—R/W. IDSEL for two 1 Mbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF50 0000h–FF5F FFFFh FF10 0000h–FF1F FFFFh
3:0	**FWH Address Range Select (FWH_40_IDSEL)**—R/W. IDSEL for two 1 Mbyte FWH memory ranges. The IDSEL programmed in this field addresses the following memory ranges: FF40 0000h–FF4F FFFFh FF00 0000h–FF0F FFFFh

8.1.34 FWH_DEC_EN2—FWH Decode Enable 2 Register (LPC I/F—D31:F0)

Offset Address:	F0h	Attribute:	R/W
Default Value:	0Fh	Size:	8 bits

This register determines which memory ranges are decoded on the PCI bus and forwarded to the FWH. The 82801E C-ICH subtractively decodes cycles on PCI unless POS_DEC_EN is set to 1.

Bit	Description
7:4	Reserved.
3	**FWH Address Range Enable (FWH_70_EN)**—R/W. Enables decoding two 1 Mbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FF70 0000h–FF7F FFFFh FF30 0000h–FF3F FFFFh
2	**FWH Address Range Enable (FWH_60_EN)**—R/W. Enables decoding two 1 Mbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FF60 0000h–FF6F FFFFh FF20 0000h–FF2F FFFFh
1	**FWH Address Range Enable (FWH_50_EN)**—R/W. Enables decoding two 1 Mbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FF50 0000h–FF5F FFFFh FF10 0000h–FF1F FFFFh
0	**FWH Address Range Enable (FWH_40_EN)**—R/W. Enables decoding two 1 Mbyte FWH memory ranges. 0 = Disable. 1 = Enable the following ranges for the FWH FF40 0000h–FF4F FFFFh FF00 0000h–FF0F FFFFh

8.1.35 FUNC_DIS—Function Disable Register (LPC I/F—D31:F0)

Offset Address:	F2h	Attribute:	R/W
Default Value:	00h	Size:	16-bit
Lockable:	No	Power Well:	Core

Bit	Description
15:9	Reserved.
8	**SMBus For BIOS (SMB_FOR_BIOS)**—R/W. This bit is used in conjunction with bit 3 in this register. 0 = No effect. 1 = Allows the SMBus I/O space to be accessible by software when bit 3 in this register is set. The PCI configuration space is hidden in this case. Note that if bit 3 is set alone, the decode of both SMBus PCI configuration and I/O space will be disabled.
7:4	Reserved.
3	**SMBus Controller Disable (F3_Disable)**—R/W. Software sets this bit to disable the SMBus Host Controller function. BIOS must not enable I/O or memory address space decode, interrupt generation or any other functionality for functions that are to be disabled. 0 = SMBus controller is enabled 1 = SMBus controller is disabled
2	**USB Controller 1 Disable (F2_Disable)**—R/W. Software sets this bit to disable the USB Controller #1 function. BIOS must not enable I/O or memory address space decode, interrupt generation or any other functionality for functions that are to be disabled. 0 = USB Controller #1 is enabled 1 = USB Controller #1 is disabled
1	**IDE Controller Disable (F1_Disable)**—R/W. Software sets this bit to disable the IDE controller function. BIOS must not enable I/O or memory address space decode, interrupt generation or any other functionality for functions that are to be disabled. 0 = IDE controller is enabled 1 = IDE controller is disabled
0	Reserved.

Note: The Function Disable bits really only hide the configuration space by blocking configuration cycles. They do not really disable the unit. They do not prevent the individual function from decoding I/O or memory cycles if so enabled. If software has previously enabled a unit and set the IOSE and MSE, and BME bits, then it should disable them before setting the corresponding Fx_DISABLE bit. Likewise, any interrupt logic associated with the unit should also be disabled before setting the Fx_DISABLE bit.

8.2 DMA I/O Registers

Table 93. DMA Registers (Sheet 1 of 2)

Port	Alias	Register Name/Function	Default	Type
00h	10h	Channel 0 DMA Base & Current Address Register	Undefined	R/W
01h	11h	Channel 0 DMA Base & Current Count Register	Undefined	R/W
02h	12h	Channel 1 DMA Base & Current Address Register	Undefined	R/W
03h	13h	Channel 1 DMA Base & Current Count Register	Undefined	R/W
04h	14h	Channel 2 DMA Base & Current Address Register	Undefined	R/W
05h	15h	Channel 2 DMA Base & Current Count Register	Undefined	R/W
06h	16h	Channel 3 DMA Base & Current Address Register	Undefined	R/W
07h	17h	Channel 3 DMA Base & Current Count Register	Undefined	R/W
08h	18h	Channel 0–3 DMA Command Register	Undefined	WO
		Channel 0–3 DMA Status Register	Undefined	RO
0Ah	1Ah	Channel 0–3 DMA Write Single Mask Register	000001XXb	WO
0Bh	1Bh	Channel 0–3 DMA Channel Mode Register	000000XXb	WO
0Ch	1Ch	Channel 0–3 DMA Clear Byte Pointer Register	Undefined	WO
0Dh	1Dh	Channel 0–3 DMA Master Clear Register	Undefined	WO
0Eh	1Eh	Channel 0–3 DMA Clear Mask Register	Undefined	WO
0Fh	1Fh	Channel 0–3 DMA Write All Mask Register	0Fh	R/W
80h	90h	Reserved Page Register	Undefined	R/W
81h	91h	Channel 2 DMA Memory Low Page Register	Undefined	R/W
82h	–	Channel 3 DMA Memory Low Page Register	Undefined	R/W
83h	93h	Channel 1 DMA Memory Low Page Register	Undefined	R/W
84h–86h	94h–96h	Reserved Page Registers	Undefined	R/W
87h	97h	Channel 0 DMA Memory Low Page Register	Undefined	R/W
88h	98h	Reserved Page Register	Undefined	R/W
89h	99h	Channel 6 DMA Memory Low Page Register	Undefined	R/W
8Ah	9Ah	Channel 7 DMA Memory Low Page Register	Undefined	R/W
8Bh	9Bh	Channel 5 DMA Memory Low Page Register	Undefined	R/W
8Ch–8Eh	9Ch–9Eh	Reserved Page Registers	Undefined	R/W
8Fh	9Fh	Refresh Low Page Register	Undefined	R/W
C0h	C1h	Channel 4 DMA Base & Current Address Register	Undefined	R/W
C2h	C3h	Channel 4 DMA Base & Current Count Register	Undefined	R/W
C4h	C5h	Channel 5 DMA Base & Current Address Register	Undefined	R/W
C6h	C7h	Channel 5 DMA Base & Current Count Register	Undefined	R/W
C8h	C9h	Channel 6 DMA Base & Current Address Register	Undefined	R/W
CAh	CBh	Channel 6 DMA Base & Current Count Register	Undefined	R/W
CCh	CDh	Channel 7 DMA Base & Current Address Register	Undefined	R/W

Table 93. DMA Registers (Sheet 2 of 2)

Port	Alias	Register Name/Function	Default	Type
CEh	CFh	Channel 7 DMA Base & Current Count Register	Undefined	R/W
D0h	D1h	Channel 4–7 DMA Command Register	Undefined	WO
		Channel 4–7 DMA Status Register	Undefined	RO
D4h	D5h	Channel 4–7 DMA Write Single Mask Register	000001XXb	WO
D6h	D7h	Channel 4–7 DMA Channel Mode Register	000000XXb	WO
D8h	D9h	Channel 4–7 DMA Clear Byte Pointer Register	Undefined	WO
DAh	DBh	Channel 4–7 DMA Master Clear Register	Undefined	WO
DCh	DDh	Channel 4–7 DMA Clear Mask Register	Undefined	WO
DEh	DFh	Channel 4–7 DMA Write All Mask Register	0Fh	R/W

8.2.1 DMABASE_CA—DMA Base and Current Address Registers

I/O Address:	Ch. #0 = 00h; Ch. #1 = 02h Ch. #2 = 04h; Ch. #3 = 06h Ch. #5 = C4h Ch. #6 = C8h Ch. #7 = CCh;	Attribute:	RO
		Size:	16-bit (per channel), but accessed in two 8-bit quantities
Default Value:	Undef		
Lockable:	No	Power Well:	Core

Bit	Description
15:0	**Base and Current Address**—R/W. This register determines the address for the transfers to be performed. The address specified points to two separate registers. On writes, the value is stored in the *Base Address* register and copied to the *Current Address* register. On reads, the value is returned from the *Current Address* register. 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 out. Therefore, if bit 15 was a 1, it will be lost. The register is accessed in 8 bit quantities. The byte is pointed to by the current byte pointer flip/flop. Before accessing an address register, the byte pointer flip/flop should be cleared to ensure that the low byte is accessed first.

8.2.2 DMABASE_CC—DMA Base and Current Count Registers

I/O Address:	Ch. #0 = 01h; Ch. #1 = 03h Ch. #2 = 05h; Ch. #3 = 07h Ch. #5 = C6h; Ch. #6 = CAh Ch. #7 = CEh;	Attribute: Size:	R/W 16-bit (per channel), but accessed in two 8-bit quantities
Default Value:	Undefined		
Lockable:	No	Power Well:	Core

Bit	Description
15:0	**Base and Current Count**—R/W. This register determines the number of transfers to be performed. The address specified points to two separate registers. On writes the value is stored in the *Base Count* register and copied to the *Current Count* register. On reads the value is returned from the *Current Count* register. The actual number of transfers is one more than the number programmed in the Base Count Register (i.e., programming a count of 4h results in 5 transfers). The count is decremented in the Current Count register after each transfer. When the value in the register rolls from zero to FFFFh, a terminal count is generated. If the channel is in auto-initialize mode, the Current Count register will be reloaded from the Base Count register after a terminal count is generated. For transfers to/from an 8-bit slave (channels 0–3), the count register indicates the number of bytes to be transferred. For transfers to/from a 16-bit slave (channels 5–7), the count register indicates the number of words to be transferred. The register is accessed in 8 bit quantities. The byte is pointed to by the current byte pointer flip/flop. Before accessing a count register, the byte pointer flip/flop should be cleared to ensure that the low byte is accessed first.

8.2.3 DMAMEM_LP—DMA Memory Low Page Registers

I/O Address:	Ch. #0 = 87h; Ch. #1 = 83h Ch. #2 = 81h; Ch. #3 = 82h Ch. #5 = 8Bh; Ch. #6 = 89h Ch. #7 = 8Ah;	Attribute:	R/W
Default Value:	Undefined	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:0	**DMA Low Page (ISA Address bits [23:16])**—R/W. This register works in conjunction with the DMA controller's Current Address Register to define the complete 24-bit address for the DMA channel. This register remains static throughout the DMA transfer.

8.2.4 DMACMD—DMA Command Register

I/O Address:	Ch. #0–3 = 08h; Ch. #4–7 = D0h	Attribute:	WO
Default Value:	Undefined	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:5	Reserved. Must be 0.
4	**DMA Group Arbitration Priority**—WO. Each channel group is individually assigned either fixed or rotating arbitration priority. At part reset, each group is initialized in fixed priority. 0 = Fixed priority to the channel group 1 = Rotating priority to the group.
3	Reserved. Must be 0
2	**DMA Channel Group Enable**—WO. Both channel groups are enabled following part reset. 0 = Enable the DMA channel group. 1 = Disable. Disabling channel group 4–7 also disables channel group 0–3, which is cascaded through channel 4.
1:0	Reserved. Must be 0.

8.2.5 DMASTS—DMA Status Register

I/O Address:	Ch. #0–3 = 08h; Ch. #4–7 = D0h	Attribute:	RO
Default Value:	Undefined	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:4	**Channel Request Status**—RO. When a valid DMA request is pending for a channel, the corresponding bit is set to 1. When a DMA request is not pending for a particular channel, the corresponding bit is set to 0. The source of the DREQ may be hardware or a software request. Note that channel 4 is the cascade channel, so the request status of channel 4 is a logical OR of the request status for channels 0 through 3. 4 = Channel 0 5 = Channel 1 (5) 6 = Channel 2 (6) 7 = Channel 3 (7)
3:0	**Channel Terminal Count Status**—RO. When a channel reaches terminal count (TC), its status bit is set to 1. If TC has not been reached, the status bit is set to 0. Channel 4 is programmed for cascade, so the TC bit response for channel 4 is irrelevant. 0 = Channel 0 1 = Channel 1 (5) 2 = Channel 2 (6) 3 = Channel 3 (7)

8.2.6 DMA_WRSMSK—DMA Write Single Mask Register

I/O Address:	Ch. #0–3 = 0Ah; Ch. #4–7 = D4h	Attribute:	WO
Default Value:	0000 01xx	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:3	Reserved. Must be 0.
2	**Channel Mask Select**—WO. 0 = Enable DREQ for the selected channel. The channel is selected through bits [1:0]. Therefore, only one channel can be masked/unmasked at a time. 1 = Disable DREQ for the selected channel.
1:0	**DMA Channel Select**—WO. These bits select the DMA Channel Mode Register to program. 00 = Channel 0 (4) 01 = Channel 1 (5) 10 = Channel 2 (6) 11 = Channel 3 (7)

8.2.7 DMACH_MODE—DMA Channel Mode Register

I/O Address:	Ch. #0–3 = 0Bh; Ch. #4–7 = D6h	Attribute:	WO
Default Value:	0000 00xx	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:6	**DMA Transfer Mode**—WO. Each DMA channel can be programmed in one of four different modes: 00 = Demand mode 01 = Single mode 10 = Reserved 11 = Cascade mode
5	**Address Increment/Decrement Select**—WO. This bit controls address increment/decrement during DMA transfers. 0 = Address increment. (default after part reset or Master Clear) 1 = Address decrement.
4	**Autoinitialize Enable**—WO. 0 = Autoinitialize feature is disabled and DMA transfers terminate on a terminal count. A part reset or Master Clear disables autoinitialization. 1 = DMA restores the Base Address and Count registers to the current registers following a terminal count (TC).
3:2	**DMA Transfer Type**—WO. These bits represent the direction of the DMA transfer. When the channel is programmed for cascade mode, (bits[7:6] = “11”) the transfer type is irrelevant. 00 = Verify - No I/O or memory strobes generated 01 = Write - Data transferred from the I/O devices to memory 10 = Read - Data transferred from memory to the I/O device 11 = Illegal
1:0	**DMA Channel Select**—WO. These bits select the DMA Channel Mode Register that will be written by bits [7:2]. 00 = Channel 0 (4) 01 = Channel 1 (5) 10 = Channel 2 (6) 11 = Channel 3 (7)

8.2.8 DMA Clear Byte Pointer Register

I/O Address:	Ch. #0–3 = 0Ch; Ch. #4–7 = D8h	Attribute:	WO
Default Value:	xxxx xxxx	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:0	**Clear Byte Pointer**—WO. No specific pattern. Command enabled with a write to the I/O port address. Writing to this register initializes the byte pointer flip/flop to a known state. It clears the internal latch used to address the upper or lower byte of the 16-bit Address and Word Count Registers. The latch is also cleared by part reset and by the Master Clear command. This command precedes the first access to a 16-bit DMA controller register. The first access to a 16 bit register will then access the significant byte, and the second access automatically accesses the most significant byte.

8.2.9 DMA Master Clear Register

I/O Address:	Ch. #0–3 = 0Dh; Ch. #4–7 = DAh	Attribute:	WO
Default Value:	xxxx xxxx	Size:	8-bit

Bit	Description
7:0	**Master Clear**—WO. No specific pattern. Enabled with a write to the port. This has the same effect as the hardware Reset. The Command, Status, Request, and Byte Pointer flip/flop registers are cleared and the Mask Register is set.

8.2.10 DMA_CLMSK—DMA Clear Mask Register

I/O Address:	Ch. #0–3 = 0Eh; Ch. #4–7 = DCh	Attribute:	WO
Default Value:	xxxx xxxx	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:0	**Clear Mask Register**—WO. No specific pattern. Command enabled with a write to the port.

8.2.11 DMA_WRMSK—DMA Write All Mask Register

I/O Address:	Ch. #0–3 = 0Fh; Ch. #4–7 = DEh	Attribute:	R/W
Default Value:	0000 1111	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:4	Reserved. Must be 0.
3:0	**Channel Mask Bits**—R/W. This register permits all four channels to be simultaneously enabled/disabled instead of enabling/disabling each channel individually, as is the case with the Mask Register - Write Single Mask Bit. In addition, this register has a read path to allow the status of the channel mask bits to be read. A channel's mask bit is automatically set to 1 when the Current Byte/Word Count Register reaches terminal count (unless the channel is in auto-initialization mode). Setting the bit(s) to a 1 disables the corresponding DREQ(s). Setting the bit(s) to a 0 enables the corresponding DREQ(s). Bits [3:0] are set to 1 upon part reset or Master Clear. When read, bits [3:0] indicate the DMA channel [3:0] ([7:4]) mask status. Bit 0 = Channel 0 (4) 1 = Masked, 0 = Not Masked Bit 1 = Channel 1 (5) 1 = Masked, 0 = Not Masked Bit 2 = Channel 2 (6) 1 = Masked, 0 = Not Masked Bit 3 = Channel 3 (7) 1 = Masked, 0 = Not Masked **NOTE:** Disabling channel 4 also disables channels 0–3 due to the cascade of channels 0–3 through channel 4.

8.3 Timer I/O Registers

Port	Aliases	Register Name/Function	Default Value	Type
40h	50h	Counter 0 Interval Time Status Byte Format	0XXXXXXXb	RO
		Counter 0 Counter Access Port Register	Undefined	R/W
41h	51h	Counter 1 Interval Time Status Byte Format	0XXXXXXXb	RO
		Counter 1 Counter Access Port Register	Undefined	R/W
42h	52h	Counter 2 Interval Time Status Byte Format	0XXXXXXXb	RO
		Counter 2 Counter Access Port Register	Undefined	R/W
43h	53h	Timer Control Word Register	Undefined	WO
		Timer Control Word Register Read Back	XXXXXXX0b	WO
		Counter Latch Command	X0h	WO

8.3.1 TCW—Timer Control Word Register

I/O Address:	43h	Attribute:	WO
Default Value:	All bits undefined	Size:	8 bits

This register is programmed prior to any counter being accessed to specify counter modes. Following part reset, the control words for each register are undefined and each counter output is 0. Each timer must be programmed to bring it into a known state.

Bit	Description
7:6	**Counter Select**—WO. The Counter Selection bits select the counter the control word acts upon as shown below. The Read Back Command is selected when bits[7:6] are both 1. 00 = Counter 0 select 01 = Counter 1 select 10 = Counter 2 select 11 = Read Back Command
5:4	**Read/Write Select**—WO. These bits are the read/write control bits. The actual counter programming is done through the counter port (40h for counter 0, 41h for counter 1, and 42h for counter 2). 00 = Counter Latch Command 01 = Read/Write Least Significant Byte (LSB) 10 = Read/Write Most Significant Byte (MSB) 11 = Read/Write LSB then MSB
3:1	**Counter Mode Selection**—WO. These bits select one of six possible modes of operation for the selected counter. 000 = Mode 0 Out signal on end of count (=0) 001 = Mode 1 Hardware retriggerable one-shot x10 = Mode 2 Rate generator (divide by n counter) x11 = Mode 3 Square wave output 100 = Mode 4 Software triggered strobe 101 = Mode 5 Hardware triggered strobe
0	**Binary/BCD Countdown Select**—WO. 0 = Binary countdown is used. The largest possible binary count is 2^{16} 1 = Binary coded decimal (BCD) count is used. The largest possible BCD count is 10^{4}

There are two special commands that can be issued to the counters through this register, the Read Back Command and the Counter Latch Command. When these commands are chosen, several bits within this register are redefined. These register formats are described below.

8.3.1.1 RDBK_CMD—Read Back Command

The Read Back Command is used to determine the count value, programmed mode, and current states of the OUT pin and Null count flag of the selected counter or counters. Status and/or count may be latched in any or all of the counters by selecting the counter during the register write. The count and status remain latched until read, and further latch commands are ignored until the count is read. Both count and status of the selected counters may be latched simultaneously by setting both bit 5 and bit 4 to 0. If both are latched, the first read operation from that counter returns the latched status. The next one or two reads, depending on whether the counter is programmed for one or two byte counts, returns the latched count. Subsequent reads return an unlatched count.

Bit	Description
7:6	**Read Back Command.** This field must be “11” to select the Read Back Command.
5	**Latch Count of Selected Counters.** 0 = Current count value of the selected counters will be latched 1 = Current count will not be latched
4	**Latch Status of Selected Counters.** 0 = Status of the selected counters will be latched 1 = Status will not be latched
3	**Counter 2 Select.** 1 = Counter 2 count and/or status will be latched
2	**Counter 1 Select.** 1 = Counter 1 count and/or status will be latched
1	**Counter 0 Select.** 1 = Counter 0 count and/or status will be latched.
0	Reserved. Must be 0.

8.3.1.2 LTCH_CMD—Counter Latch Command

The Counter Latch Command latches the current count value. This command is used to insure that the count read from the counter is accurate. The count value is then read from each counter’s count register through the Counter Ports Access Ports Register (40h for counter 0, 41h for counter 1, and 42h for counter 2). The count must be read according to the programmed format (i.e., if the counter is programmed for two byte counts, two bytes must be read). The two bytes do not have to be read one right after the other (read, write, or programming operations for other counters may be inserted between the reads). If a counter is latched once and then latched again before the count is read, the second Counter Latch Command is ignored.

Bit	Description
7:6	**Counter Selection.** These bits select the counter for latching. If “11” is written, then the write is interpreted as a read back command. 00 = Counter 0 01 = Counter 1 10 = Counter 2
5:4	**Counter Latch Command.** 00 = Selects the Counter Latch Command.
3:0	Reserved. Must be 0.

8.3.2 SBYTE_FMT—Interval Timer Status Byte Format Register

I/O Address:	Counter 0 = 40h, Counter 1 = 41h, Counter 2 = 42h	Attribute: Size:	RO 8 bits per counter
Default Value:	Bits[6:0] undefined, Bit 7=0		

Each counter's status byte can be read following a Read Back Command. If latch status is chosen (bit 4=0, Read Back Command) as a read back option for a given counter, the next read from the counter's Counter Access Ports Register (40h for counter 0, 41h for counter 1, and 42h for counter 2) returns the status byte. The status byte returns the following:

Bit	Description
7	**Counter OUT Pin State**—RO. 0 = OUT pin of the counter is also a 0. 1 = OUT pin of the counter is also a 1.
6	**Count Register Status**—RO. This bit indicates when the last count written to the Count Register (CR) has been loaded into the counting element (CE). The exact time this happens depends on the counter mode, but until the count is loaded into the counting element (CE), the count value will be incorrect. 0 = Count has been transferred from CR to CE and is available for reading. 1 = Null Count. Count has not been transferred from CR to CE and is not yet available for reading.
5:4	**Read/Write Selection Status**—RO. These reflect the read/write selection made through bits[5:4] of the control register. The binary codes returned during the status read match the codes used to program the counter read/write selection. 00 = Counter Latch Command 01 = Read/Write Least Significant Byte (LSB) 10 = Read/Write Most Significant Byte (MSB) 11 = Read/Write LSB then MSB
3:1	**Mode Selection Status**—RO. These bits return the counter mode programming. The binary code returned matches the code used to program the counter mode, as listed under the bit function above. 000 = Mode 0 Out signal on end of count (=0) 001 = Mode 1 Hardware retriggerable one-shot x10 = Mode 2 Rate generator (divide by n counter) x11 = Mode 3 Square wave output 100 = Mode 4 Software triggered strobe 101 = Mode 5 Hardware triggered strobe
0	**Countdown Type Status**—RO. This bit reflects the current countdown type. 0 = Binary countdown 1 = Binary Coded Decimal (BCD) countdown.

8.3.3 Counter Access Ports Register

I/O Address:	Counter 0 = 40h, Counter 1 = 41h, Counter 2 = 42h	Attribute:	R/W
Default Value:	All bits undefined	Size:	8 bit

Bit	Description
7:0	**Counter Port**—R/W. Each counter port address is used to program the 16-bit Count Register. The order of programming (either LSB only, MSB only, or LSB then MSB) is defined with the Interval Counter Control Register at port 43h. The counter port is also used to read the current count from the Count Register, and return the status of the counter programming following a Read Back Command.

8.4 8259 Interrupt Controller (PIC) Registers

8.4.1 Interrupt Controller I/O MAP

The interrupt controller registers are located at 20h and 21h for the master controller (IRQ[0:7]), and at A0h and A1h for the slave controller (IRQ[8:13]). These registers have multiple functions depending on the data written to them. Table 94 lists the different register possibilities for each address.

Table 94. PIC Registers

Port	Aliases	Register Name/Function	Default Value	Type
20h	24h, 28h, 2Ch, 30h, 34h, 38h, 3Ch	Master PIC ICW1 Init. Cmd Word 1 Register	Undefined	WO
		Master PIC OCW2 Op Ctrl Word 2 Register	001XXXXXb	WO
		Master PIC OCW3 Op Ctrl Word 3 Register	X01XXX10b	R/W
21h	25h, 29h, 2Dh, 31h, 35h, 39h, 3Dh	Master PIC ICW2 Init. Cmd Word 2 Register	Undefined	WO
		Master PIC ICW3 Init. Cmd Word 3 Register	Undefined	WO
		Master PIC ICW4 Init. Cmd Word 4 Register	01h	WO
		Master PIC OCW1 Op Ctrl Word 1 Register	00h	R/W
A0h	A4h, A8h, ACh, B0h, B4h, B8h, BCh	Slave PIC ICW1 Init. Cmd Word 1 Register	Undefined	WO
		Slave PIC OCW2 Op Ctrl Word 2 Register	001XXXXXb	WO
		Slave PIC OCW3 Op Ctrl Word 3 Register	X01XXX10b	R/W
A1h	A5h, A9h, ADh, B1h, B5h, B9h, BDh	Slave PIC ICW2 Init. Cmd Word 2 Register	Undefined	WO
		Slave PIC ICW3 Init. Cmd Word 3 Register	Undefined	WO
		Slave PIC ICW4 Init. Cmd Word 4 Register	01h	WO
		Slave PIC OCW1 Op Ctrl Word 1 Register	00h	R/W
4D0h	–	Master PIC Edge/Level Triggered Register	00h	R/W
4D1h	–	Slave PIC Edge/Level Triggered Register	00h	R/W

8.4.2 ICW1—Initialization Command Word 1 Register

Offset Address:	Master Controller–020h Slave Controller–0A0h	Attribute:	WO
		Size:	8 bit /controller
Default Value:	All bits undefined		

A write to Initialization Command Word 1 starts the interrupt controller initialization sequence, during which the following occurs:

1. The Interrupt Mask register is cleared.
2. IRQ7 input is assigned priority 7.
3. The slave mode address is set to 7.
4. Special Mask Mode is cleared and Status Read is set to IRR.

Once this write occurs, the controller expects writes to ICW2, ICW3, and ICW4 to complete the initialization sequence.

Bit	Description
7:5	**ICW/OCW select**—WO. These bits are MCS-85 specific, and not needed. 000 = Should be programmed to “000”
4	**ICW/OCW select**—WO. 1 = This bit must be a 1 to select ICW1 and enable the ICW2, ICW3, and ICW4 sequence.
3	**Edge/Level Bank Select (LTIM)**—WO. Disabled. Replaced by the edge/level triggered control registers (ELCR).
2	**ADI**—WO. 0 = Ignored for the 82801E C-ICH. Should be programmed to 0.
1	**Single or Cascade (SNGL)**—WO. 0 = Must be programmed to a 0 to indicate two controllers operating in cascade mode.
0	**ICW4 Write Required (IC4)**—WO. 1 = This bit must be programmed to a 1 to indicate that ICW4 needs to be programmed.

8.4.3 ICW2—Initialization Command Word 2 Register

Offset Address:	Master Controller–021h Slave Controller–0A1h	Attribute:	WO
		Size:	8 bit /controller
Default Value:	All bits undefined		

ICW2 is used to initialize the interrupt controller with the five most significant bits of the interrupt vector address. The value programmed for bits[7:3] is used by the processor to define the base address in the interrupt vector table for the interrupt routines associated with each IRQ on the controller. Typical ISA ICW2 values are 08h for the master controller and 70h for the slave controller.

Bit	Description
7:3	**Interrupt Vector Base Address**—WO. Bits [7:3] define the base address in the interrupt vector table for the interrupt routines associated with each interrupt request level input.
2:0	**Interrupt Request Level**—WO. When writing ICW2, these bits should all be 0. During an interrupt acknowledge cycle, these bits are programmed by the interrupt controller with the interrupt to be serviced. This is combined with bits [7:3] to form the interrupt vector driven onto the data bus during the second INTA# cycle. The code is a three bit binary code: **Code / Master Interrupt / Slave Interrupt** 000 IRQ0 IRQ8 001 IRQ1 IRQ9 010 IRQ2 IRQ10 011 IRQ3 IRQ11 100 IRQ4 IRQ12 101 IRQ5 IRQ13 110 IRQ6 IRQ14 111 IRQ7 IRQ15

8.4.4 ICW3—Master Controller Initialization Command Word 3 Register

Offset Address:	21h	Attribute:	WO
Default Value:	All bits undefined	Size:	8 bits

Bit	Description
7:3	0 = These bits must be programmed to zero.
2	**Cascaded Interrupt Controller IRQ Connection**—WO. This bit indicates that the slave controller is cascaded on IRQ2. When IRQ8#–IRQ15 is asserted, it goes through the slave controller's priority resolver. The slave controller's INTR output onto IRQ2. IRQ2 then goes through the master controller's priority solver. If it wins, the INTR signal is asserted to the processor, and the returning interrupt acknowledge returns the interrupt vector for the slave controller. 1 = This bit must always be programmed to a 1.
1:0	0 = These bits must be programmed to zero.

8.4.5 ICW3—Slave Controller Initialization Command Word 3 Register

Offset Address: A1h Attribute: WO
Default Value: All bits undefined Size: 8 bits

Bit	Description
7:3	0 = These bits must be programmed to zero.
2:0	**Slave Identification Code**—WO. These bits are compared against the slave identification code broadcast by the master controller from the trailing edge of the first internal INTA# pulse to the trailing edge of the second internal INTA# pulse. These bits must be programmed to 02h to match the code broadcast by the master controller. When 02h is broadcast by the master controller during the INTA# sequence, the slave controller assumes responsibility for broadcasting the interrupt vector.

8.4.6 ICW4—Initialization Command Word 4 Register

Offset Address: Master Controller–021h Attribute: WO
Slave Controller–0A1h Size: 8 bits

Bit	Description
7:5	0 = These bits must be programmed to zero.
4	**Special Fully Nested Mode (SFNM)**—WO. 0 = Should normally be disabled by writing a 0 to this bit. 1 = Special fully nested mode is programmed.
3	**Buffered Mode (BUF)**—WO. 0 = Must be programmed to 0 for the 82801E C-ICH. This is non-buffered mode.
2	**Master/Slave in Buffered Mode**—WO. Not used. 0 = Should always be programmed to 0.
1	**Automatic End of Interrupt (AEOI)**—WO. 0 = This bit should normally be programmed to 0. This is the normal end of interrupt. 1 = Automatic End of Interrupt (AEOI) mode is programmed. AEOI is discussed in Section 3.7.4.
0	**Microprocessor Mode**—WO. 1 = Must be programmed to 1 to indicate that the controller is operating in an Intel Architecture-based system.

8.4.7 OCW1—Operational Control Word 1 (Interrupt Mask) Register

Offset Address: Master Controller–021h Attribute: R/W
Slave Controller–0A1h Size: 8 bits
Default Value: 00h

Bit	Description
7:0	**Interrupt Request Mask**—R/W. When a 1 is written to any bit in this register, the corresponding IRQ line is masked. When a 0 is written to any bit in this register, the corresponding IRQ mask bit is cleared and interrupt requests will again be accepted by the controller. Masking IRQ2 on the master controller will also mask the interrupt requests from the slave controller.

8.4.8 OCW2—Operational Control Word 2 Register

Offset Address:	Master Controller–020h	Attribute:	WO
	Slave Controller–0A0h	Size:	8 bits
Default Value:	Bit[4:0]=undefined, Bit[7:5]=001		

Following a part reset or ICW initialization, the controller enters the fully nested mode of operation. Non-specific EOI without rotation is the default. Both rotation mode and specific EOI mode are disabled following initialization.

Bit	Description
7:5	**Rotate and EOI Codes (R, SL, EOI)**—WO. These three bits control the Rotate and End of Interrupt modes and combinations of the two. 000 = Rotate in Auto EOI Mode (Clear) 001 = Non-specific EOI command 010 = No Operation 011 = Specific EOI Command 100 = Rotate in Auto EOI Mode (Set) 101 = Rotate on Non-Specific EOI Command 110 = †Set Priority Command 111 = †Rotate on Specific EOI Command †L0–L2 Are Used
4:3	**OCW2 Select**—WO. When selecting OCW2, bits 4:3 = “00”
2:0	**Interrupt Level Select (L2, L1, L0)**—WO. L2, L1, and L0 determine the interrupt level acted upon when the SL bit is active. A simple binary code, outlined below, selects the channel for the command to act upon. When the SL bit is inactive, these bits do not have a defined function; programming L2, L1 and L0 to 0 is sufficient in this case. **Bits** **Interrupt Level** 000 IRQ0/8 001 IRQ1/9 010 IRQ2/10 011 IRQ3/11 100 IRQ4/12 101 IRQ5/13 110 IRQ6/14 111 IRQ7/15

8.4.9 OCW3—Operational Control Word 3 Register

Offset Address: Master Controller–020h Attribute: WO
Slave Controller–0A0h Size: 8 bits
Default Value: Bit[6,0]=0, Bit[7,4:2]=undefined, Bit[5,1]=1

Bit	Description
7	Reserved. Must be 0.
6	**Special Mask Mode (SMM)**—WO. 1 = The Special Mask Mode can be used by an interrupt service routine to dynamically alter the system priority structure while the routine is executing, through selective enabling/disabling of the other channel's mask bits. Bit 5, the ESMM bit, must be set for this bit to have any meaning.
5	**Enable Special Mask Mode (ESMM)**—WO. 0 = Disable. The SMM bit becomes a "don't care". 1 = Enable the SMM bit to set or reset the Special Mask Mode.
4:3	**OCW3 Select**—WO. When selecting OCW3, bits 4:3 = “01”
2	**Poll Mode Command**—WO. 0 = Disable. Poll Command is not issued. 1 = Enable. The next I/O read to the interrupt controller is treated as an interrupt acknowledge cycle. An encoded byte is driven onto the data bus, representing the highest priority level requesting service.
1:0	**Register Read Command**—WO. These bits provide control for reading the In-Service Register (ISR) and the Interrupt Request Register (IRR). When bit 1=0, bit 0 will not affect the register read selection. When bit 1=1, bit 0 selects the register status returned following an OCW3 read. If bit 0=0, the IRR will be read. If bit 0=1, the ISR will be read. Following ICW initialization, the default OCW3 port address read will be "read IRR". To retain the current selection (read ISR or read IRR), always write a 0 to bit 1 when programming this register. The selected register can be read repeatedly without reprogramming OCW3. To select a new status register, OCW3 must be reprogrammed prior to attempting the read. 00 = No Action 01 = No Action 10 = Read IRQ Register 11 = Read IS Register

8.4.10 ELCR1—Master Controller Edge/Level Triggered Register

Offset Address:	4D0h	Attribute:	R/W
Default Value:	00h	Size:	8 bits

In edge mode, (bit[x] = 0), the interrupt is recognized by a low to high transition. In level mode (bit[x] = 1), the interrupt is recognized by a high level. The cascade channel, IRQ2, the heartbeat timer (IRQ0), and the keyboard controller (IRQ1), cannot be put into level mode.

Bit	Description
7	**IRQ7 ECL**—R/W. 0 = Edge. 1 = Level.
6	**IRQ6 ECL**—R/W. 0 = Edge. 1 = Level.
5	**IRQ5 ECL**—R/W. 0 = Edge. 1 = Level.
4	**IRQ4 ECL**—R/W. 0 = Edge. 1 = Level.
3	**IRQ3 ECL**—R/W. 0 = Edge. 1 = Level.
2:0	Reserved. Must be 0.

8.4.11 ELCR2—Slave Controller Edge/Level Triggered Register

Offset Address:	4D1h	Attribute:	R/W
Default Value:	00h	Size:	8 bits

In edge mode (bit[x] = 0) the interrupt is recognized by a low-to-high transition. In level mode (bit[x] = 1) the interrupt is recognized by a high level. The real time clock interrupt (IRQ8#) and the floating point error interrupt (IRQ13) cannot be programmed for level mode.

Bit	Description
7	**IRQ15 ECL**—R/W. 0 = Edge. 1 = Level.
6	**IRQ14 ECL**—R/W. 0 = Edge. 1 = Level.
5	Reserved. Must be 0.
4	**IRQ12 ECL**—R/W. 0 = Edge. 1 = Level.
3	**IRQ11 ECL**—R/W. 0 = Edge. 1 = Level.
2	**IRQ10 ECL**—R/W. 0 = Edge. 1 = Level.
1	**IRQ9 ECL**—R/W. 0 = Edge. 1 = Level.
0	Reserved. Must be 0.

8.5 Advanced Interrupt Controller (APIC)

8.5.1 APIC Register Map

The APIC is accessed via an indirect addressing scheme. Two registers are visible by software for manipulation of most of the APIC registers. These registers are mapped into memory space. The registers are shown in Table 95.

Table 95. APIC Direct Registers

Address	Register	Size	Type
FEC0_0000h	Index Register	8 bits	R/W
FEC0_0010h	Data Register	32 bits	R/W
FECO_0020h	IRQ Pin Assertion Register	8 bits	WO
FECO_0040h	EOI Register	8 bits	WO

Table 96 lists the registers that can be accessed within the APIC via the Index Register. When accessing these registers, accesses must be done a DWord at a time. For example, software should never access byte 2 from the Data register before accessing bytes 0 and 1. The hardware will not attempt to recover from a bad programming model in this case.

Table 96. APIC Indirect Registers

Index	Register	Size	Type
00h	ID	32 bits	R/W
01h	Version	32 bits	RO
02h	Arbitration ID	32 bits	RO
03h	Boot Configuration	32 bits	R/W
04h–0Fh	Reserved		RO
10h –11h	Redirection Table 0	64 bits	R/W
12h–13h	Redirection Table 1	64 bits	R/W
...	...	...	...
3Eh–3Fh	Redirection Table 23	64 bits	R/W
40h–FFh	Reserved		RO

8.5.2 IND—Index Register

Memory Address FEC0_0000h Attribute: R/W
Default Value: 00h Size: 8 bits

The Index Register will select which APIC indirect register to be manipulated by software. The selector values for the indirect registers are listed in Table 96. Software programs this register to select the desired APIC internal register

Bit	Description
7:0	**APIC Index**—R/W. This is an 8 bit pointer into the I/O APIC register table.

8.5.3 DAT—Data Register

Memory Address	FEC0_0010h	Attribute:	R/W
Default Value:	00000000h	Size:	32 bits

This is a 32 bit register specifying the data to be read or written to the register pointed to by the Index register. This register can only be accessed in DWord quantities.

Bit	Description
7:0	**APIC Data**—R/W. This is a 32 bit register for the data to be read or written to the APIC indirect register pointed to by the Index register.

8.5.4 IRQPA—IRQ Pin Assertion Register

Memory Address	FEC0_0020h	Attribute:	WO
Default Value:	N/A	Size:	32 bits

The IRQ Pin Assertion Register is present to provide a mechanism to scale the number of interrupt inputs into the I/O APIC without increasing the number of dedicated input pins. When a device that supports this interrupt assertion protocol requires interrupt service, that device will issue a write to this register. Bits 4:0 written to this register contain the IRQ number for this interrupt. The only valid values are 0–23. Bits 31:5 are ignored. To provide for future expansion, peripherals should always write a value of 0 for Bits 31:5.

See Section 3.8.4 for more details on how PCI devices will use this field.

Note: Writes to this register are only allowed by the processor and by masters on the 82801E C-ICH's PCI bus. Writes by devices on PCI buses above the 82801E C-ICH (e.g., a PCI segment on a P64H) are not supported.

Bit	Description
31:5	Reserved. Bits 31:5 are ignored.
4:0	**IRQ Number**—WO. Bits 4:0 written to this register contain the IRQ number for this interrupt. The only valid values are 0–23.

8.5.5 EOIR—EOI Register

Memory Address	FEC0_0040h	Attribute:	WO
Default Value:	N/A	Size:	32 bits

The EOI register is present to provide a mechanism to maintain the level triggered semantics for level-triggered interrupts issued on the parallel bus.

When a write is issued to this register, the I/O APIC will check the lower 8 bits written to this register, and compare it with the vector field for each entry in the I/O Redirection Table. When a match is found, the Remote_IRR bit for that I/O Redirection Entry will be cleared.

Note: This is similar to what already occurs when the APIC sees the EIO message on the serial bus. Note that if multiple I/O Redirection entries, for any reason, assign the same vector for more than one interrupt input, each of those entries will have the Remote_IRR bit reset to 0. The interrupt which was prematurely reset will not be lost because if its input remained active when the Remote_IRR bit is cleared, the interrupt will be reissued and serviced at a later time. Note that only bits 7:0 are actually used. Bits 31:8 are ignored by the 82801E C-ICH.

Note: To provide for future expansion, the processor should always write a value of 0 to Bits 31:8.

Bit	Description
31:8	Reserved. To provide for future expansion, the processor should always write a value of 0 to Bits 31:8.
7:0	**Redirection Entry Clear**—WO. When a write is issued to this register, the I/O APIC will check this field, and compare it with the vector field for each entry in the I/O Redirection Table. When a match is found, the Remote_IRR bit for that I/O Redirection Entry will be cleared.

8.5.6 ID—Identification Register

Index Offset:	00h	Attribute:	R/W
Default Value:	00000000h	Size:	32 bits

The APIC ID serves as a physical name of the APIC. The APIC bus arbitration ID for the APIC is derived from its I/O APIC ID. This register is reset to zero on power up reset.

Bit	Description
31:28	Reserved.
27:24	**APIC ID**—R/W. Software must program this value before using the APIC.
23:0	Reserved.

8.5.7 VER—Version Register

Index Offset: 01h Attribute: RO
Default Value: 00170002h Size: 32 bits

Each I/O APIC contains a hardwired Version Register that identifies different implementation of APIC and their versions. The maximum redirection entry information also is in this register, to let software know how many interrupt are supported by this APIC.

Bit	Description
31:24	Reserved.
23:16	**Maximum Redirection Entries**—RO. This is the entry number (0 being the lowest entry) of the highest entry in the redirection table. It is equal to the number of interrupt input pins minus one and is in the range 0 through 239. In the 82801E C-ICH this field is hardwired to 17h to indicate 24 interrupts.
15	**PRQ**—RO. This bit is set to 1 to indicate that this version of the I/O APIC implements the IRQ Assertion register and allows PCI devices to write to it to cause interrupts.
14:8	Reserved.
7:0	**Version**—RO. This is a version number that identifies the implementation version.

8.5.8 ARBID—Arbitration ID Register

Index Offset: 02h Attribute: RO
Default Value: 00000000h Size: 32 bits

This register contains the bus arbitration priority for the APIC. If the APIC clock is running, the ARBID register is loaded whenever the APIC ID register is loaded. A rotating priority scheme is used for APIC bus arbitration. The winner of the arbitration becomes the lowest priority agent and assumes an arbitration ID of 0.

Bit	Description
31:28	Reserved.
27:24	**I/O APIC Identification**—RO. This 4 bit field contains the I/O APIC Arbitration ID.
23:0	Reserved.

8.5.9 BOOT_CONFIG—Boot Configuration Register

Index Offset: 03h Attribute: R/W
Default Value: 00000000h Size: 32 bits

This register is used to control the interrupt delivery mechanism for the APIC.

Bit	Description
31:1	Reserved.
0	**Delivery Type (DT)**—R/W. 0 = Interrupt delivery mechanism is via the APIC serial bus (default). 1 = Interrupt delivery mechanism is a processor-side bus message.

8.5.10 Redirection Table

Index Offset:	10h–11h (vector 0) through 3E–3Fh (vector 23)	Attribute:	R/W
Default Value:	Bit 16–1, Bits[15:12]=0. All other bits undefined	Size:	64 bits each, (accessed as two 32 bit quantities)

The Redirection Table has a dedicated entry for each interrupt input pin. The information in the Redirection Table is used to translate the interrupt manifestation on the corresponding interrupt pin into an APIC message.

The APIC will respond to an edge-triggered interrupt as long as the interrupt is held until after the acknowledge cycle has begun. Once the interrupt is detected, a delivery status bit internally to the I/O APIC is set. The state machine will step ahead and wait for an acknowledgment from the APIC bus unit that the interrupt message was sent over the APIC bus. Only then will the I/O APIC be able to recognize a new edge on that interrupt pin. That new edge will only result in a new invocation of the handler if its acceptance by the destination APIC causes the Interrupt Request Register bit to go from 0 to 1. (In other words, if the interrupt was not already pending at the destination.)

Bit	Description
63:56	**Destination**—R/W. If bit 11 of this entry is 0 [Physical], then bits [59:56] specifies an APIC ID. If bit 11 of this entry is 1 [Logical], then bits [63:56] specify the logical destination address of a set of processors.
55:17	Reserved.
16	**Mask**—R/W. 0 = Not masked: An edge or level on this interrupt pin results in the delivery of the interrupt to the destination. 1 = Masked: Interrupts are not delivered nor held pending. Setting this bit after the interrupt is accepted by a local APIC has no effect on that interrupt. This behavior is identical to the device withdrawing the interrupt before it is posted to the processor. It is software's responsibility to deal with the case where the mask bit is set after the interrupt message has been accepted by a local APIC unit but before the interrupt is dispensed to the processor.
15	**Trigger Mode**—R/W. This field indicates the type of signal on the interrupt pin that triggers an interrupt. 0 = Edge triggered. 1 = Level triggered.
14	**Remote IRR**—R/W. This bit is used for level triggered interrupts; its meaning is undefined for edge triggered interrupts. 0 = Reset when an EOI message is received from a local APIC. 1 = Set when Local APIC/s accept the level interrupt sent by the I/O APIC.
13	**Interrupt Input Pin Polarity**—R/W. This bit specifies the polarity of each interrupt signal connected to the interrupt pins. 0 = Active high. 1 = Active low.
12	**Delivery Status**—RO. This field contains the current status of the delivery of this interrupt. Writes to this bit have no effect. 0 = Idle. No activity for this interrupt. 1 = Pending. Interrupt has been injected, but delivery is held up due to the APIC bus being busy or the inability of the receiving APIC unit to accept the interrupt at this time.

Bit	Description
11	**Destination Mode**—R/W. This field determines the interpretation of the Destination field. 0 = Physical. Destination APIC ID is identified by bits [59:56]. 1 = Logical. Destinations are identified by matching bit [63:56] with the Logical Destination in the Destination Format Register and Logical Destination Register in each Local APIC.
10:8	**Delivery Mode**—R/W. This field specifies how the APICs listed in the destination field should act upon reception of this signal. Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode. These encodings are: 000 = Fixed. Deliver the signal on the INTR signal of all processor cores listed in the destination. Trigger Mode can be edge or level. 001 = Lowest Priority. Deliver the signal on the INTR signal of the processor core that is executing at the lowest priority among all the processors listed in the specified destination. Trigger Mode can be edge or level. 010 = SMI (System Management Interrupt). Requires the interrupt to be programmed as edge triggered. The vector information is ignored but must be programmed to all zeroes for future compatibility. 011 = Reserved 100 = NMI. Deliver the signal on the NMI signal of all processor cores listed in the destination. Vector information is ignored. NMI is treated as an edge triggered interrupt even if it is programmed as level triggered. For proper operation this redirection table entry must be programmed to edge triggered. The NMI delivery mode does not set the RIRR bit. Once the interrupt is detected, it will be sent over the APIC bus. If the redirection table is incorrectly set to level, the loop count will continue counting through the redirection table addresses. Once the count for the NMI pin is reached again, the interrupt will be sent over the APIC bus again. 101 = INIT. Deliver the signal to all processor cores listed in the destination by asserting the INIT signal. All addressed local APICs will assume their INIT state. INIT is always treated as an edge triggered interrupt even if programmed as level triggered. For proper operation this redirection table entry must be programmed to edge triggered. The INIT delivery mode does not set the RIRR bit. Once the interrupt is detected, it will be sent over the APIC bus. If the redirection table is incorrectly set to level, the loop count will continue counting through the redirection table addresses. Once the count for the INIT pin is reached again, the interrupt will be sent over the APIC bus again 110 = Reserved 111 = ExtINT. Deliver the signal to the INTR signal of all processor cores listed in the destination as an interrupt that originated in an externally connected 8259A compatible interrupt controller. The INTA cycle that corresponds to this ExtINT delivery will be routed to the external controller that is expected to supply the vector. Requires the interrupt to be programmed as edge triggered.
7:0	**Vector**—R/W. This field contains the interrupt vector for this interrupt. Values range between 10h and FEh.

8.6 Real Time Clock Registers

8.6.1 I/O Register Address Map

The RTC internal registers and RAM are organized as two banks of 128 bytes each, called the standard and extended banks. The first 14 bytes of the standard bank contain the RTC time and date information along with four registers, A–D, that are used for configuration of the RTC. The extended bank contains a full 128 bytes of battery backed SRAM and will be accessible even when the RTC module is disabled (via the RTC configuration register). Registers A–D do not physically exist in the RAM.

All data movement between the host processor and the real-time clock is done through registers mapped to the standard I/O space. The register map appears in Table 97.

Table 97. RTC I/O Registers

I/O Locations	If U128E bit = 0	Function
70h and 74h	Also alias to 72h and 76h	Real-Time Clock (Standard RAM) Index Register
71h and 75h	Also alias to 73h and 77h	Real-Time Clock (Standard RAM) Target Register
72h and 76h		Extended RAM Index Register (if enabled)
73h and 77h		Extended RAM Target Register (if enabled)

NOTES:

1. I/O locations 70h and 71h are the standard ISA location for the real-time clock. The map for this bank is shown in Table 98. Locations 72h and 73h are for accessing the extended RAM. The extended RAM bank is also accessed using an indexed scheme. I/O address 72h is used as the address pointer and I/O address 73h is used as the data register. Index addresses above 127h are not valid. If the extended RAM is not needed, it may be disabled.
2. Software must preserve the value of bit 7 at I/O addresses 70h and 74h. When writing to these addresses, software must first read the value, and then write the same value for bit 7 during the sequential address write.

8.6.2 Indexed Registers

The RTC contains two sets of indexed registers that are accessed using the two separate Index and Target registers (70h/71h or 72h/73h). These are shown in Table 98.

Table 98. RTC (Standard) RAM Bank

Index	Name
00h	Seconds
01h	Seconds Alarm.
02h	Minutes
03h	Minutes Alarm
04h	Hours
05h	Hours Alarm
06h	Day of Week
07h	Day of Month

Index	Name
08h	Month
09h	Year
0Ah	Register A
0Bh	Register B
0Ch	Register C
0Dh	Register D
0Eh–7Fh	114 Bytes of User RAM

8.6.2.1 RTC_REGA—Register A

RTC Index:	0A	Attribute:	R/W
Default Value:	Undefined	Size:	8-bit
Lockable:	No	Power Well:	RTC

This register is used for general configuration of the RTC functions. None of the bits are affected by RSMRST# or any other 82801E C-ICH reset signal.

Bit	Description
7	**Update In Progress (UIP)**—R/W. This bit may be monitored as a status flag. 0 = The update cycle will not start for at least 492 µs. The time, calendar, and alarm information in RAM is always available when the UIP bit is 0. 1 = The update is soon to occur or is in progress.
6:4	**Division Chain Select (DV[2:0])**—R/W. These three bits control the divider chain for the oscillator, and are not affected by RSMRST# or any other reset signal. DV[2] corresponds to bit 6. 010 = Normal Operation 11X = Divider Reset 101 = Bypass 15 stages (test mode only) 100 = Bypass 10 stages (test mode only) 011 = Bypass 5 stages (test mode only) 001 = Invalid 000 = Invalid
3:0	**RS[3:0] Rate Select**—R/W. Selects one of 13 taps of the 15 stage divider chain. The selected tap can generate a periodic interrupt if the PIE bit is set in Register B. Otherwise this tap will set the PF flag of Register C. If the periodic interrupt is not to be used, these bits should all be set to zero. RS3 corresponds to bit 3. 0000 = Interrupt never toggles 0001 = 3.90625 ms 0010 = 7.8125 ms 0011 = 122.070 µs 0100 = 244.141 µs 0101 = 488.281 µs 0110 = 976.5625 µs 0111 = 1.953125 ms 1000 = 3.90625 ms 1001 = 7.8125 ms 1010 = 15.625 ms 1011 = 31.25 ms 1100 = 62.5 ms 1101 = 125 ms 1110 = 250 ms 1111 = 500 ms

8.6.2.2 RTC_REGB—Register B (General Configuration)

RTC Index:	0Bh	Attribute:	R/W
Default Value:	U0U00UUU (U: Undefined)	Size:	8-bit
Lockable:	No	Power Well:	RTC

Bit	Description
7	**Update Cycle Inhibit (SET)**—R/W. Enables/Inhibits the update cycles. This bit is not affected by RSMRST# nor any other reset signal. 0 = Update cycle occurs normally once each second. 1 = A current update cycle will abort and subsequent update cycles will not occur until SET is returned to zero. When set is one, the BIOS may initialize time and calendar bytes safely.
6	**Periodic Interrupt Enable (PIE)**—R/W. This bit is cleared by RSMRST#, but not on any other reset. 0 = Disable. 1 = Allows an interrupt to occur with a time base set with the RS bits of register A.
5	**Alarm Interrupt Enable (AIE)**—R/W. This bit is cleared by RSMRST#, but not on any other reset. 0 = Disable. 1 = Allows an interrupt to occur when the AF is set by an alarm match from the update cycle. An alarm can occur once a second, one an hour, once a day, or one a month.
4	**Update-ended Interrupt Enable (UIE)**—R/W. This bit is cleared by RSMRST#, but not on any other reset. 0 = Disable. 1 = Allows an interrupt to occur when the update cycle ends.
3	**Square Wave Enable (SQWE)**—R/W. This bit serves no function in the 82801E C-ICH. It is left in this register bank to provide compatibility with the Motorola* 146818B. The 82801E C-ICH has no SQW pin. This bit is cleared by RSMRST#, but not on any other reset.
2	**Data Mode (DM)**—R/W. Specifies either binary or BCD data representation. This bit is not affected by RSMRST# nor any other reset signal. 0 = BCD 1 = Binary
1	**Hour Format (HOURFORM)**—R/W. Indicates the hour byte format. This bit is not affected by RSMRST# nor any other reset signal. 0 = Twelve-hour mode. In twelve hour mode, the seventh bit represents AM as zero and PM as one. 1 = Twenty-four hour mode.
0	**Daylight Savings Enable (DSE)**—R/W. Triggers two special hour updates per year. The days for the hour adjustment are those specified in United States federal law as of 1987, which is different than previous years. This bit is not affected by RSMRST# nor any other reset signal. 0 = Daylight Savings Time updates do not occur. 1 = a) Update on the first Sunday in April, where time increments from 1:59:59 AM to 3:00:00 AM. b) Update on the last Sunday in October when the time first reaches 1:59:59 AM, it is changed to 1:00:00 AM. The time must increment normally for at least two update cycles (seconds) previous to these conditions for the time change to occur properly.

8.6.2.3 RTC_REGC—Register C (Flag Register)

RTC Index:	0Ch	Attribute:	RO
Default Value:	00U00000 (U: Undefined)	Size:	8-bit
Lockable:	No	Power Well:	RTC

Writes to Register C have no effect.

Bit	Description
7	**Interrupt Request Flag (IRQF)**—RO. IRQF = (PF * PIE) + (AF * AIE) + (UF *UFE). This also causes the CH_IRQ_B signal to be asserted. This bit is cleared upon RSMRST# or a read of Register C.
6	**Periodic Interrupt Flag (PF)**—RO. This bit is cleared upon RSMRST# or a read of Register C. 0 = If no taps are specified via the RS bits in Register A, this flag will not be set. 1 = Periodic interrupt Flag will be 1 when the tap specified by the RS bits of register A is 1.
5	**Alarm Flag (AF)**—RO. 0 = This bit is cleared upon RTCRST# or a read of Register C. 1 = Alarm Flag will be set after all Alarm values match the current time.
4	**Update-ended Flag (UF)**—RO. 0 = The bit is cleared upon RSMRST# or a read of Register C. 1 = Set immediately following an update cycle for each second.
3:0	Reserved. Will always report 0.

8.6.2.4 RTC_REGD—Register D (Flag Register)

RTC Index:	0Dh	Attribute:	R/W
Default Value:	10UUUUUU (U: Undefined)	Size:	8-bit
Lockable:	No	Power Well:	RTC

Bit	Description
7	**Valid RAM and Time Bit (VRT)**—R/W. 0 = This bit should always be written as a 0 for write cycle; however, it will return a 1 for read cycles. 1 = The Valid Ram and Time bit is set to 1 when the PWRGD (power good) signal provided is high. This feature is not typically used.
6	Reserved. This bit always returns a 0 and should be set to 0 for write cycles.
5:0	**Date Alarm**—R/W. These bits store the date of month alarm value. If set to 000000b, then a don't care state is assumed. The host must configure the date alarm for these bits to do anything, yet they can be written at any time. If the date alarm is not enabled, these bits will return zeros to mimic the functionality of the Motorola* 146818B. These bits are not affected by RESET.

8.7 Processor Interface Registers

8.7.1 NMI_SC—NMI Status and Control Register

I/O Address:	61h	Attribute:	R/W (some bits RO)
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7	**SERR# NMI Source Status (SERR#_NMI_STS)**—RO. 1 = PCI agent detected a system error and pulses the PCI SERR# line. This interrupt source is enabled by setting bit 2 to 0. To reset the interrupt, set bit 2 to 1 and then set it to 0. When writing to port 61h, this bit must be 0.
6	**IOCHK# NMI Source Status (IOCHK_NMI_STS)**—RO. 1 = An ISA agent (via SERIRQ) asserted IOCHK# on the ISA bus. This interrupt source is enabled by setting bit 3 to 0. To reset the interrupt, set bit 3 to 0 and then set it to 1. When writing to port 61h, this bit must be a 0.
5	**Timer Counter 2 OUT Status (TMR2_OUT_STS)**—RO. This bit reflects the current state of the 8254 counter 2 output. Counter 2 must be programmed following any PCI reset for this bit to have a determinate value. When writing to port 61h, this bit must be a 0.
4	**Refresh Cycle Toggle (REF_TOGGLE)**—RO. This signal toggles from either 0 to 1 or 1 to 0 at a rate that is equivalent to when refresh cycles would occur. When writing to port 61h, this bit must be a 0.
3	**IOCHK# NMI Enable (IOCHK_NMI_EN)**—R/W. 0 = Enabled. 1 = Disabled and cleared.
2	**PCI SERR# Enable (PCI_SERR_EN)**—R/W. 0 = SERR# NMIs are enabled. 1 = SERR# NMIs are disabled and cleared.
1	**Speaker Data Enable (SPKR_DAT_EN)**—R/W. 0 = SPKR output is a 0. 1 = SPKR output is equivalent to the Counter 2 OUT signal value.
0	**Timer Counter 2 Enable (TIM_CNT2_EN)**—R/W. 0 = Disable. 1 = Enable

8.7.2 NMI_EN—NMI Enable (and Real Time Clock Index)

I/O Address:	70h	Attribute:	R/W (Special)
Default Value:	80h	Size:	8-bit
Lockable:	No	Power Well:	Core

Note: The RTC Index field is write-only for normal operation. This field can only be read in Alt-Access Mode. Note, however, that this register is aliased to Port 74h (documented in Table 97), and all bits are readable at that address.

Bits	Description
7	**NMI Enable (NMI_EN)**—R/W. 0 = Enable NMI sources. 1 = Disable All NMI sources.
6:0	**Real Time Clock Index Address (RTC_INDX)**—R/W. This data goes to the RTC to select which register or CMOS RAM address is being accessed.

8.7.3 PORT92—Fast A20 and Init Register

I/O Address:	92h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:2	Reserved.
1	**Alternate A20 Gate (ALT_A20_GATE)**—R/W. This bit is ORed with the A20GATE input signal to generate A20M# to the processor. 0 = A20M# signal can potentially go active. 1 = This bit is set when INIT# goes active.
0	**Interrupt Now (INIT_NOW)**—R/W. When this bit transitions from a 0 to a 1, the 82801E C-ICH will force INIT# active for 16 PCI clocks.

8.7.4 COPROC_ERR—Coprocessor Error Register

I/O Address:	F0h	Attribute:	WO
Default Value:	00h	Size:	8-bits
Lockable:	No	Power Well:	Core

Bits	Description
7:0	**Coprocessor Error (COPROC_ERR)**—WO. Any value written to this register will cause IGNNE# to go active, if FERR# had generated an internal IRQ13. For FERR# to generate an internal IRQ13, the COPROC_ERR_EN bit (Device 31:Function 0, Offset D0, Bit 13) must be 1.

8.7.5 RST_CNT—Reset Control Register

I/O Address:	CF9h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:4	Reserved.
3	**Full Reset (FULL_RST)**—R/W. When this bit is set to 1 and SYS_RST (bit 1) is set to 1 (indicating hard reset, not soft reset) and the RST_CPU bit (bit 2) is written from 0 to 1, the 82801E C-ICH will do a full reset.
2	**Reset Processor (RST_CPU)**—R/W. When this bit transitions from a 0 to a 1, it initiates a hard or soft reset, as determined by the SYS_RST bit (bit 1 of this register).
1	**System Reset (SYS_RST)**—R/W. This bit is used to determine a hard or soft reset to the processor. 1 = When RST_CPU bit goes from 0 to 1, the 82801E C-ICH performs a hard reset by activating PCIRST# for 1 millisecond. It also resets the resume well bits (except for those noted throughout the datasheet). 0 = When RST_CPU bit goes from 0 to 1, the 82801E C-ICH performs a soft reset by activating INIT# for 16 PCI clocks.
0	Reserved.

8.8 Power Management Registers (D31:F0)

Warning: **This section is not to be considered as part of the 82801E C-ICH behavioral specification. It is intended to identify key areas and steps taken by the BIOS to make sure there are no conflicts with the functional definition. These registers are from legacy power management support and are not supported in the 82801E C-ICH.**

The power management registers are distributed within the PCI Device 31: Function 0 space, as well as a separate I/O range. Each register is described below. Unless otherwise indicate, bits are in the main (core) power well.

Bits not explicitly defined in each register are assumed to be reserved. When writing to a reserved or not implemented bit, the value should always be 0. Software should not attempt to use the value read from a reserved bit, as it may not be consistently 1 or 0.

8.8.1 Power Management PCI Configuration Registers (D31:F0)

Table 99 shows a small part of the configuration space for PCI Device 31: Function 0. It includes only those registers dedicated for power management. Some of the registers are only used for Legacy Power management schemes.

Table 99. PCI Configuration Map (PM—D31:F0)

Offset	Mnemonic	Register Name/Function	Default	Type
40h–43h	ACPI_BASE	ACPI Base Address	00000001h	R/W
44h	ACPI_CNTL	ACPI Control	00h	R/W
A0h	GEN_PMCON_1	General Power Management Configuration 1	0000h	R/W
A2h	GEN_PMCON_2	General Power Management Configuration 2	0000h	R/W
A4h	GEN_PMCON_3	General Power Management Configuration 3	00h	R/W
B8–BBh	GPI_ROUT	GPI Route Control	00000000h	R/W
C0	TRP_FWD_EN	I/O Monitor Trap Forwarding Enable		
C4–CAh	MON[*n*]_TRP_RNG	I/O Monitor[4:7] Trap Range	0000h	R/W
CCh	MON_TRP_MSK	I/O Monitor Trap Range Mask	0000h	R/W

8.8.1.1 GEN_PMCON_1—General PM Configuration 1 Register (PM—D31:F0)

Offset Address:	A0h	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI, Legacy
		Power Well:	Core

Bit	Description
15:11	Reserved. Must be 00.
10	**Software SMI Rate Select (SWSMI_RATE_SEL)**—R/W. 0 = SWSMI Timer will time out in 64 ms ± 4 ms (default). 1 = SWSMI Timer will time out in 1.5 ms ± 0.5 ms.
9	**PWRBTN#**. Not implemented.
8:7	Reserved.
6	**IA64_EN**. not supported by 82801E C-ICH. This bit must be set to 0.
5	**CPU SLP# Enable (CPUSLP_EN)**—R/W. Must be set to 0.
4:2	Reserved.
1:0	**Periodic SMI# rate Select (PER_SMI_SEL)**—R/W. Set by software to control the rate at which periodic SMI# is generated. 00 = 1 minute 01 = 32 seconds 10 = 16 seconds 11 = 8 seconds

8.8.1.2 GEN_PMCON_2—General PM Configuration 2 Register (PM—D31:F0)

Offset Address:	A2h	Attribute:	R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI, Legacy
		Power Well:	Resume

Bit	Description
7:2	Reserved.
1	**CPU Power Failure (CPUPWR_FLR)**—R/WC. 0 = Software clears this bit by writing a 1 to the bit position. 1 = Indicates that the VRMPWRGD signal from the processor's VRM went low.
0	**PWROK Failure (PWROK_FLR)**—R/WC. 0 = Software clears this bit by writing a 1 to the bit position,. 1 = This bit will be set any time PWROK goes low. The bit will be cleared only by software by writing a 1 to this bit. **NOTE:** Traditional designs have a reset button logically ANDed with the PWROK signal from the power supply and the processor's voltage regulator module. If this is done with the 82801E C-ICH, the PWROK_FLR bit will be set. The 82801E C-ICH treats this internally as if the RSMRST# signal had gone active. However, it is not treated as a full power failure. If PWROK goes inactive and then active (but RSMRST# stays high), then the 82801E C-ICH will reboot (regardless of the state of the AFTERG3 bit). If the RSMRST# signal also goes low before PWROK goes high, then this is a full power failure and the reboot policy is controlled by the AFTERG3 bit.

8.8.1.3 GEN_PMCON_3—General PM Configuration 3 Register (PM—D31:F0)

Offset Address:	A4h	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Usage:	ACPI, Legacy
		Power Well:	RTC

Bit	Description
7:3	Reserved.
2	**RTC Power Status (RTC_PWR_STS)**—R/WC. 0 = Software clears this bit by writing a 1 to the bit position. 1 = Indicates that the RTC battery was removed or failed. This bit is set when RTCRST# signal is low. **NOTE:** Clearing CMOS in an ICH-based platform can be done by using a jumper on RTCRST# or GPI, or using SAFEMODE strap. Implementations should not attempt to clear CMOS by using a jumper to pull VccRTC low.
1	**Power Failure (PWR_FLR)**—R/WC. This bit is in the RTC well and is not cleared by any type of reset except RTCRST#. 0 = Indicates that the trickle current has not failed since the last time the bit was cleared. Software clears this bit by writing a 1 to the bit position. 1 = Indicates that the trickle current (from the main battery or trickle supply) was removed or failed. **NOTE:** Clearing CMOS in an ICH-based platform can be done by using a jumper on RTCRST# or GPI, or using SAFEMODE strap. Implementations should not attempt to clear CMOS by using a jumper to pull VccRTC low.
0	**After G3 State Select (AFTERG3_EN)**—R/W. Determines what state to go to when power is re-applied after a power failure (G3 state). Must be 0. 0 = System will return to S0 state (boot) after power is re-applied.

8.8.1.4 GPI_ROUT—GPI Routing Control Register (PM—D31:F0)

Offset Address:	B8h–BBh	Attribute:	R/W
Default Value:	0000h	Size:	32-bit
Lockable:	No	Power Well:	Resume

Bit	Description
31:30	**GPI[15] Route**—R/W. See bits 1:0 for description.
Same pattern for GPI[14] through GPI[3]	
5:4	**GPI[2] Route**—R/W. See bits 1:0 for description.
3:2	**GPI[1] Route**—R/W. See bits 1:0 for description.
1:0	**GPI[0] Route**—R/W. GPIO[13:11,8:6,4:3,1:0] can be routed to cause an SMI or SCI when the GPI[n]_STS bit is set. If the GPIO is not set to an input, this field has no effect. 00 = No effect. 01 = SMI# (if corresponding GPE1_EN bit is also set) 10 = SCI (if corresponding GPE1_EN bit is also set) 11 = Reserved

Note: GPIOs that are not implemented will not have the corresponding bits implemented in this register.

8.8.1.5 TRP_FWD_EN—IO Monitor Trap Forwarding Enable Register (PM—D31:F0)

Offset Address:	C0h	Attribute:	R/W (Special)
Default Value:	00h	Size:	8 bits
Lockable:	No	Usage:	Legacy Only
Power Well:	Core		

The 82801E C-ICH uses this register to enable the monitors to forward cycles to LPC, independent of the POS_DEC_EN bit and the bits that enable the monitor to generate an SMI#. The only criteria is that the address passes the decoding logic as determined by the MON[*n*]_TRP_RNG and MON_TRP_MSK register settings.

Bit	Description
7	**Monitor 7 Forward Enable (MON7_FWD_EN)**—R/W. 0 = Disable. Cycles trapped by I/O Monitor 7 will not be forwarded to LPC. 1 = Enable. Cycles trapped by I/O Monitor 7 will be forwarded to LPC.
6	**Monitor 6 Forward Enable (MON6_FWD_EN)**—R/W. 0 = Disable. Cycles trapped by I/O Monitor 6 will not be forwarded to LPC. 1 = Enable. Cycles trapped by I/O Monitor 6 will be forwarded to LPC.
5	**Monitor 5 Forward Enable (MON5_FWD_EN)**—R/W. 0 = Disable. Cycles trapped by I/O Monitor 5 will not be forwarded to LPC. 1 = Enable. Cycles trapped by I/O Monitor 5 will be forwarded to LPC.
4	**Monitor 4 Forward Enable (MON4_FWD_EN)**—R/W. 0 = Disable. Cycles trapped by I/O Monitor 4 will not be forwarded to LPC. 1 = Enable. Cycles trapped by I/O Monitor 4 will be forwarded to LPC.
3:0	Reserved.

8.8.1.6 MON[*n*]_TRP_RNG—I/O Monitor [4:7] Trap Range Register for Devices 4–7 (PM—D31:F0)

Offset Address:	C4h, C6h, C8h, CAh	Attribute:	R/W
Default Value:	00h	Size:	16 bits
Lockable:	No	Usage:	Legacy Only
Power Well:	Core		

These registers set the ranges that Device Monitors 4–7 should trap. Offset 4Ch corresponds to Monitor 4. Offset C6h corresponds to Monitor 5, etc.

If the trap is enabled in the MON_SMI register and the address is in the trap range (and passes the mask set in the MON_TRP_MSK register) the 82801E C-ICH generates an SMI#. This SMI# occurs if the address is positively decoded by another device on PCI or by the 82801E C-ICH (because it would be forwarded to LPC or some other 82801E C-ICH internal registers). The trap ranges should not point to registers in the 82801E C-ICH's internal IDE, USB, or LAN I/O space. If the cycle is to be claimed by the 82801E C-ICH and targets one of the permitted 82801E C-ICH internal registers (interrupt controller, RTC, etc.), the cycle will complete to the intended target and an SMI# will be generated (this is the same functionality as the ICH component). If the cycle is to be claimed by the 82801E C-ICH and the intended target is on LPC, an SMI# will be generated but the cycle will only be forwarded to the intended target if forwarding to LPC is enabled via the TRP_FWD_EN register settings.

Bit	Description
15:0	**Monitor Trap Base Address (MON[*n*]_TRAP_BASE)**—R/W. Base I/O locations that MON[*n*] traps (where *n* = 4, 5, 6 or 7). The range can be mapped anywhere in the processor I/O space (0–64 Kbyte). Any access to the range will generate an SMI# if enabled by the associated DEV[*n*]_TRAP_EN bit in the MON_SMI register (PMBASE +40h).

8.8.1.7 MON_TRP_MSK—I/O Monitor Trap Range Mask Register for Devices 4–7 (PM—D31:F0)

Offset Address:	CCh	Attribute:	R/W
Default Value:	00h	Size:	16 bits
Lockable:	No	Usage:	Legacy Only
Power Well:	Core		

Bit	Description
15:12	**Monitor 7 Forward Mask (MON7_MASK)**—R/W. Selects low 4-bit mask for the I/O locations that MON7 will trap. Similar to MON4_MASK.
11:8	**Monitor 6 Forward Mask (MON6_MASK**)—R/W. Selects low 4-bit mask for the I/O locations that MON6 will trap. Similar to MON4_MASK.
7:4	**Monitor 5 Forward Mask (MON5_MASK)**—R/W. Selects low 4-bit mask for the I/O locations that MON5 will trap. Similar to MON4_MASK.
3:0	**Monitor 4 Forward Mask (MON4_MASK)**—R/W. Selects low 4-bit mask for the I/O locations that MON7 will trap. When a mask bit is set to a 1, the corresponding bit in the base I/O selection will not be decoded. For example, if MON4_TRAP_BASE = 1230h, and MON4_MSK = 0011b, the 82801E C-ICH will decode 1230h, 1231h, 1232h, and 1233h for Monitor 4.

8.8.2 APM I/O Decode

Warning: **This section is not to be considered as part of the 82801E C-ICH behavioral specification. It is intended to identify key areas and steps taken by the BIOS to make sure there are no conflicts with the functional definition. These registers are from legacy power management support and are not supported in the 82801E C-ICH.**

Table 100 shows the I/O registers associated with APM support. This register space is enabled in the PCI Device 31: Function 0 space (APMDEC_EN), and cannot be moved (fixed I/O location).

Table 100. APM Register Map

Address	Mnemonic	Register Name/Function	Default	Type
B2h	APM_CNT	Advanced Power Management Control Port	00h	R/W
B3h	APM_STS	Advanced Power Management Status Port	00h	R/W

8.8.2.1 APM_CNT—Advanced Power Management Control Port Register

I/O Address: B2h
Default Value: 00h
Lockable: No
Power Well: Core
Attribute: R/W
Size: 8-bit
Usage: Legacy Only

Bit	Description
7:0	Used to pass an APM command between the OS and the SMI handler. Writes to this port not only store data in the APMC register but also generate an SMI# when the APMC_EN bit is set.

8.8.2.2 APM_STS—Advanced Power Management Status Port Register

I/O Address: B3h
Default Value: 00h
Lockable: No
Power Well: Core
Attribute: R/W
Size: 8-bit
Usage: Legacy Only

Bit	Description
7:0	Used to pass data between the OS and the SMI handler. Basically, this is a scratchpad register and is not affected by any other register or function (other than a PCI reset).

8.8.3 Power Management I/O Registers

Note: **These registers are from legacy power management support and are not supported in the 82801E C-ICH.**

Table 101 shows the registers associated with ACPI and Legacy power management support. These registers are enabled in the PCI Device 31: Function 0 space (PM_IO_EN), and can be moved to any I/O location (128-byte aligned). The registers are defined to be compliant with the ACPI 1.0 specification, and use the same bit names.

Note: All reserved bits and registers will always return 0 when read, and will have no effect when written.

Warning: This section is not to be considered as part of the 82801E C-ICH behavioral specification. It is intended to identify key areas and steps taken by the BIOS to make sure there are no conflicts with the functional definition.

Table 101. ACPI and Legacy I/O Register Map

PMBASE+ Offset	Register Name	ACPI Pointer	Default	Attributes
00–01h	PM1 Status	PM1a_EVT_BLK	0000h	R/W
02–03h	PM1 Enable	PM1a_EVT_BLK+2	0000h	R/W
04–07h	PM1 Control	PM1a_CNT_BLK	00000000h	R/W
08–0Bh	PM1 Timer	PMTMR_BLK	00000000h	RO
0Ch-0Fh	Reserved	—	—	—
10h–13h	Processor Control	P_BLK	00000000h	R/W
14h	Reserved	—	—	—
15h–20h	Reserved	—	—	—
28–29h	General Purpose Event 0 Status	GPE0_BLK	0000h	R/W
2A–2Bh	General Purpose Event 0 Enables	GPE0_BLK+2	0000h	R/W
2C–2D	General Purpose Event 1 Status	GPE1_BLK	0000h	R/W
2E–2F	General Purpose Event 1 Enables	GPE1_BLK+2	0000h	R/W
30–31h	SMI# Control and Enable	—	0000h	R/W
34–35h	SMI Status Register	—	0000h	R/W
36–3Fh	Reserved	—	0000h	RO
40h	Monitor SMI Status	—	0000h	R/W
42h	Reserved	—	—	—
44h	Device Trap Status	—	0000h	R/W
48h	Trap Enable register	—	0000h	R/W
4Ch–4Dh	Bus Address Tracker	—	Last Cycle	RO
4Eh	Bus Cycle Tracker	—	Last Cycle	RO
50–5Fh	Reserved	—	—	—
60h–7Fh	Reserved for TCO Registers	—	—	—

8.8.3.1 PM1_STS—Power Management 1 Status Register

I/O Address:	PMBASE + 00h (***ACPI PM1a_EVT_BLK)***	Attribute:	R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI or Legacy
Power Well:	Bits 0–7: Core, Bits 8–15: Resume, except Bit 11 in RTC		

If bit 10 or 8 in this register is 1 and the corresponding _EN bit is set in the PM1_EN register, 82801E C-ICH generates a Wake Event. Once back in an S0 state (or if already in S0 state when the event occurs), 82801E C-ICH also generates an SCI if the SCI_EN bit is set or an SMI# if the SCI_EN bit is not set.

Note: Bit 5 does not cause an SMI# or a wake event. Bit 0 does not cause a wake event but can cause an SMI# or SCI.

Bit	Description
15	**Wake Status (WAK_STS)** Not implemented. Must be set to 0.
14:12	Reserved.
11	**Power Button Override Status (PRBTNOR_STS)** Not implemented. Must be set to 0.
10	**RTC Status (RTC_STS)**—R/WC. This bit is not affected by hard resets caused by a CF9 write but is reset by RSMRST#. 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware when the RTC generates an alarm (assertion of the IRQ8# signal). Additionally if the RTC_EN bit is set, the setting of the RTC_STS bit will generate a wake event.
9	Reserved.
8	**Power Button Status (PWRBTN_STS)** Not implemented. Must be set to 0.
7:6	Reserved.
5	**Global Status (GBL _STS)**—R/WC. 1 = Set when an SCI is generated due to BIOS wanting the attention of the SCI handler. BIOS has a corresponding bit, BIOS_RLS, which will cause an SCI and set this bit. 0 = The SCI handler should then clear this bit by writing a 1 to the bit location.
4:1	Reserved.
0	**Timer Overflow Status (TMROF_STS)**—R/WC. 1 = This bit gets set any time bit 22 of the 24-bit timer goes high (bits are numbered from 0 to 23). This will occur every 2.3435 seconds. When the TMROF_EN bit is set, then the setting of the TMROF_STS bit will additionally generate an SCI or SMI# (depending on the SCI_EN). 0 = The SCI or SMI# handler clears this bit by writing a 1 to the bit location.

8.8.3.2 PM1_EN—Power Management 1 Enable Register

I/O Address:	PMBASE + 02h (***ACPI PM1a_EVT_BLK + 2)***	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI or Legacy
Power Well:	Bits 0–7: Core, Bits 8–15: Resume		

Bit	Description
15:11	Reserved.
10	**RTC Event Enable (RTC_EN)**—R/W. This is the RTC alarm enable bit. 1 = An SCI (or SMI#) will occur when this bit is set and the RTC_STS bit goes active. 0 = No SCI (or SMI#) is generated then RTC_STS goes active. ***Warning:*** This bit needs to backed by the RTC well to allow an RTC event to wake after a power failure. This bit is not cleared by any reset other than RTCRST#.
8	**Power Button Enable (PWRBTN_EN)** Not implemented. Must be set to 0.
5	**Global Enable (GBL_EN)**—R/W. When both the GBL_EN and the GBL_STS are set, an SCI is raised. 0 = Disable. 1 = Enable SCI on GBL_STS going active.
0	**Timer Overflow Interrupt Enable (TMROF_EN)**—R/W. Works in conjunction with the SCI_EN bit as described below: **TMROF_EN** / **SCI_EN** / **Effect when TMROF_STS is set** 0 / x / No SMI# or SCI 1 / 0 / SMI# 1 / 1 / SCI

8.8.3.3 PM1_CNT—Power Management 1 Control Register

I/O Address: PMBASE + 04h (*ACPI PM1a_CNT_BLK)*
Default Value: 0000h
Lockable: No
Power Well: Bits 0–7: Core, Bits 8–15: Resume
Attribute: R/W
Size: 32-bit
Usage: ACPI or Legacy

Bit	Description
13	**Sleep Enable (SLP_EN)—**WO. Not implemented. Must be set to 0.
12:10	**Sleep Type (SLP_TYP)**—R/W. Not implemented. Must be set to 0.
2	**Global Release (GBL_RLS)**—WO. 1 = ACPI software writes a 1 to this bit to raise an event to the BIOS. BIOS software has corresponding enable and status bits to control its ability to receive ACPI events. 0 = This bit always reads as 0.
0	**SCI Enable (SCI_EN)**—R/W. Selects the SCI interrupt or the SMI# interrupt for various events including the bits in the PM1_STS register (bit 10, 8, 0), and bits in GPE0_STS. 0 = These events will generate an SMI#. 1 = These events will generate an SCI.

8.8.3.4 PM1_TMR—Power Management 1 Timer Register

I/O Address: PMBASE + 08h (*ACPI PMTMR_BLK)*
Default Value: xx000000h
Lockable: No
Power Well: Core
Attribute: RO
Size: 32-bit
Usage: ACPI

Bit	Description
31:24	Reserved.
23:0	**Timer Value (TMR_VAL)**—RO. Returns the running count of the PM timer. This counter runs off a 3.579545 MHz clock (14.31818 MHz divided by 4). It is reset to zero during a PCI reset and then continues counting as long as the system is in the S0 state. Anytime bit 22 of the timer goes HIGH to LOW (bits referenced from 0 to 23), the TMROF_STS bit is set. The High-to-Low transition will occur every 2.3435 seconds. If the TMROF_EN bit is set, an SCI interrupt is also generated.

8.8.3.5 PROC_CNT—Processor Control Register

I/O Address:	PMBASE + 10h (***ACPI P_BLK)***	Attribute:	R/W
Default Value:	00000000h	Size:	32-bit
Lockable:	No (bits 7:5 are write once)	Usage:	ACPI or Legacy
Power Well:	Core		

Bit	Description
31:18	Reserved.
17	**Throttle Status (THTL_STS)**—RO. 0 = No clock throttling is occurring (maximum processor performance). 1 = Indicates that the clock state machine is in some type of low power state (where the processor is not running at its maximum performance): thermal throttling or hardware throttling.
16:9	Reserved.
8	**Force Thermal Throttling (FORCE_THTL)**—R/W. Software can set this bit to force the thermal throttling function. This has the same effect as the THRM# signal being active for 2 seconds. 0 = No forced throttling. 1 = Throttling at the duty cycle specified in THRM_DTY starts immediately (no 2 second delay), and no SMI# is generated.
7:5	**Thermal Duty Cycle (THRM_DTY).** This write-once 3-bit field determines the duty cycle of the throttling when the thermal override condition occurs. The duty cycle indicates the approximate percentage of time the STPCLK# signal is asserted while in the throttle mode. The STPCLK# throttle period is 1024 PCICLKs. Note that the throttling only occurs if the system is in the C0 state. If in the C2 state, no throttling occurs. There is no enable bit for thermal throttling, because it should not be disabled. Once the THRM_DTY field is written, any subsequent writes will have no effect until PCIRST# goes active. **THRM_DTY / Throttle Mode / PCI Clocks** 000 / RESERVED (Default) (Will be 50%) / 512 001 / 87.5% / 896 010 / 75.0% / 768 011 / 62.5% / 640 100 / 50% / 512 101 / 37.5% / 384 110 / 25% / 256 111 / 12.5% / 128
4	**Throttling Enable (THTL_EN).** When this bit is set and the system is in a C0 state, processor-controlled STPCLK# throttling is enabled. The duty cycle is selected in the THTL_DTY field. 0 = Disable 1 = Enable
3:1	**Throttling Duty Cycle (THTL_DTY).** This 3-bit field determines the duty cycle of the throttling when the THTL_EN bit is set. The duty cycle indicates the approximate percentage of time the STPCLK# signal is asserted (low) while in the throttle mode. The STPCLK# throttle period is 1024 PCICLKs. **THTL_DTY / Throttle Mode / PCI Clocks** 000 / RESERVED (Default) (Will be 50%) / 512 001 / 87.5% / 896 010 / 75.0% / 768 011 / 62.5% / 640 100 / 50% / 512 101 / 37.5% / 384 110 / 25% / 256 111 / 12.5% / 128
0	Reserved.

8.8.3.6 GPE0_STS—General Purpose Event 0 Status Register

I/O Address:	PMBASE + 28h (*ACPI GPE0_BLK)*	Attribute:	R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI
Power Well:	Resume		

Note: This register is symmetrical to the General Purpose Event 0 Enable Register. 82801E C-ICH will generate a SCI if the SCI_EN bit is set, or an SMI# if the SCI_EN bit is not set. All should be reset by SCI/SMI# on THRMOR_STS since there is no corresponding x_EN bit. None of these bits are reset by CF9h write. All are reset by RSMRST#.

Bit	Description
15:12	Reserved.
11	**PME Status (PME_STS)**—R/WC. Not implemented. Must be set to 0.
8	**RI_STS**—R/WC. 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware when the RI# input signal goes active.
7	**SMBus Wake Status (SMB_WAK_STS)**—R/WC**.** SMBus Wake Status—R/WC. The SMBus controller can independently cause an SMI# or SCI; thus, this bit does not need to do so (unlike the other bits in this register). 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware to indicate that the wake event was caused by the 82801E C-ICH's SMBus logic. This bit is set by the WAKE/SMI# command type, even if the system is already awake. The SMI handler should then clear this bit.
6	**TCO SCI Status (TCOSCI_STS)**—R/WC. 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware when the TCO logic causes an SCI.
5:4	Reserved. Must be set to 0.
3	**USB Controller 1 Status (USB1_STS)**—R/WC. 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware when USB Controller 1 needs to cause a wake.
2	Reserved.
1	**Thermal Interrupt Override Status (THRMOR_STS)**—R/WC. 0 = Software clears this bit by writing a 1 to the bit position. 1 = This bit is set by hardware anytime a thermal over-ride condition occurs and starts throttling the processor's clock at the THRM_DTY ratio. This will not cause an SMI#, SCI, or wake event.
0	**Thermal Interrupt Status (THRM_STS)**—R/WC. 0 = Software clears this bit by writing a 1 to the bit position. 1 = Set by hardware anytime the THRM# signal is driven active as defined by the THRM_POL bit. Additionally, if the THRM_EN bit is set, then the setting of the THRM_STS bit will also generate a power management event (SCI or SMI#).

8.8.3.7 GPE0_EN—General Purpose Event 0 Enables Register

I/O Address:	PMBASE + 2Ah (***ACPI GPE0_BLK + 2)***	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI
Power Well:	Bits 0–7 Resume, Bits 8–15 RTC		

Note: This register is symmetrical to the General Purpose Event 0 Status Register. The resume well bits are all cleared by RSMRST#. The RTC sell bits are cleared by RTCRST#.

Bit	Description
15:12	Reserved.
11	**PME# Enable (PME_EN)**—Not implemented. Must be set to 0.
10:9	Reserved. Must be set to 0.
8	**RI_EN**—R/W. The value of this bit will be maintained through a G3 state and is not affected by a hard reset caused by RSMRST# or a CF9h write. Assertion of RTCRST# resets this bit. 0 = Disable. 1 = Enables the setting of the RI_STS to generate a wake event.
7	Reserved.
6	**TCO SCI Enable (TCOSCI_EN)**—R/W. 0 = Disable. 1 = Enables the setting of the TCOSCI_STS to generate an SCI.
5:4	Reserved.
3	**USB Controller 1 Enable (USB1_EN)**—R/W. 0 = Disable. 1 = Enables the setting of the USB1_STS to generate a wake event.
2	**Thermal Pin Polarity (THRM#_POL)**—R/W. This bit controls the polarity of the THRM# pin needed to set the THRM_STS bit. 0 = Low value on the THRM# signal will set the THRM_STS bit. 1 = HIGH value on the THRM# signal will set the THRM_STS bit.
1	Reserved.
0	**Thermal Signal Reporting Enable (THRM_EN)**—R/W. 0 = Disable. 1 = Active assertion of the THRM# signal (as defined by the THRM_POL bit) will set the THRM_STS bit and generate a power management event (SCI or SMI).

8.8.3.8 GPE1_STS—General Purpose Event 1 Status Register

I/O Address:	PMBASE + 2Ch (***ACPI GPE1_BLK)***	Attribute:	R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI
Power Well:	Resume		

Note: This register is symmetrical to the General Purpose Event 1 Enable Register. GPIOs that are not implemented will not have the corresponding bits implemented in this register.

Note: Bits 5 and 2 are not implemented since GPIO5 and GPIO2 are not implemented.

Bit	Description
15:6	**GPI[15:6] Status (GPI[15:6]_STS)**—R/WC. 0 = Software clears each bit by writing a 1 to the bit position when the corresponding GPIO signal is not active. (The status bit cannot be cleared while the corresponding signal is still active). 1 = These bits are set any time the corresponding GPIO is set up as an input and the corresponding GPIO signal is low (or high if the corresponding GP_INV bit is set).
5	Reserved.
4:3	**GPI[4:3] Status (GPI[4:3]_STS)**—R/WC. 0 = Software clears each bit by writing a 1 to the bit position when the corresponding GPIO signal is not active. (The status bit cannot be cleared while the corresponding signal is still active). 1 = These bits are set any time the corresponding GPIO is set up as an input and the corresponding GPIO signal is low (or high if the corresponding GP_INV bit is set).
2	Reserved.
1:0	**GPI[1:0] Status (GPI[1:0]_STS)**—R/WC. 0 = Software clears each bit by writing a 1 to the bit position when the corresponding GPIO signal is not active. (The status bit cannot be cleared while the corresponding signal is still active). 1 = These bits are set any time the corresponding GPIO is set up as an input and the corresponding GPIO signal is low (or high if the corresponding GP_INV bit is set).

8.8.3.9 GPE1_EN—General Purpose Event 1 Enable Register

I/O Address:	PMBASE + 2Eh (***ACPI GPE1_BLK + 2)***	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	ACPI
Power Well:	Resume		

Note: This register is symmetrical to the General Purpose Event 1 Status Register. GPIOs that are not implemented will not have the corresponding bits implemented in this register. All of the bits in this register will be cleared by RSMRST#.

Note: Bits 5 and 2 are not implemented since GPIO5 and GPIO2 are not implemented.

Bit	Description
15:6	**GPI[15:6] Enable (GPI[15:6]_EN)**—R/W. 1 = Enable the corresponding GPI[n]_STS bit being set to cause an SMI#, SCI. 0 = Disable.
5	Reserved.
4:3	**GPI[4:3] Enable (GPI[4:3]_EN)**—R/W. 1 = Enable the corresponding GPI[n]_STS bit being set to cause an SMI#, SCI. 0 = Disable.
2	Reserved.
1:0	**GPI[1:0] Enable (GPI[1:0]_EN**)—R/W. 1 = Enable the corresponding GPI[n]_STS bit being set to cause an SMI#, SCI. 0 = Disable.

8.8.3.10 SMI_EN—SMI Control and Enable Register

I/O Address:	PMBASE + 30h	Attribute:	R/W
Default Value:	0000h	Size:	32 bit
Lockable:	No	Usage:	ACPI or Legacy
Power Well:	Core		

Note: This register is symmetrical to the SMI status register.

Bit	Description
31:15	Reserved.
14	**Periodic SMI# Enable (PERIODIC_EN**)—R/W. 0 = Disable. 1 = Enables the 82801E C-ICH to generate an SMI# when the PERIODIC_STS bit is set in the SMI_STS register.
13	**TCO Enable (TCO_EN)**—R/W. 0 = Disables TCO logic generating an SMI#. Note that if the NMI2SMI_EN bit is set, SMIs that are caused by re-routed NMIs will not be gated by the TCO_EN bit. Even if the TCO_EN bit is 0, NMIs will still be routed to cause SMIs. 1 = Enables the TCO logic to generate SMI#.
12	Reserved.

Bit	Description
11	**Microcontroller SMI# Enable (MCSMI_EN)**—R/W. 0 = Disable. 1 = Enables 82801E C-ICH to trap accesses to the microcontroller range (62h or 66h) and generate an SMI#. Note that "trapped" cycles will be claimed by the 82801E C-ICH on PCI, but not forwarded to LPC.
10:8	Reserved.
7	**BIOS Release (BIOS_RLS)**—WO. 0 = This bit will always return 0 on reads. Writes of 0 to this bit have no effect. 1 = Enables the generation of an SCI interrupt for ACPI software when a one is written to this bit position by BIOS software.
6	**Software SMI# Timer Enable (SWSMI_TMR_EN)**—R/W. 0 = Disable. Clearing the SWSMI_TMR_EN bit before the timer expires will reset the timer and the SMI# will not be generated. 1 = Starts Software SMI# Timer. When the SWSMI timer expires (the time-out period depends upon the SWSMI_RATE_SEL bit setting), SWSMI_TMR_STS is set and an SMI# is generated. SWSMI_TMR_EN stays set until cleared by software.
5	**APMC Enable (APMC_EN)**—R/W. 0 = Disable. Writes to the APM_CNT register will not cause an SMI#. 1 = Enables writes to the APM_CNT register to cause an SMI#.
4	**SLP SMI Enable (SLP_SMI_EN)**—R/W. 0 = Disables the generation of SMI# on SLP_EN. Note that this bit must be 0 before the software attempts to transition the system into a sleep state by writing a 1 to the SLP_EN bit. 1 = A write of 1 to the SLP_EN bit (bit 13 in PM1_CNT register) will generate an SMI#, and the system will not transition to the sleep state based on that write to the SLP_EN bit.
3	**Legacy USB Enable (LEGACY_USB_EN)**—R/W. 0 = Disable. 1 = Enables legacy USB circuit to cause SMI#.
2	**BIOS Enable (BIOS_EN)**—R/W. 0 = Disable. 1 = Enables the generation of SMI# when ACPI software writes a 1 to the GBL_RLS bit.
1	**End of SMI (EOS)**—R/W (special). This bit controls the arbitration of the SMI signal to the processor. This bit must be set for the 82801E C-ICH to assert SMI# low to the processor. 1 = When this bit is set, SMI# signal will be deasserted for 4 PCI clocks before its assertion. In the SMI handler, the processor should clear all pending SMIs (by servicing them and then clearing their respective status bits), set the EOS bit, and exit SMM. This will allow the SMI arbiter to re-assert SMI upon detection of an SMI event and the setting of a SMI status bit. 0 = Once the 82801E C-ICH asserts SMI# low, the EOS bit is automatically cleared.
0	**Global SMI Enable (GBL_SMI_EN)**—R/W. 0 = No SMI# will be generated by 82801E C-ICH. This bit is reset by a PCI reset event. 1 = Enables the generation of SMI# in the system upon any enabled SMI event.

8.8.3.11 SMI_STS—SMI Status Register

I/O Address:	PMBASE + 34h	Attribute:	R/W
Default Value:	0000h	Size:	32-bit
Lockable:	No	Usage:	ACPI or Legacy
Power Well:	Core		

Note: If the corresponding _EN bit is set when the _STS bit is set, the 82801E C-ICH will cause an SMI# (except bits 8:10 and 12, which do not need enable bits since they are logic ORs of other registers that have enable bits).

Bit	Description
31:17	Reserved.
16	**SMBus SMI Status (SMBUS_SMI_STS)**—R/WC. 1 = Indicates that the SMI# was caused by either the SMBus Slave receiving a message, or the SMBALERT# signal going active. This bit will be set on SMBALERT# assertion only if the SMBus Host Controller is programmed to generate SMIs (not interrupts). 0 = This bit is cleared by writing a 1 to its bit position.
15	**SERR IRQ SMI Status (SERIRQ_SMI_STS)**—RO. 1 = Indicates that the SMI# was caused by the SERIRQ decoder. 0 = SMI# was not caused by SERIRQ decoder. This is not a sticky bit.
14	**Periodic Status (PERIODIC_STS)**—R/WC. 1 = This bit will be set at the rate determined by the PER_SMI_SEL bits. If the PERIODIC_EN bit is also set, the 82801E C-ICH will generate an SMI#. 0 = This bit is cleared by writing a 1 to its bit position.
13	**TCO Status (TCO_STS)**—RO. 0 = SMI# not caused by TCO logic. 1 = Indicates the SMI# was caused by the TCO logic. Note that this is not a wake event.
12	**Device Monitor Status (DEVMON_STS)**—RO. 1 = Set under any of the following conditions: - Any of the DEV[7:4]_TRAP_STS bits are set and the corresponding DEV[7:4]_TRAP_EN bits are also set. - Any of the DEVTRAP_STS bits are set and the corresponding DEVTRAP_EN bits are also set. 0 = SMI# not caused by Device Monitor.
11	**Microcontroller SMI# Status (MCSMI_STS)**—R/WC. 0 = Indicates that there has been no access to the power management microcontroller range (62h or 66h). This bit is cleared by software writing a 1 to the bit position. 1 = Set if there has been an access to the power management microcontroller range (62h or 66h). If this bit is set, and the MCSMI_EN bit is also set, the 82801E C-ICH will generate an SMI#.
10	**GPE1 Status (GPE1_STS**)—RO. This bit is a logical OR of the bits in the GPE1_STS register that are also set up to cause an SMI# (as indicated by the GPI_ROUT registers) and have the corresponding bit set in the GPE1_EN register. Bits that are not routed to cause an SMI# will have no effect on the GPE1_STS bit. 0 = SMI# was not generated by a GPI assertion. 1 = SMI# was generated by a GPI assertion.
9	**GPE0 Status (GPE0_STS)**—RO. This bit is a logical OR of the bits in the GPE0_STS register that also have the corresponding bit set in the GPE0_EN register. 0 = SMI# was not generated by a GPE0 event. 1 = SMI# was generated by a GPE0 event.
8	**PM1 Status Register (PM1_STS_REG)**—RO. This is an OR of the bits in the ACPI PM1 Status Register (offset PMBASE+00h) that can cause an SMI#. 0 = SMI# was not generated by a PM1_STS event. 1 = SMI# was generated by a PM1_STS event.

Bit	Description
7	Reserved.
6	**Software SMI Timer Status (SWSMI_TMR_STS)**—R/WC. 1 = Set by the hardware when the Software SMI# Timer expires. 0 = Software clears this bit by writing a 1 to the bit location.
5	**APM Status (APM_STS)**—R/WC. 1 = SMI# was generated by a write access to the APM control register with the APMC_EN bit set. 0 = Software clears this bit by writing a 1 to the bit location.
4	**SLP SMI Status (SLP_SMI_STS)**—R/WC. 1 = Indicates an SMI# was caused by a write of 1 to SLP_EN bit when SLP_SMI_EN bit is also set. 0 = Software clears this bit by writing a 1 to the bit location.
3	**Legacy USB Status (LEGACY_USB_STS**)—RO. This bit is a logical OR of each of the SMI status bits in the USB Legacy Keyboard/Mouse Control Registers ANDed with the corresponding enable bits. This bit will not be active if the enable bits are not set. 0 = SMI# was not generated by USB Legacy event. 1 = SMI# was generated by USB Legacy event.
2	**BIOS Status (BIOS_STS)**—R/WC. 1 = SMI# was generated due to ACPI software requesting attention (writing a 1 to the GBL_RLS bit with the BIOS_EN bit set). 0 = This bit cleared by software writing a 1 to its bit position.
1:0	Reserved.

8.8.3.12 MON_SMI—Device Monitor SMI Status and Enable Register

I/O Address:	PMBASE +40h	Attribute:	R/W, R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	Legacy Only
Power Well:	Core		

Bit	Description
15:12	**Device 7:4 Trap Status (DEV[7:4]_TRAP_STS)**—R/WC. Bit 12 corresponds to Monitor 4, bit 13 corresponds to Monitor 5 etc. 1 = SMI# was caused by an access to the corresponding device monitor's I/O range. 0 = SMI# was not caused by the associated device monitor.
11:8	**Device 7:4 Trap Enable (DEV[7:4]_TRAP_EN)**—R/W. Bit 8 corresponds to Monitor 4, bit 9 corresponds to Monitor 5 etc. 1 = Enables SMI# due to an access to the corresponding device monitor's I/O range. 0 = Disable.
7:0	Reserved.

8.8.3.13 DEVACT_STS—Device Activity Status Register

I/O Address:	PMBASE +44h	Attribute:	R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Usage:	Legacy Only
Power Well:	Core		

This register is used in conjunction with the Periodic SMI# timer to detect any system activity for legacy power management.

Bit	Description
15:14	Reserved.
13	**ADLIB Activity Status (ADLIB_ACT_STS)**—R/WC. 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
12	**Keyboard Controller Activity Status (KBC_ACT_STS)**—R/WC. KBC (60/64h). 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
11	**MIDI Activity Status (MIDI_ACT_STS)**—R/WC. 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
10	**Audio Activity Status (AUDIO_ACT_STS)**—R/WC. Audio (Sound Blaster "ORed" with MSS). 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
9	**PIRQ[D or H] Activity Status (PIRQDH_ACT_STS)**—R/WC. 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.
8	**PIRQ[C or G] Activity Status (PIRQCG_ACT_STS)**—R/WC. 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.
7	**PIRQ[B or F] Activity Status (PIRQBF_ACT_STS)**—R/WC. 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.
6	**PIRQ[A or E] Activity Status (PIRQAE_ACT_STS)**—R/WC. 0 = The corresponding PCI interrupts have not been active. 1 = At least one of the corresponding PCI interrupts has been active. Clear this bit by writing a 1 to the bit location.
5	**Legacy Activity Status (LEG_ACT_STS)**—R/WC. Parallel Port, Serial Port 1, Serial Port 2, Floppy Disk Controller. 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
4	Reserved.
3	**IDE Secondary Drive 1 Activity Status (IDES1_ACT_STS)**—R/WC. 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.

Bit	Description
2	**IDE Secondary Drive 0 Activity Status (IDES0_ACT_STS)**—R/WC. 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
1	**IDE Primary Drive 1 Activity Status (IDEP1_ACT_STS)**—R/WC. 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.
0	**IDE Primary Drive 0 Activity Status (IDEP0_ACT_STS)**—R/WC. 0 = Indicates that there has been no access to this device's I/O range. 1 = This device's I/O range has been accessed. Clear this bit by writing a 1 to the bit location.

8.8.3.14 DEVTRAP_EN—Device Trap Enable Register

I/O Address:	PMBASE +48h	Attribute:	R/W
Default Value	0000h	Size:	16-bit
Lockable:	No	Usage:	Legacy Only
Power Well:	Core		

This register enables the individual trap ranges to generate an SMI# when the corresponding status bit in the DEVACT_STS register is set. When a range is enabled, I/O cycles associated with that range will not be forwarded to LPC or IDE.

Bit	Description
15:14	Reserved.
13	**ADLIB Trap Enable (ADLIB_TRP_EN)**—R/W. 0 = Disable. 1 = Enable.
12	**KBC Trap Enable (KBC_TRP_EN)**—R/W. KBC (60/64h). 0 = Disable. 1 = Enable.
11	**MIDI Trap Enable (MIDI_TRP_EN)**—R/W. 0 = Disable. 1 = Enable.
10	**Audio Trap Enable (AUDIO_TRP_EN)**—R/W. Audio (Sound Blaster "ORed" with MSS). 0 = Disable. 1 = Enable.
9:6	Reserved.
5	**LEG_IO_TRP_EN**—R/W. Parallel Port, Serial Port 1, Serial Port 2, Floppy Disk Controller. 0 = Disable. 1 = Enable.
4	Reserved.
3	**IDE Secondary Drive 1 Trap Enable (IDES1_TRP_EN)**—R/W. 0 = Disable. 1 = Enable.

Bit	Description
2	**IDE Secondary Drive 0 Trap Enable (IDES0_TRP_EN)—**R/W. 0 = Disable. 1 = Enable.
1	**IDE Primary Drive 1 Trap Enable (IDEP1_TRP_EN)**—R/W. 0 = Disable. 1 = Enable.
0	**IDE Primary Drive 0 Trap Enable (IDEP0_TRP_EN)—**R/W. 0 = Disable. 1 = Enable.

8.8.3.15 BUS_ADDR_TRACK—Bus Address Tracker Register

I/O Address:	PMBASE +4Ch	Attribute:	RO
Lockable:	No	Size:	16-bit
Power Well:	Core	Usage:	Legacy Only

This register could be used by the SMI# handler to assist in determining what was the last cycle from the processor.

The value stored in these registers are based on an SMI event, which can include an asynchronous SMI, not just synchronous SMIs such as an SMI_TRAP.

Bit	Description
15:0	Corresponds to the low 16 bits of the last I/O cycle, as would be defined by the PCI AD[15:0] signals on the PCI bus (even though it may not be a real PCI cycle). The value is latched based on SMI# active. This functionality is useful for figuring out which I/O was last being accessed.

8.8.3.16 BUS_CYC_TRACK—Bus Cycle Tracker Register

I/O Address:	PMBASE +4Eh	Attribute:	RO
Lockable:	No	Size:	8-bit
Power Well:	Core	Usage:	Legacy Only

This register could be used by the SMM handler to assist in determining what was the last cycle from the processor.

The value stored in these registers are based on an SMI event, which can include an asynchronous SMI, not just synchronous SMIs such as an SMI_TRAP.

Bit	Description
7:4	Corresponds to the byte enables, as would be defined by the PCI C/BE# signals on the PCI bus (even though it may not be a real PCI cycle). The value is latched based on SMI# going active.
3:0	Corresponds to the cycle type, as would be defined by the PCI C/BE# signals on the PCI bus (even though it may not be a real PCI cycle). The value is latched based on SMI# going active.

8.9 System Management TCO Registers (D31:F0)

The TCO logic is accessed via registers mapped to the PCI configuration space (Device 31:Function 0) and the system I/O space. For TCO PCI Configuration registers, see LPC Device 31:Function 0 PCI Configuration registers.

8.9.1 TCO Register I/O Map

The TCO I/O registers reside in a 32-byte range pointed to by a TCOBASE value, which is, ACPIBASE + 60h in the PCI configuration space. The following table shows the mapping of the registers within that 32-byte range. Each register is described in the sections below.

Table 102. TCO I/O Register Map

Offset	Mnemonic	Register Name: Function	Type
00h	TCO_RLD	TCO Timer Reload and Current Value	R/W
01h	TCO_TMR	TCO Timer Initial Value	R/W
02h	TCO_DAT_IN	TCO Data In	R/W
03h	TCO_DAT_OUT	TCO Data Out	R/W
04h–05h	TCO1_STS	TCO Status	R/W
06h–07h	TCO2_STS	TCO Status	R/W
08h–09h	TCO1_CNT	TCO Control	R/W
0Ah–0Bh	TCO2_CNT	TCO Control	R/W
0Ch–0Dh	TCO_MESSAGE1, TCO_MESSAGE2	Used by BIOS to indicate POST/Boot progress	R/W
0Eh	TCO_WDSTATUS	Watchdog Status Register	R/W
0Fh		Reserved	RO
10h	SW_IRQ_GEN	Software IRQ Generation Register	R/W
11h–1Fh		Reserved	RO

8.9.2 TCO1_RLD—TCO Timer Reload and Current Value Register

I/O Address:	TCOBASE +00h	Attribute:	R/W
Default Value:	0000h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:0	**TCO Timer Value.** Reading this register will return the current count of the TCO timer. Writing any value to this register will reload the timer to prevent the time-out. Bits 7:6 will always be 0.

8.9.3 TCO1_TMR—TCO Timer Initial Value Register

I/O Address:	TCOBASE +01h	Attribute:	R/W
Default Value:	0004h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:6	Reserved.
5:0	**TCO Timer Initial Value.** Value that is loaded into the timer each time the TCO_RLD register is written. Values of 0h–3h will be ignored and should not be attempted. The timer is clocked at approximately 0.6 seconds, and this allows time-outs ranging from 2.4 seconds to 38 seconds.

8.9.4 TCO1_DAT_IN—TCO Data In Register

I/O Address:	TCOBASE +02h	Attribute:	R/W
Default Value:	0000h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:0	**TCO Data In Value.** Data Register for passing commands from the OS to the SMI handler. Writes to this register will cause an SMI and set the OS_TCO_SMI bit in the TCO_STS register.

8.9.5 TCO1_DAT_OUT—TCO Data Out Register

I/O Address:	TCOBASE +03h	Attribute:	R/W
Default Value:	0000h	Size:	8-bit
Lockable:	No	Power Well:	Core

Bit	Description
7:0	**TCO Data Out Value.** Data Register for passing commands from the SMI handler to the OS. Writes to this register will set the TCO_INT_STS bit in the TCO_STS register. It will also cause an interrupt, as selected by the TCO_INT_SEL bits.

8.9.6 TCO1_STS—TCO1 Status Register

I/O Address:	TCOBASE +04h	Attribute:	R/WC RO
Default Value:	0000h	Size:	16-bit
Lockable:	No	Power Well:	Core (Except bit 7, in RTC)

Bit	Description
15:13	Reserved.
12	**Hub Interface SERR Status (HUBSERR_STS)**—R/WC. 1 = 82801E C-ICH received an SERR# message via the hub interface. The software must read the memory controller hub (or its equivalent) to determine the reason for the SERR#. 0 = Software clears this bit by writing a 1 to the bit position.
11	**Hub Interface NMI Status (HUBNMI_STS)**—R/WC. 1 = 82801E C-ICH received an NMI message via the hub interface. The software must read the memory controller hub (or its equivalent) to determine the reason for the NMI. 0 = Software clears this bit by writing a 1 to the bit position.
10	**Hub Interface SMI Status (HUBSMI_STS)**—R/WC. 1 = 82801E C-ICH received an SMI message via the hub interface. The software must read the memory controller hub (or its equivalent) to determine the reason for the SMI#. 0 = Software clears this bit by writing a 1 to the bit position.
9	**Hub Interface SCI Status (HUBSCI_STS)**—R/WC. 1 = 82801E C-ICH received an SCI message via the hub interface. The software must read the memory controller hub (or its equivalent) to determine the reason for the SCI. 0 = Software clears this bit by writing a 1 to the bit position.
8	**BIOS Write Status (BIOSWR_STS)**—R/WC. 1 = 82801E C-ICH sets this bit and generates and SMI# to indicate an illegal attempt to write to the BIOS. This occurs when either: a) The BIOSWP bit is changed from 0 to 1 and the BLD bit is also set, or b) any write is attempted to the BIOS and the BIOSWP bit is also set. 0 = Software clears this bit by writing a 1 to the bit position. **NOTE:** On write cycles attempted to the 4 Mbyte lower alias to the BIOS space, the BIOSWR_STS will not be set.
7	**New Century Status (NEWCENTURY_STS)**—R/WC. This bit is in the RTC well. 1 = This bit is set when the Year byte (RTC I/O space, index offset 09h) rolls over from 99 to 00. Setting this bit will cause an SMI# (but not a wake event). 0 = Cleared by writing a 1 to the bit position or by RTCRST# going active. Note that the NEWCENTURY_STS bit is not valid when the RTC battery is first installed (or when RTC power has not been maintained). Software can determine if RTC power has not been maintained by checking the RTC_PWR_STS bit or by other means (e.g., a checksum on RTC RAM). If RTC power is determined to have not been maintained, BIOS should set the time to a legal value and then clear the NEWCENTURY_STS bit. The NEWCENTURY_STS bit may take up to 3 RTC clocks for the bit to be cleared after a “1” is written to the bit to clear it. After writing a “1” to this bit, software should not exit the SMI handler until verifying that the bit has actually been cleared. This will ensure that the SMI is not re-entered.
6:4	Reserved.
3	**Time Out Status (TIMEOUT)**—R/WC. 1 = Set by 82801E C-ICH to indicate that the SMI was caused by the TCO timer reaching 0. 0 = Software clears this bit by writing a 1 to the bit position.
2	**TCO Interrupt Status (TCO_INT_STS)**—R/WC. 1 = SMI handler caused the interrupt by writing to the TCO_DAT_OUT register. 0 = Software clears this bit by writing a 1 to the bit position.
1	**Software TCO SMI Status (SW_TCO_SMI)**—R/WC. 1 = Software caused an SMI# by writing to the TCO_DAT_IN register. 0 = Software clears this bit by writing a 1 to the bit position.
0	**NMI to SMI Status (NMI2SMI_STS)**—RO. 1 = Set by the 82801E C-ICH when an SMI# occurs because an event occurred that would otherwise have caused an NMI (because NMI2SMI_EN is set). 0 = Cleared by clearing the associated NMI status bit.

8.9.7 TCO2_STS—TCO2 Status Register

I/O Address:	TCOBASE +06h	Attribute:	R/WC, RO
Default Value:	0000h	Size:	16-bit
Lockable:	No	Power Well:	Resume (Except Bit 0, in RTC)

Bit	Description
15:4	Reserved.
3	**BAD_BIOS:** This bit is set by the 82801E C-ICH when it detects FFh on the first BIOS read (i.e. the BIOS is bad). 82801E C-ICH clears this bit to 0 if the first BIOS read is not FFh. This is detected when the initial read returns FFh from the FWH. This bit is not intended to be read by the BIOS or software. It is only used for sending the TCO/Heartbeat messages to the D110. Reads to this bit always return 0 and writes have no effect.
2	**Boot Status (BOOT_STS):** 1 = Set to 1 when the SECOND_TO_STS bit goes from 0 to 1 and the processor has not fetched the first instruction. 0 = Cleared by 82801E C-ICH based on RSMRST# or by software writing a 1 to this bit. Note that software should first clear the SECOND_TO_STS bit before writing a 1 to clear the BOOT_STS bit. If rebooting due to a second TCO timer time-out and if the BOOT_STS bit is set, the 82801E C-ICH will reboot using the 'safe' multiplier (1111). This allows the system to recover from a processor frequency multiplier that is too high, and allows the BIOS to check the BOOT_STS bit at boot. If the bit is set and the frequency multiplier is 1111, then the BIOS knows that the processor has been programmed to an illegal multiplier.
1	**Second TCO Time-out Status (SECOND_TO_STS)**—R/WC. 1 = The 82801E C-ICH sets this bit to a 1 to indicate that the TCO timer timed out a second time (probably due to system lock). If this bit is set the 82801E C-ICH will reboot the system after the second time-out. The reboot is done by asserting PCIRST#. 0 = This bit is cleared by writing a 1 to the bit position or by a RSMRST#.
0	**Intruder Detect (INTRD_DET)**—R/WC. 1 = Set by 82801E C-ICH to indicate that an intrusion was detected. This bit is set even if the system is in G3 state. 0 = This bit is only cleared by writing a 1 to the bit position, or by RTCRST# assertion.

8.9.8 TCO1_CNT—TCO1 Control Register

I/O Address:	TCOBASE +08h	Attribute:	R/W, R/WC
Default Value:	0000h	Size:	16-bit
Lockable:	No	Power Well:	Core

Bit	Description
15:12	Reserved.
11	**TCO Timer Halt (TCO_TMR_HLT)—**R/W. 0 = The TCO Timer is enabled to count. 1 = The TCO Timer will halt. It will not count and, thus, cannot reach a value that will cause an SMI# or set the SECOND_TO_STS bit. When set, this bit prevents rebooting and prevents Alert On LAN event messages from being transmitted on the SMLINK (but not Alert On LAN heartbeat messages).
10	**Send Now (SENDNOW)**—R/W (special). 1 = Writing a 1 to this bit will cause the ICH to send an Alert On LAN Event message over the SMLINK interface, with the Software Event bit set. 0 = The ICH will clear this bit when it has completed sending the message. Software must not set this bit to 1 again until the ICH has set it back to 0. Setting the SENDNOW bit causes the 82801E C-ICH integrated LAN Controller to reset, which can have unpredictable side-effects. Unless software protects against these side effects, software should not attempt to set this bit.
9	**NMI to SMI Enable (NMI2SMI_EN)**—R/W. 0 = Normal NMI functionality. 1 = Forces all NMIs to instead cause SMIs. The functionality of this bit is dependent upon the settings of the NMI_EN bit and the GBL_SMI_EN bit as detailed in the following table: **NMI_EN** / **GBL_SMI_EN** / **Description** 0 / 0 / No SMI# at all because GBL_SMI_EN = 0 0 / 1 / SMI# will be caused due to NMI events 1 / 0 / No SMI# at all because GBL_SMI_EN = 0 1 / 1 / No SMI# due to NMI because NMI_EN = 1
8	**NMI Now (NMI_NOW)**—R/WC. 1 = Writing a 1 to this bit causes an NMI. This allows the BIOS or SMI handler to force an entry to the NMI handler. 0 = This bit is cleared by writing a 1 to the bit position. The NMI handler is expected to clear this bit. Another NMI will not be generated until the bit is cleared.
7:0	Reserved.

8.9.9 TCO2_CNT—TCO2 Control Register

I/O Address:	TCOBASE +0Ah	Attribute:	R/W
Default Value:	0000h	Size:	16-bit
Lockable:	No	Power Well:	Resume

Bit	Description
15:3	Reserved.
2:1	**INTRUDER# Signal Select (INTRD_SEL)**—R/W. Selects the action to take if the INTRUDER# signal goes active. 00 = No interrupt or SMI# 01 = Interrupt (as selected by TCO_INT_SEL). 10 = SMI 11 = Reserved
0	Reserved.

8.9.10 TCO_MESSAGE1 and TCO_MESSAGE2 Registers

I/O Address:	TCOBASE +0Ch (Message 1) TCOBASE +0Dh (Message 2)	Attribute:	R/W
Default Value:	00h	Size:	8-bit
Lockable:	No	Power Well:	Resume

Bit	Description
7:0	**TCO Message (TCO_MESSAGE[*n*])**—R/W.The value written into this register will be sent out via the SMLINK interface in the MESSAGE field of the Alert On LAN message. BIOS can write to this register to indicate its boot progress which can be monitored externally.

8.9.11 TCO_WDSTATUS—TCO2 Control Register

Offset Address:	TCOBASE + 0Eh	Attribute:	R/W
Default Value:	00h	Size:	8 bits
Power Well:	Resume		

Bit	Description
7:0	**Watchdog Status (WDSTATUS)**—R/W. The value written to this register will be sent in the Alert On LAN message on the SMLINK interface. It can be used by the BIOS or system management software to indicate more details on the boot progress. This register will be reset to the default of 00h based on RSMRST# (but not PCI reset).

8.9.12 SW_IRQ_GEN—Software IRQ Generation Register

Offset Address:	TCOBASE + 10h	Attribute:	R/W
Default Value:	03h	Size:	8 bits
Power Well:	Resume		

Bit	Description
7:2	Reserved.
1	**IRQ12 Cause (IRQ12_CAUSE)**—R/W. The state of this bit is logically ANDed with the IRQ12 signal as received by the 82801E C-ICH's SERIRQ logic. This bit must be a "1" (default) if the 82801E C-ICH is expected to receive IRQ12 assertions from a SERIRQ device.
0	**IRQ1 Cause (IRQ1_CAUSE)**—R/W. The state of this bit is logically ANDed with the IRQ1 signal as received by the 82801E C-ICH's SERIRQ logic. This bit must be a "1" (default) if the 82801E C-ICH is expected to receive IRQ1 assertions from a SERIRQ device.

8.10 General Purpose I/O Registers (D31:F0)

The control for the general purpose I/O signals is handled through a separate 64-byte I/O space. The base offset for this space is selected by the GPIO_BAR register. Table 103 summarizes the 82801E C-ICH GPIO implementation.

The GPIO signals are divided into several types:

Type	82801E C-ICH Allocation (Note: Some GPIO are not implemented)
General Purpose Input on Core Well	Input GPIO[0:7] Can cause a wake event, SMI#, or SCI.
General Purpose Input on Resume Well	Input GPIO[8:15] Can cause a wake event, SMI#, or SCI.
General Purpose Output on Core Well	Output GPIO[16:23]
General Purpose Input/Output on Resume Well	GPIO[24:31]

Note: For the following registers, if a bit is allocated for a GPIO that doesn't exist, unless otherwise indicated, the bit will always read as 0 and values written to that bit will have no effect.

Table 103. Summary of GPIO Implementation

GPIO	Type	Alternate Function (Note 1)	Power Well	Notes
GPIO[0]	Input Only	REQ[A]#	Core	GPIO_USE_SEL bit 0 enables REQ/GNT[A]# pair. Input active status read from GPE1_STS register bit 0. Input active high/low set through GPI_INV register bit 0.
GPIO[1]	Input Only	REQ[B]# or REQ[5]#	Core	GPIO_USE_SEL bit 1 enables REQ/GNT[B]# pair (See note 3). Input active status read from GPE1_STS register bit 1. Input active high/low set through GPI_INV register bit 1.
GPIO[2]	N/A	N/A	N/A	Not implemented
GPIO[3:4]	Input Only	PIRQ[F:G]#	Core	GPIO_USE_SEL bits [3:4] enable PIRQ[F:G]#. Input active status read from GPE1_STS reg. bits [3:4]. Input active high/low set through GPI_INV reg. bit [3:4].
GPIO[5]	N/A	N/A	N/A	Not implemented
GPIO[7]	Input Only	Unmuxed	Core	Input active status read from GPE1_STS register bit 7. Input active high/low set through GPI_INV register bit 7
GPIO[8]	Input Only	Unmuxed	Resume	Input active status read from GPE1_STS register bit 8. Input active high/low set through GPI_INV register bit 8.
GPIO[9:10]	N/A	N/A	N/A	Not implemented
GPIO[11]	Input Only	SMBALERT#	Resume	GPIO_USE_SEL bit 11 enables SMBALERT# Input active status read from GPE1_STS register bit 11. Input active high/low set through GPI_INV register bit 11.
GPIO[12]	Input Only	Unmuxed	Resume	Input active status read from GPE1_STS register bit 12. Input active high/low set through GPI_INV register bit 12.
GPIO[13]	Input Only	Unmuxed	Resume	Input active status read from GPE1_STS register bit 13. Input active high/low set through GPI_INV register bit 13.
GPIO[14:15]	N/A	N/A	N/A	Not Implemented
GPIO[16]	Output Only	GNT[A]#	Core	Output controlled via GP_LVL register bit 16. TTL driver output
GPIO[17]	Output Only	GNT[B]# or GNT[5]#	Core	Output controlled via GP_LVL register bit 17. TTL driver output
GPIO[21]	Output Only	Unmuxed	Core	This GPO defaults high. Output controlled via GP_LVL register bit 21. TTL driver output
GPIO[25]	Input / Output	Unmuxed	Resume	Blink enabled via GPO_BLINK register bit 25. Input active status read from GP_LVL register bit 25 Output controlled via GP_LVL register bit 25. TTL driver output
GPIO[26]	N/A	N/A	N/A	Not implemented
GPIO[27:28]	Input / Output	Unmuxed	Resume	Input active status read from GP_LVL register bits [27:28] Output controlled via GP_LVL register bits [27:28] TTL driver output
GPIO[29:31]	N/A	N/A	N/A	Not implemented

NOTES:
1. All GPIOs default to their alternate function
2. All inputs are sticky. The status bit will remain set as long as the input was asserted for 2 clocks.
3. GPIO[7:6,4:3,1:0] are 5V tolerant, and all GPIs can be routed to cause an SCI or SMI#
4. If GPIO_USE_SEL bit 1 is set to 1 and GEN_CNT bit 25 is also set to 1 then REQ/GNT[5]# is enabled. See Section 8.1.22.

8.10.1 GPIO Register I/O Address Map

Table 104. Registers to Control GPIO

Offset	Mnemonic	Register Name	Default	Access
		General Registers		
00h–03h	GPIO_USE_SEL	GPIO Use Select	1A00 3180h	R/W
04h–07h	GP_IO_SEL	GPIO Input/Output Select	0000 FFFFh	R/W
08h–0Bh	—	Reserved	00h	RO
0hC–0Fh	GP_LVL	GPIO Level for Input or Output	1B3F 0000h	R/W
10h–13h	—	Reserved	00h	RO
		Output Control Registers		
14h–17h	GPO_TTL	GPIO TTL Select	06630000h	RO
18h–1Bh	GPO_BLINK	GPIO Blink Enable	00040000h	R/W
1Ch–1Fh	—	Reserved	0	RO
		Input Control Registers		
20h–2Bh	—	Reserved	00000000h	RO
2Ch–2Fh	GPI_INV	GPIO Signal Invert	00000000h	R/W

8.10.2 GPIO_USE_SEL—GPIO Use Select Register

Offset Address: GPIOBASE + 00h
Default Value: 1A003180h
Lockable: Yes
Attribute: R/W
Size: 32-bit
Power Well: Core for 0:7 and 16:23
Resume for 8:15 and 24:31

Bit	Description
21,11, 5:0	**GPIO Use Select (GPIO_USE_SEL)**—R/W. Each bit in this register enables the corresponding GPIO (if it exists) to be used as a GPIO, rather than for the native function. 0 = Signal used as native function. 1 = Signal used as a GPIO. **NOTE:** Bits 31:29, 26, 15:14, and 10:9 are not implemented because there is no corresponding GPIO. **NOTE:** Bits 28:27, 25:22, 20:18,13:12, 8 and 6 are not implemented because the corresponding GPIOs are not multiplexed.

8.10.3 GP_IO_SEL—GPIO Input/Output Select Register

Offset Address:	GPIOBASE +04h	Attribute:	R/W
Default Value:	0000FFFFh	Size:	32-bit
Lockable:	No	Power Well:	Resume

Bit	Description
31:29	Reserved.
28:27	**GPIO[n] Select (GPIO[n]_SEL)**—R/W. 0 = Output. The corresponding GPIO signal is an output. 1 = Input. The corresponding GPIO signal is an input.
26	Reserved.
25:24	**GPIO[n] Select (GPIO[n]_SEL)**—R/W. 0 = Output. The corresponding GPIO signal is an output. 1 = Input. The corresponding GPIO signal is an input.
23:16	Always 0. The GPIOs are fixed as outputs.
15:0	Always 1. These GPIOs are fixed as inputs.

NOTE:
1. There will be some delay on GPIO[24:28] going to their default state based on the rising edge of RSMRST#. This is the case since these signals are in the resume well and resume well outputs are not valid until after RSMRST# goes high.

8.10.4 GP_LVL—GPIO Level for Input or Output Register

Offset Address:	GPIOBASE +0Ch	Attribute:	R/W, RO
Default Value:	1B3F 0000h	Size:	32-bit
Lockable:	No	Power Well:	See bit descriptions

Bit	Description
31:29	Reserved.
28:27	**GPIO Level (GP_LVL[n])—**R/W. If GPIO[n] is programmed to be an output (via the corresponding bit in the GP_IO_SEL register), then the bit can be updated by software to drive a high or low value on the output pin. If GPIO[n] is programmed as an input, then software can read the bit to determine the level on the corresponding input pin. These bits correspond to GPIO that are in the Resume well, and will be reset to their default values by RSMRST# but not by PCIRST#. 0 = Low 1 = High
26	Reserved.
25:24	**GPIO Level (GP_LVL[n])—**R/W. If GPIO[n] is programmed to be an output (via the corresponding bit in the GP_IO_SEL register), then the bit can be updated by software to drive a high or low value on the output pin. If GPIO[n] is programmed as an input, then software can read the bit to determine the level on the corresponding input pin. These bits correspond to GPIO that are in the Resume well, and will be reset to their default values by RSMRST# but not by PCIRST#. 0 = Low 1 = High
23:16	**GPIO Level (GP_LVL[n])—**R/W. These bits can be updated by software to drive a high or low value on the output pin. These bits correspond to GPIO that are in the Core well, and will be reset to their default values by PCIRST#. 0 = Low 1 = High
15:14	Reserved.
13:11	For GPI[13:11] the active status of a GPI is read from the corresponding bit in GPE1_STS register.
10:9	Reserved.
8:6	For GPI[8:6], the active status of a GPI is read from the corresponding bit in GPE1_STS register.
5	Reserved.
4:3	For GPI[4:3], the active status of a GPI is read from the corresponding bit in GPE1_STS register.
2	Reserved.
1:0	For GPI[1:0], the active status of a GPI is read from the corresponding bit in GPE1_STS register.

8.10.5 GPO_BLINK—GPO Blink Enable Register

Offset Address:	GPIOBASE +18h	Attribute:	R/W
Default Value:	0004 0000h	Size:	32-bit
Lockable:	No	Power Well:	See bit description

Bit	Description
31:29	Reserved.
28:27	**GPIO Blink (GP_BLINK[n])—**R/W. The setting of these bits will have no effect if the corresponding GPIO is programmed as an input. These bits correspond to GPIO that are in the Resume well and will be reset to their default values by RSMRST# but not by PCIRST#. 0 = The corresponding GPIO will function normally. 1 = If the corresponding GPIO is programmed as an output, the output signal will blink at a rate of approximately once per second. The high and low times have approximately 50% duty cycle. The GP_LVL bit is not altered when this bit is set.
26	Reserved.
25	**GPIO Blink (GP_BLINK[n])—**R/W. The setting of these bits will have no effect if the corresponding GPIO is programmed as an input. These bits correspond to GPIO that are in the Resume well and will be reset to their default values by RSMRST# but not by PCIRST#. 0 = The corresponding GPIO will function normally. 1 = If the corresponding GPIO is programmed as an output, the output signal will blink at a rate of approximately once per second. The high and low times have approximately 50% duty cycle. The GP_LVL bit is not altered when this bit is set.
24:20	Reserved.
19:18	**GPIO Blink (GP_BLINK[n])—**R/W. The setting of these bits will have no effect if the corresponding GPIO is programmed as an input. These bits correspond to GPIO that are in the Core well, and will be reset to their default values by PCIRST#. 0 = The corresponding GPIO will function normally. 1 = If the corresponding GPIO is programmed as an output, the output signal will blink at a rate of approximately once per second. The high and low times have approximately 50% duty cycle. The GP_LVL bit is not altered when this bit is set.
17:0	Reserved.

NOTE:
1. GPIO[18] blinks, by default, immediately after reset. This signal could be connected to an LED to indicate a failed boot (by programming BIOS to clear GP_BLINK[18] after successful POST).

8.10.6 GPI_INV—GPIO Signal Invert Register

Offset Address:	GPIOBASE +2Ch	Attribute:	R/W
Default Value:	00000000h	Size:	32-bit
Lockable:	No	Power Well:	See bit description

Bit	Description
31:14	Reserved.
13:11	**GPIO Signal High/Low Select (GP_INV[n])—**R/W. These bits are used to allow both active-low and active-high inputs to cause SMI# or SCI. The setting of these bits will have no effect if the corresponding GPIO is programmed as an output. These bits are reset to their default values by RSMRST# but not by PCIRST#. 0 = The corresponding GPI_STS bit will be set when the 82801E C-ICH detects the state of the input pin to be high. 1 = The corresponding GPI_STS bit will be set when the 82801E C-ICH detects the state of the input pin to be low.
10:9	Reserved.
8	**GPIO Signal High/Low Select (GP_INV[n])—**R/W. These bits are used to allow both active-low and active-high inputs to cause SMI# or SCI. The setting of these bits will have no effect if the corresponding GPIO is programmed as an output. These bits are reset to their default values by RSMRST# but not by PCIRST#. 0 = The corresponding GPI_STS bit will be set when the 82801E C-ICH detects the state of the input pin to be high. 1 = The corresponding GPI_STS bit will be set when the 82801E C-ICH detects the state of the input pin to be low.
7:6	**GPIO Signal High/Low Select (GP_INV[n])—**R/W. These bits are used to allow both active-low and active-high inputs to cause SMI# or SCI. The setting of these bits will have no effect if the corresponding GPIO is programmed as an output. These bits are reset to their default values by PCIRST#. 0 = The corresponding GPI_STS bit will be set when the 82801E C-ICH detects the state of the input pin to be high. 1 = The corresponding GPI_STS bit will be set when the 82801E C-ICH detects the state of the input pin to be low.
5	Reserved.
4:3	**GPIO Signal High/Low Select (GP_INV[n])—**R/W. These bits are used to allow both active-low and active-high inputs to cause SMI# or SCI. The setting of these bits will have no effect if the corresponding GPIO is programmed as an output. These bits are reset to their default values by PCIRST#. 0 = The corresponding GPI_STS bit will be set when the 82801E C-ICH detects the state of the input pin to be high. 1 = The corresponding GPI_STS bit will be set when the 82801E C-ICH detects the state of the input pin to be low.
2	Reserved.
1:0	**GPIO Signal High/Low Select (GP_INV[n])—**R/W. These bits are used to allow both active-low and active-high inputs to cause SMI# or SCI. The setting of these bits will have no effect if the corresponding GPIO is programmed as an output. These bits are reset to their default values by PCIRST#. 0 = The corresponding GPI_STS bit will be set when the 82801E C-ICH detects the state of the input pin to be high. 1 = The corresponding GPI_STS bit will be set when the 82801E C-ICH detects the state of the input pin to be low.

IDE Controller Registers (D31:F1) 9

9.1 PCI Configuration Registers (IDE—D31:F1)

Note: Registers that are not shown should be treated as Reserved (See “PCI Configuration Map” on page 162 for details).

All of the IDE registers are in the Core well. None of these registers can be locked.

Table 105. PCI Configuration Map (IDE—D31:F1)

Offset	Mnemonic	Register Name/Function	Default	Type
00h–01h	VID	Vendor ID	8086h	RO
02h–03h	DID	Device ID	245Bh	RO
04h–05h	CMD	Command Register	00h	R/W
06h–07h	STS	Device Status	0280h	R/W
08h	RID	Revision ID	See Note 1	RO
09h	PI	Programming Interface	80h	RO
0Ah	SCC	Sub Class Code	01h	RO
0Bh	BCC	Base Class Code	01h	RO
0Dh	MLT	Master Latency Timer	00	RO
0Eh	HTYPE	Header Type	00h	RO
20h–23h	BAR	Base Address Register	00000001h	R/W
2Ch–2Dh	SVID	Subsystem Vendor ID	00h	R/Write-Once
2Eh–2Fh	SID	Subsystem ID	00h	R/Write-Once
40h–41h	IDE_TIMP	Primary IDE Timing	0000h	R/W
42h–43h	ID_TIMS	Secondary IDE Timing	0000h	R/W
44h	SIDETIM	Slave IDE Timing	00h	R/W
48h	SDMAC	Synchronous DMA Control Register	00h	R/W
4Ah–4Bh	SDMATIM	Synchronous DMA Timing Register	0000h	R/W
54h	IDE_CONFIG	IDE I/O Configuration Register	00h	R/W

NOTES:
1. Refer to the Specification Update for the value of the Revision ID Register
2. The 82801E C-ICH IDE controller is not arbitrated as a PCI device; therefore, it does not need a master latency timer.

9.1.1 VID—Vendor ID Register (IDE—D31:F1)

Offset Address:	00–01h	Attribute:	RO
Default Value:	8086h	Size:	16-bit
Lockable:	No	Power Well:	Core

Bit	Description
15:0	**Vendor ID Value.** This is a 16 bit value assigned to Intel. Intel VID = 8086h

9.1.2 DID—Device ID Register (IDE—D31:F1)

Offset Address:	02–03h	Attribute:	RO
Lockable:	No	Size:	16-bit
Default Value:	245Bh	Power Well:	Core

Bit	Description
15:0	**Device ID Value.** This is a 16 bit value assigned to the 82801E C-ICH IDE controller.

9.1.3 CMD—Command Register (IDE—D31:F1)

Address Offset:	04h–05h	Attribute:	RO, R/W
Default Value:	00h	Size:	16 bits

Bit	Description
15:10	Reserved.
9	**Fast Back to Back Enable (FBE)—RO. Reserved as 0.**
8	**SERR# Enable—RO. Reserved as 0.**
7	**Wait Cycle Control—RO. Reserved as 0.**
6	**Parity Error Response—RO. Reserved as 0.**
5	**VGA Palette Snoop—RO. Reserved as 0.**
4	**Postable Memory Write Enable (PMWE)—RO. Reserved as 0.**
3	**Special Cycle Enable (SCE)—RO. Reserved as 0.**
2	**Bus Master Enable (BME)—**R/W. Controls the 82801E C-ICH's ability to act as a PCI master for IDE Bus Master transfers.
1	**Memory Space Enable (MSE)—RO. Reserved as 0.**
0	**I/O Space Enable (IOSE)—**R/W. This bit controls access to the I/O space registers. 0 = Disables access to the Legacy IDE ports (both Primary and Secondary) as well as the Bus Master IO registers. 1 = Enable. Note that the Base Address register for the Bus Master registers should be programmed before this bit is set. **NOTE:** Separate bits are provided (IDE Decode Enable, in the IDE Timing register) to independently disable the Primary or Secondary I/O spaces.

9.1.4 STS—Device Status Register (IDE—D31:F1)

Address Offset: 06–07h
Default Value: 0280h
Attribute: R/WC, RO
Size: 16 bits

Bit	Description
15	**Detected Parity Error (DPE)—RO. Reserved as 0.**
14	**Signaled System Error (SSE)—RO. Reserved as 0.**
13	**Received Master-Abort Status (RMA)—**R/WC. 1 = Bus Master IDE interface function, as a master, generated a master abort. 0 = Cleared by writing a 1 to it.
12	Reserved as 0—RO.
11	**Signaled Target-Abort Status (STA)—**R/WC. 1 = 82801E C-ICH IDE interface function is targeted with a transaction that the 82801E C-ICH terminates with a target abort. 0 = Cleared by writing a 1 to it.
10:9	**DEVSEL# Timing Status (DEVT)—**RO. 01 = Hardwired; however, the 82801E C-ICH does not have a real DEVSEL# signal associated with the IDE unit, so these bits have no effect.
8	**Data Parity Error Detected—RO. Reserved as 0.**
7	**Fast Back-to-Back Capable—RO. Reserved as 1.**
6	**User Definable Features (UDF)—RO. Reserved as 0.**
5	**66 MHz Capable—RO. Reserved as 0.**
4:0	Reserved.

9.1.5 RID—Revision ID Register (HUB-PCI—D30:F0)

Offset Address: 08h
Default Value: See bit description
Attribute: RO
Size: 8 bits

Bit	Description
7:0	**Revision Identification Number**—RO. This 8-bit value indicates the revision number for the 82801E C-ICH IDE controller. Refer to the Specification Update for the value of the Revision ID Register.

9.1.6 PI—Programming Interface (IDE—D31:F1)

Address Offset: 09h
Default Value: 80h
Attribute: RO
Size: 8 bits

Bit	Description
7:0	**Programming Interface Value—**RO. 80h = The 1b in bit 7 indicates that this IDE controller is capable of bus master operation.

9.1.7 SCC—Sub Class Code (IDE—D31:F1)

Address Offset: 0Ah Attribute: RO
Default Value: 01h Size: 8 bits

Bit	Description
7:0	**Sub Class Code—**RO. 01h = IDE device, in the context of a mass storage device.

9.1.8 BCC—Base Class Code (IDE—D31:F1)

Address Offset: 0Bh Attribute: RO
Default Value: 01h Size: 8 bits

Bit	Description
7:0	**Base Class Code—**RO. 01 = Mass storage device

9.1.9 MLT—Master Latency Timer (IDE—D31:F1)

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

Bit	Description
7:0	**Bus Master Latency—**RO. The IDE controller is implemented internally, and is not arbitrated as a PCI device, so it does not need a Master Latency Timer. Hardwired to 00h.

9.1.10 BM_BASE—Bus Master Base Address Register (IDE—D31:F1)

Address Offset: 20h–23h Attribute: R/W
Default Value: 00000001h Size: 32 bits

The Bus Master IDE interface function uses Base Address register 5 to request a 16 byte IO space to provide a software interface to the Bus Master functions. Only 12 bytes are actually used (6 bytes for primary, 6 bytes for secondary). Only bits [15:4] are used to decode the address.

Bit	Description
31:16	Reserved.
15:4	**Base Address—**R/W. Base address of the I/O space (16 consecutive I/O locations).
3:1	Reserved.
0	**Resource Type Indicator (RTE)—**RO. Hardwired to 1, indicating a request for IO space.

9.1.11 IDE_SVID—Subsystem Vendor ID (IDE—D31:F1)

Address Offset:	2Ch–2Dh	Attribute:	R/Write-Once
Default Value:	00h	Size:	16 bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	**Subsystem Vendor ID (SVID)—**R/Write-Once. The SVID register, in combination with the Subsystem ID (SID) register, enables the operating system (OS) to distinguish subsystems from each other. Software (BIOS) sets the value in this register. After that, the value can be read, but subsequent writes to this register have no effect. The value written to this register will also be readable via the corresponding SVID registers for the USB#1, USB#2 and SMBus functions.

9.1.12 IDE_SID—Subsystem ID (IDE—D31:F1)

Address Offset:	2Eh–2Fh	Attribute:	R/Write-Once
Default Value:	00h	Size:	16 bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	**Subsystem ID (SID)—**R/Write-Once. The SID register, in combination with the SVID register, enables the operating system (OS) to distinguish subsystems from each other. Software (BIOS) sets the value in this register. After that, the value can be read, but subsequent writes to this register have no effect. The value written to this register will also be readable via the corresponding SID registers for the USB#1, USB#2 and SMBus functions.

9.1.13 IDE_TIM—IDE Timing Register (IDE—D31:F1)

Address Offset:	Primary: 40–41h Secondary: 42–43h	Attribute:	R/W
Default Value:	0000h	Size:	16 bits

This register controls the timings driven on the IDE cable for PIO and 8237 style DMA transfers. It also controls operation of the buffer for PIO transfers.

Bit	Description
15	**IDE Decode Enable (IDE)—**R/W. Individually enable/disable the Primary or Secondary decode. The IDE I/O Space Enable bit in the Command register must be set in order for this bit to have any effect. Additionally, separate configuration bits are provided (in the IDE I/O Configuration register) to individually disable the primary or secondary IDE interface signals, even if the IDE Decode Enable bit is set. 0 = Disable. 1 = Enables the 82801E C-ICH to decode the associated Command Blocks (1F0h–1F7h for primary, 170h–177h for secondary) and Control Block (3F6h for primary and 376h for secondary).
14	**Drive 1 Timing Register Enable (SITRE)—**R/W. 0 = Use bits 13:12, 9:8 for both drive 0 and drive 1. 1 = Use bits 13:12, 9:8 for drive 0, and use the Slave IDE Timing register for drive 1
13:12	**IORDY Sample Point (ISP).** The setting of these bits determine the number of PCI clocks between IDE IOR#/IOW# assertion and the first IORDY sample point. 00 = 5 clocks 01 = 4 clocks 10 = 3 clocks 11 = Reserved
11:10	Reserved.
9:8	**Recovery Time (RCT)—**R/W. The setting of these bits determines the minimum number of PCI clocks between the last IORDY sample point and the IOR#/IOW# strobe of the next cycle. 00 = 4 clocks 01 = 3 clocks 10 = 2 clocks 11 = 1 clock
7	**Drive 1 DMA Timing Enable (DTE1)—**R/W. 0 = Disable. 1 = Enable the fast timing mode for DMA transfers only for this drive. PIO transfers to the IDE data port will run in compatible timing.
6	**Drive 1 Prefetch/Posting Enable (PPE1)—**R/W. 0 = Disable. 1 = Enable Prefetch and posting to the IDE data port for this drive.
5	**Drive 1 IORDY Sample Point Enable (IE1)—**R/W. 0 = Disable IORDY sampling for this drive. 1 = Enable IORDY sampling for this drive.
4	**Drive 1 Fast Timing Bank (TIME1)—**R/W. 0 = Accesses to the data port will use compatible timings for this drive. 1 = When this bit = 1 and bit 14 = 0, accesses to the data port will use bits 13:12 for the IORDY sample point, and bits 9:8 for the recovery time. When this bit = 1 and bit 14 = 1, accesses to the data port will use the IORDY sample point and recover time specified in the slave IDE timing register.
3	**Drive 0 DMA Timing Enable (DTE0)—**R/W. 0 = Disable. 1 = Enable fast timing mode for DMA transfers only for this drive. PIO transfers to the IDE data port will run in compatible timing.

Bit	Description
2	**Drive 0 Prefetch/Posting Enable (PPE0)—**R/W. 0 = Disable prefetch and posting to the IDE data port for this drive. 1 = Enable prefetch and posting to the IDE data port for this drive.
1	**Drive 0 IORDY Sample Point Enable (IE0)—**R/W. 0 = Disable IORDY sampling is disabled for this drive. 1 = Enable IORDY sampling for this drive.
0	**Drive 0 Fast Timing Bank (TIME0)—**R/W. 0 = Accesses to the data port will use compatible timings for this drive. 1 = Accesses to the data port will use bits 13:12 for the IORDY sample point, and bits 9:8 for the recovery time

9.1.14 SLV_IDETIM—Slave (Drive 1) IDE Timing Register (IDE—D31:F1)

Address Offset: 44h Attribute: R/W
Default Value: 00h Size: 8 bits

Bit	Description
7:6	**Secondary Drive 1 IORDY Sample Point (SISP1)—**R/W. Determines the number of PCI clocks between IDE IOR#/IOW# assertion and the first IORDY sample point, if the access is to drive 1 data port and bit 14 of the IDE timing register for secondary is set. 00 = 5 clocks 01 = 4 clocks 10 = 3 clocks 11 = Reserved
5:4	**Secondary Drive 1 Recovery Time (SRCT1)—**R/W. Determines the minimum number of PCI clocks between the last IORDY sample point and the IOR#/IOW# strobe of the next cycle, if the access is to drive 1 data port and bit 14 of the IDE timing register for secondary is set. 00 = 4 clocks 01 = 3 clocks 10 = 2 clocks 11 = 1 clocks
3:2	**Primary Drive 1 IORDY Sample Point (PISP1)—**R/W. Determines the number of PCI clocks between IOR#/IOW# assertion and the first IORDY sample point, if the access is to drive 1 data port and bit 14 of the IDE timing register for primary is set. 00 = 5 clocks 01 = 4 clocks 10 = 3 clocks 11 = Reserved
1:0	**Primary Drive 1 Recovery Time (PRCT1)—**R/W. Determines the minimum number of PCI clocks between the last IORDY sample point and the IOR#/IOW# strobe of the next cycle, if the access is to drive 1 data port and bit 14 of the IDE timing register for primary is set. 00 = 4 clocks 01 = 3 clocks 10 = 2 clocks 11 = 1 clocks

9.1.15 SDMA_CNT—Synchronous DMA Control Register (IDE—D31:F1)

Address Offset: 48h
Default Value: 00h
Attribute: R/W
Size: 8 bits

Bit	Description
7:4	Reserved.
3	**Secondary Drive 1 Synchronous DMA Mode Enable (SSDE1)—**R/W. 0 = Disable (default). 1 = Enable Synchronous DMA mode for secondary channel drive 1
2	**Secondary Drive 0 Synchronous DMA Mode Enable (SSDE0)—**R/W. 0 = Disable (default). 1 = Enable Synchronous DMA mode for secondary drive 0.
1	**Primary Drive 1 Synchronous DMA Mode Enable (PSDE1)—**R/W. 0 = Disable (default). 1 = Enable Synchronous DMA mode for primary channel drive 1
0	**Primary Drive 0 Synchronous DMA Mode Enable (PSDE0)—**R/W. 0 = Disable (default). 1 = Enable Synchronous DMA mode for primary channel drive 0.

9.1.16 SDMA_TIM—Synchronous DMA Timing Register (IDE—D31:F1)

Address Offset: 4A–4Bh
Default Value: 0000h
Attribute: R/W
Size: 16 bits

Bit	Description		
15:14	Reserved.		
13:12	**Secondary Drive 1 Cycle Time (SCT1)—**R/W. For Ultra ATA mode, the setting of these bits determines the minimum write strobe cycle time (CT). The DMARDY#-to-STOP (RP) time is also determined by the setting of these bits.		
	SCB1 = 0 (33 MHz clk)	**SCB1 = 1 (66 MHz clk)**	**FAST_SCB1 = 1 (133 MHz clk)**
	00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved
	01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks
	10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved
	11 = Reserved	11 = Reserved	11 = Reserved
11:10	Reserved.		
9:8	**Secondary Drive 0 Cycle Time (SCT0)—**R/W. For Ultra ATA mode, the setting of these bits determines the minimum write strobe cycle time (CT). The DMARDY#-to-STOP (RP) time is also determined by the setting of these bits.		
	SCB1 = 0 (33 MHz clk)	**SCB1 = 1 (66 MHz clk)**	**FAST_SCB1 = 1 (133 MHz clk)**
	00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved
	01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks
	10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved
	11 = Reserved	11 = Reserved	11 = Reserved
7:6	Reserved.		

Bit	Description		
5:4	**Primary Drive 1 Cycle Time (PCT1)—**R/W. For Ultra ATA mode, the setting of these bits determines the minimum write strobe cycle time (CT). The DMARDY#-to-STOP (RP) time is also determined by the setting of these bits.		
	PCB1 = 0 (33 MHz clk)	**PCB1 = 1 (66 MHz clk)**	**FAST_PCB1 = 1 (133 MHz clk)**
	00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved
	01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks
	10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved
	11 = Reserved	11 = Reserved	11 = Reserved
3:2	Reserved.		
1:0	**Primary Drive 0 Cycle Time (PCT0)—**R/W. For Ultra ATA mode, the setting of these bits determines the minimum write strobe cycle time (CT). The DMARDY#-to-STOP (RP) time is also determined by the setting of these bits.		
	PCB1 = 0 (33 MHz clk)	**PCB1 = 1 (66 MHz clk)**	**FAST_PCB1 = 1 (133 MHz clk)**
	00 = CT 4 clocks, RP 6 clocks	00 = Reserved	00 = Reserved
	01 = CT 3 clocks, RP 5 clocks	01 = CT 3 clocks, RP 8 clocks	01 = CT 3 clks, RP 16 clks
	10 = CT 2 clocks, RP 4 clocks	10 = CT 2 clocks, RP 8 clocks	10 = Reserved
	11 = Reserved	11 = Reserved	11 = Reserved

9.1.17 IDE_CONFIG—IDE I/O Configuration Register

Address Offset: 54h Attribute: R/W
Default Value: 00h Size: 32 bits

Bit	Description
31:20	Reserved.
19:18	**Secondary IDE Signal Mode (SEC_SIG_MODE)**—R/W. 00 = Normal (Enabled) 01 = Tri-state (Disabled) 10 = Drive low (Disabled) 11 = Reserved These bits are used to control the mode of the Secondary IDE signal pins. These bits should always be set to 00b for systems that do not implement swap bay support.
17:16	**Primary IDE Signal Mode (PRIM_SIG_MODE)**—R/W. 00 = Normal (Enabled) 01 = Tri-state (Disabled) 10 = Drive low (Disabled) 11 = Reserved These bits are used to control the mode of the Primary IDE signal pins. These bits should always be set to 00b for systems that do not implement swap bay support.
15	**Fast Secondary Drive 1 Base Clock (FAST_SCB1)**—R/W. This bit is used in conjuction with the SCT1 bits to enable/disable Ultra ATA/100 timings for the Secondary Slave drive. 0 = Disable Ultra ATA/100 timing for the Secondary Slave drive. 1 = Enable Ultra ATA/100 timing for the Secondary Slave drive (overrides bit 3 in this register).
14	**Fast Secondary Drive 0 Base Clock (FAST_SCB0)**—R/W. This bit is used in conjuction with the SCT0 bits to enable/disable Ultra ATA/100 timings for the Secondary Master drive. 0 = Disable Ultra ATA/100 timing for the Secondary Master drive. 1 = Enable Ultra ATA/100 timing for the Secondary Master drive (overrides bit 2 in this register).

Bit	Description
13	**Fast Primary Drive 1 Base Clock (FAST_PCB1)**—R/W. This bit is used in conjuction with the PCT1 bits to enable/disable Ultra ATA/100 timings for the Primary Slave drive. 0 = Disable Ultra ATA/100 timing for the Primary Slave drive. 1 = Enable Ultra ATA/100 timing for the Primary Slave drive (overrides bit 1 in this register).
12	**Fast Primary Drive 0 Base Clock (FAST_PCB0)**—R/W. This bit is used in conjuction with the PCT0 bits to enable/disable Ultra ATA/100 timings for the Primary Master drive. 0 = Disable Ultra ATA/100 timing for the Primary Master drive. 1 = Enable Ultra ATA/100 timing for the Primary Master drive (overrides bit 0 in this register).
11	Reserved.
10	**Write Buffer PingPong Enable (WR_PingPong_EN)**—R/W. 0 = Disabled. The buffer will behave similar to PIIX4. 1 = Enables the write buffer to be used in a split (ping/pong) manner.
9:8	Reserved.
7	**Secondary Slave Channel Cable Reporting**—R/W. BIOS should program this bit to tell the IDE driver which cable is plugged into the channel. 0 = 40 conductor cable is present. 1 = 80 conductor cable is present.
6	**Secondary Master Channel Cable Reporting—**R/W. Same description as bit 7
5	**Primary Slave Channel Cable Reporting—**R/W. Same description as bit 7
4	**Primary Master Channel Cable Reporting—**R/W. Same description as bit 7
3	**Secondary Drive 1 Base Clock (SCB1)**—R/W. 0 = 33 MHz base clock for Ultra ATA timings. 1 = 66 MHz base clock for Ultra ATA timings.
2	**Secondary Drive 0 Base Clock (SCBO)**—R/W. 0 = 33 MHz base clock for Ultra ATA timings. 1 = 66 MHz base clock for Ultra ATA timings.
1	**Primary Drive 1 Base Clock (PCB1)**—R/W. 0 = 33 MHz base clock for Ultra ATA timings. 1 = 66 MHz base clock for Ultra ATA timings.
0	**Primary Drive 0 Base Clock (PCB0)**—R/W. 0 = 33 MHz base clock for Ultra ATA timings. 1 = 66 MHz base clock for Ultra ATA timings.

9.2 Bus Master IDE I/O Registers (D31:F1)

The bus master IDE function uses 16 bytes of I/O space, allocated via the BMIBA register, located in Device 31:Function 1 Configuration space (offset 20h). All bus master IDE I/O space registers can be accessed as byte, word, or DWord quantities. Reading reserved bits returns an indeterminate, inconsistent value; writes to reserved bits have no affect (but should not be attempted). The description of the I/O registers is shown in Table 106.

Table 106. Bus Master IDE I/O Registers

Offset	Mnemonic	Register	Default	Type
00h	BMICP	Command Register Primary	00h	R/W
01h		Reserved	00h	RO
02h	BMISP	Status Register Primary	00h	R/WC
03h		Reserved	00h	RO
04h–07h	BMIDP	Descriptor Table Pointer Primary	undefined	R/W
08h	BMICS	Command Register Secondary	00h	R/W
09h		Reserved	00h	RO
0Ah	BMISS	Status Register Secondary	00h	R/WC
0Bh		Reserved	00h	RO
0Ch–0Fh	BMIDS	Descriptor Table Pointer Secondary	undefined	R/W

9.2.1 BMIC[P,S]—Bus Master IDE Command Register

Address Offset: Primary: 00h, Secondary: 08h — Attribute: R/W
Default Value: 00h — Size: 8 bits

Bit	Description
7:4	Reserved. Returns 0s.
3	**Read/Write Control (RWC)—**R/W. This bit sets the direction of the bus master transfer: This bit must NOT be changed when the bus master function is active. 0 = Memory reads. 1 = Memory writes
2:1	Reserved. Returns 0s.
0	**Start/Stop Bus Master (START)—**R/W. 1 = Enables bus master operation of the controller. Bus master operation begins when this bit is detected changing from a zero to a one. The controller will transfer data between the IDE device and memory only when this bit is set. Master operation can be halted by writing a '0' to this bit. 0 = All state information is lost when this bit is cleared. Master mode operation cannot be stopped and then resumed. If this bit is reset while bus master operation is still active (i.e., the Bus Master IDE Active bit of the Bus Master IDE Status register for that IDE channel is set) and the drive has not yet finished its data transfer (the Interrupt bit in the Bus Master IDE Status register for that IDE channel is not set), the bus master command is said to be aborted and data transferred from the drive may be discarded instead of being written to system memory. This bit is intended to be reset after the data transfer is completed, as indicated by either the Bus Master IDE Active bit or the Interrupt bit of the Bus Master IDE Status register for that IDE channel being set, or both.

9.2.2 BMIS[P,S]—Bus Master IDE Status Register

Address Offset:	Primary: 02h Secondary: 0Ah	Attribute:	R/WC
Default Value:	00h	Size:	8 bits

Bit	Description
7	Reserved. Returns 0.
6	**Drive 1 DMA Capable**—R/W. 0 = Not Capable. 1 = Capable. Set by device dependent code (BIOS or device driver) to indicate that drive 1 for this channel is capable of DMA transfers, and that the controller has been initialized for optimum performance. The 82801E C-ICH does not use this bit. It is intended for systems that do not attach BMIDE to the PCI bus.
5	**Drive 0 DMA Capable**—R/W. 0 = Not Capable. 1 = Capable. Set by device dependent code (BIOS or device driver) to indicate that drive 0 for this channel is capable of DMA transfers and that the controller has been initialized for optimum performance. The 82801E C-ICH does not use this bit. It is intended for systems that do not attach BMIDE to the PCI bus.
4:3	Reserved. Returns 0s.
2	**Interrupt**—R/WC. Software can use this bit to determine if an IDE device has asserted its interrupt line (IRQ14 for the Primary channel and IRQ15 for Secondary). 1 = Set by the rising edge of the IDE interrupt line, regardless of whether or not the interrupt is masked in the 8259 or the internal I/O APIC. When this bit is read as a one, all data transferred from the drive is visible in system memory. 0 = This bit is cleared by software writing a '1' to the bit position. If this bit is cleared while the interrupt is still active, this bit will remain clear until another assertion edge is detected on the interrupt line.
1	**Error**—R/WC. 1 = This bit is set when the controller encounters a target abort or master abort when transferring data on PCI. 0 = This bit is cleared by software writing a '1' to the bit position.
0	**Bus Master IDE Active (ACT)**—RO. 1 = Set by the 82801E C-ICH when the Start bit is written to the Command register. 0 = This bit is cleared by the 82801E C-ICH when the last transfer for a region is performed, where EOT for that region is set in the region descriptor. It is also cleared by the 82801E C-ICH when the Start bit is cleared in the Command register. When this bit is read as a zero, all data transferred from the drive during the previous bus master command is visible in system memory, unless the bus master command was aborted.

9.2.3 BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register

Address Offset:	Primary: 04h Secondary: 0Ch	Attribute:	R/W
Default Value:	All bits undefined	Size:	32 bits

Bit	Description
31:2	**Base address of Descriptor table (BADDR)**—R/W. Corresponds to A[31:2]. The Descriptor Table must be DWord aligned. The Descriptor Table must not cross a 64 Kbyte boundary in memory.
1:0	Reserved.

USB Controller Registers (D31:F2) 10

10.1 PCI Configuration Registers (D31:F2)

Note: Registers that are not shown should be treated as Reserved (See Section 5.2 for details).

Table 107. PCI Configuration Map (USB—D31:F2)

Offset	Mnemonic	Register Name/Function	Function 2 Default	Type
00h–01h	VID	Vendor ID	8086h	RO
02h–03h	DID	Device ID	2452h	RO
04h–05h	CMD	Command Register	0000h	R/W
06h–07h	STA	Device Status	0280h	R/W
08h	RID	Revision ID	See Note	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	03h	RO
0Bh	BCC	Base Class Code	0Ch	RO
0Dh	MLT	Master Latency Timer	00h	RO
0Eh	HTYPE	Header Type	00h	RO
20h–23h	Base	Base Address Register	00000001h	R/W
2Ch–2Dh	SVID	Subsystem Vendor ID	00h	RO
2Eh–2Fh	SID	Subsystem ID	00h	RO
3Ch	INTR_LN	Interrupt Line	00h	R/W
3Dh	INTR_PN	Interrupt Pin	04h	RO
60h	SB_RELNUM	Serial Bus Release Number	10h	RO
C0h–C1h	USB_LEGKEY	USB Legacy Keyboard/Mouse Control	2000h	R/W
C4h	USB_RES	USB Resume Enable	00h	R/W

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

10.1.1 VID—Vendor Identification Register (USB—D31:F2)

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

Bit	Description
15:0	**Vendor ID Value—**RO. This is a 16-bit value assigned to Intel.

10.1.2 DID—Device Identification Register (USB—D31:F2)

Address Offset: 02–03h Attribute: RO
Default Value: 2452h Size: 16 bits

Bit	Description
15:0	**Device ID Value**—RO**.** This is a 16-bit value assigned to the 82801E C-ICH USB Host Controllers

10.1.3 CMD—Command Register (USB—D31:F2)

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

Bit	Description
15:10	Reserved.
9	**Fast Back to Back Enable (FBE)**—RO. Reserved as 0.
8	**SERR# Enable**—RO. Reserved as 0.
7	**Wait Cycle Control**—RO. Reserved as 0.
6	**Parity Error Response**—RO. Reserved as 0.
5	**VGA Palette Snoop**—RO. Reserved as 0.
4	**Postable Memory Write Enable (PMWE)**—RO. Reserved as 0.
3	**Special Cycle Enable (SCE)**—RO. Reserved as 0.
2	**Bus Master Enable (BME)**—R/W. When set, the 82801E C-ICH can act as a master on the PCI bus for USB transfers.
1	**Memory Space Enable (MSE)**—RO. Reserved as 0.
0	**I/O Space Enable (IOSE)**—R/W. This bit controls access to the I/O space registers. 1 = Enable accesses to the USB I/O registers. The Base Address register for USB should be programmed before this bit is set. 0 = Disable

10.1.4 STA—Device Status Register (USB—D31:F2)

Address Offset: 06–07h Attribute: R/WC
Default Value: 0280h Size: 16 bits

Bit	Description
15:14	Reserved as '00b'. Read Only.
13	**Received Master-Abort Status (RMA)**—R/WC. 1 = USB, as a master, generated a master-abort. 0 = Software clears this bit by writing a 1 to the bit location.
12	Reserved. Always read as 0.
11	**Signaled Target-Abort Status (STA)**—R/WC. 1 = USB function is targeted with a transaction that the 82801E C-ICH terminates with a target abort. 0 = Software clears this bit by writing a 1 to the bit location.
10:9	**DEVSEL# Timing Status (DEVT)—**RO. This 2-bit field defines the timing for DEVSEL# assertion. These read only bits indicate the 82801E C-ICH's DEVSEL# timing when performing a positive decode. 82801E C-ICH generates DEVSEL# with medium timing for USB.
8	**Data Parity Error Detected**. Reserved as 0. Read Only.
7	**Fast Back-to-Back Capable**. Reserved as 1. Read Only.
6	**User Definable Features (UDF)**. Reserved as 0. Read Only.
5	**66 MHz Capable**. Reserved as 0. Read Only.
4:0	Reserved.

10.1.5 RID—Revision Identification Register (USB—D31:F2)

Address Offset: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

Bit	Description
7:0	**Revision Identification.** These bits contain device stepping information and are hardwired to the default value. Refer to the Specification Update for the value of the Revision ID Register.

10.1.6 PI—Programming Interface (USB—D31:F2)

Address Offset: 09h Attribute: RO
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**Programming Interface**—RO. 00h = No specific register level programming interface defined.

10.1.7 SCC—Sub Class Code Register (USB—D31:F2)

Address Offset:	0Ah	Attribute:	RO
Default Value:	03h	Size:	8 bits

Bit	Description
7:0	**Sub Class Code**—RO. 03h = Universal Serial Bus Host Controller.

10.1.8 BCC—Base Class Code Register (USB—D31:F2)

Address Offset:	0Bh	Attribute:	RO
Default Value:	0Ch	Size:	8 bits

Bit	Description
7:0	**Base Class Code**—RO. 0Ch = Serial Bus controller.

10.1.9 MLT—Master Latency Timer Register (USB—D31:F2)

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

Bit	Description
7:0	**Master Latency Timer**—RO. 00h = Because the USB controller is internally implemented with arbitration via Hub Interface (not PCI), it does not need a master latency timer. These bits are fixed at 0.

10.1.10 BASE—Base Address Register (USB—D31:F2)

Address Offset:	20–23h	Attribute:	R/W
Default Value:	00000001h	Size:	32 bits

Bit	Description
31:16	Reserved.
15:5	**Base Address**—R/W. Bits [15:5] correspond to I/O address signals AD [15:5], respectively. This gives 32 bytes of relocatable I/O space.
4:1	Reserved.
0	**Resource Type Indicator (RTE)**—RO. This bit is hardwired to 1 indicating that the base address field in this register maps to I/O space

10.1.11 SVID—Subsystem Vendor ID (USB—D31:F2)

Address Offset:	2Ch–2Dh	Attribute:	RO
Default Value:	00h	Size:	16 bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	**Subsystem Vendor ID (SVID)—**RO. The SVID register, in combination with the Subsystem ID (SID) register, enables the operating system (OS) to distinguish subsystems from each other. The value returned by reads to this register is the same as that which was written by BIOS into the IDE_SVID register.

10.1.12 SID—Subsystem ID (USB—D31:F2)

Address Offset:	2Eh–2Fh	Attribute:	RO
Default Value:	00h	Size:	16 bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	**Subsystem ID (SID)—**R/Write-Once. The SID register, in combination with the SVID register, enables the operating system (OS) to distinguish subsystems from each other. The value returned by reads to this register is the same as that which was written by BIOS into the IDE_SID register.

10.1.13 INTR_LN—Interrupt Line Register (USB—D31:F2)

Address Offset:	3Ch	Attribute:	R/W
Default Value:	00h	Size:	8 bits

Bit	Description
7:0	**Interrupt Line**—R/W. This data is not used by the 82801E C-ICH. It is to communicate to software the interrupt line that the interrupt pin is connected to.

10.1.14 INTR_PN—Interrupt Pin Register (USB—D31:F2)

Address Offset:	3Dh	Attribute:	RO
Default Value:	04h	Size:	8 bits

Bit	Description
7:3	Reserved.
2:0	**Interrupt Pin**—RO. The value of 04h in Function 2 indicates that the 82801E C-ICH will drive PIRQD# as its interrupt line for USB Controller 0 (ports 0 and 1).

10.1.15 SB_RELNUM—Serial Bus Release Number Register (USB—D31:F2)

Address Offset:	60h	Attribute:	RO
Default Value:	10h	Size:	8 bits

Bit	Description
7:0	**Serial Bus Release Number**—RO. 10h = Indicates that the USB controller is compliant with the USB specification release 1.0.

10.1.16 USB_LEGKEY—USB Legacy Keyboard/Mouse Control Register (USB—D31:F2)

Address Offset:	C0–C1	Attribute:	R/W, R/WC, RO
Default Value:	2000h	Size:	16 bits

Bit	Description
15	**SMI Caused by End of Pass-through (SMIBYENDPS)**—R/WC. Indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 7, this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. 1 = Event Occurred 0 = Software clears this bit by writing a 1 to the bit location.
14	Reserved.
13	**PCI Interrupt Enable (USBPIRQEN)**—R/W. Used to prevent the USB controller from generating an interrupt due to transactions on its ports. Note that it will probably be configured to generate an SMI using bit 4 of this register. Default to 1 for compatibility with older USB software. 1 = Enable 0 = Disable
12	**SMI Caused by USB Interrupt (SMIBYUSB)**—RO. Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the bit 4, this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. 1 = Event Occurred 0 = Software should clear the IRQ via the USB controller. Writing a 1 to this bit will have no effect.
11	**SMI Caused by Port 64 Write (TRAPBY64W)**—R/WC. Indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 3, this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. 1 = Event Occurred 0 = Software clears this bit by writing a 1 to the bit location.
10	**SMI Caused by Port 64 Read (TRAPBY64R)**—R/WC. Indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 2, this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. 1 = Event Occurred 0 = Software clears this bit by writing a 1 to the bit location.
9	**SMI Caused by Port 60 Write (TRAPBY60W)**—R/WC. Indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 1, this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. 1 = Event Occurred 0 = Software clears this bit by writing a 1 to the bit location.

Bit	Description
8	**SMI Caused by Port 60 Read (TRAPBY60R)**—R/WC. Indicates if the event occurred. Note that even if the corresponding enable bit is not set in bit 0, this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. 1 = Event Occurred 0 = Software clears this bit by writing a 1 to the bit location.
7	**SMI at End of Pass-through Enable (SMIATENDPS)**—R/W. May need to cause SMI at the end of a pass-through. Can occur if an SMI is generated in the middle of a pass through, and needs to be serviced later. 1 = Enable 0 = Disable
6	**Pass Through State (PSTATE)**—RO. 1 = Indicates that the state machine is in the middle of an A20GATE pass-through sequence. 0 = If software needs to reset this bit, it should set bit 5 to 0.
5	**A20Gate Pass-Through Enable (A20PASSEN)**—R/W. 1 = Allows A20GATE sequence Pass-Through function. SMI# will not be generated, even if the various enable bits are set. 0 = Disable
4	**SMI on USB IRQ Enable (USBSMIEN)**—R/W. 1 = USB interrupt will cause an SMI event. 0 = Disable
3	**SMI on Port 64 Writes Enable (64WEN)**—R/W. 1 = A write to port 64h will cause an SMI event. 0 = Disable
2	**SMI on Port 64 Reads Enable (64REN)**—R/W. 1 = A read to port 64h will cause an SMI event. 0 = Disable
1	**SMI on Port 60 Writes Enable (60WEN)**—R/W. 1 = A write to port 60h will cause an SMI event. 0 = Disable
0	**SMI on Port 60 Reads Enable (60REN)**—R/W. 1 = A read to port 60h will cause an SMI event. 0 = Disable

10.1.17 USB_RES—USB Resume Enable Register (USB—D31:F2)

Address Offset:	C4h	Attribute:	R/W
Default Value:	00h	Size:	8 bits

Bit	Description
7:2	Reserved.
1	**PORT1EN**—R/W. Enable the USB controller to respond to wakeup events on this port. This applies to port 1. 1 = The USB controller will monitor this port for remote wakeup and connect/disconnect events. 0 = The USB controller will not look at this port for a wakeup event.
0	**PORT0EN**—R/W. Enable the USB controller to respond to wakeup events on this port. This applies to port 0. 1 = The USB controller will monitor this port for remote wakeup and connect/disconnect events. 0 = The USB controller will not look at this port for a wakeup event.

10.2 USB I/O Registers

Some of the read/write register bits that deal with changing the state of the USB hub ports function such that on read back they reflect the current state of the port, and not necessarily the state of the last write to the register. This allows the software to poll the state of the port and wait until it is in the proper state before proceeding. A Host Controller Reset, Global Reset, or Port Reset will immediately terminate a transfer on the affected ports and disable the port. This affects the USBCMD register, bit [4] and the PORTSC registers, bits [12,6,2]. See individual bit descriptions for more detail.

Table 108. USB I/O Registers

Offset	Mnemonic	Register	Default	Type
00–01h	USBCMD	USB Command Register	0000h	R/W*
02–03h	USBSTS	USB Status Register	0020h	R/WC
04–05h	USBINTR	USB Interrupt Enable	0000h	R/W
06–07h	FRNUM	USB Frame Number	0000h	R/W (see Note 1)
08–0Bh	FRBASEADD	USB Frame List Base Address	Undefined	R/W
0Ch	SOFMOD	USB Start of Frame Modify	40h	R/W
0D–0Fh	—	Reserved	0	RO
10–11h	PORTSC0	Port 0 Status/Control	0080h	R/WC (see Note 1)
12–13h	PORTSC1	Port 1 Status/Control	0080h	R/WC (see Note 1)
14–17h	—	Reserved	0	RO
18h	LOOPDATA	Loop Back Test Data	00h	RO

NOTES:
1. These registers are Word writable only. Byte writes to these registers have unpredictable effects.

10.2.1 USBCMD—USB Command Register

I/O Offset:	Base + (00–01h)	Attribute:	R/W
Default Value:	0000h	Size:	16 bits

The Command Register indicates the command to be executed by the serial bus host controller. Writing to the register causes a command to be executed. Table 109 provides additional information on the operation of the Run/Stop and Debug bits.

Bit	Description
15:7	Reserved.
8	**Loop Back Test Mode**—R/W. 82801E C-ICH is in loop back test mode. When both ports are connected together, a write to one port will be seen on the other port and the data will be stored in I/O offset 18h. 0 = Disable loop back test mode.
7	**Max Packet (MAXP)**—R/W. This bit selects the maximum packet size that can be used for full speed bandwidth reclamation at the end of a frame. This value is used by the Host Controller to determine whether it should initiate another transaction based on the time remaining in the SOF counter. Use of reclamation packets larger than the programmed size will cause a Babble error if executed during the critical window at frame end. The Babble error results in the offending endpoint being stalled. Software is responsible for ensuring that any packet which could be executed under bandwidth reclamation be within this size limit. 1 = 64 bytes 0 = 32 bytes
6	**Configure Flag (CF)**—R/W. This bit has no effect on the hardware. It is provided only as a semaphore service for software. 1 = HCD software sets this bit as the last action in its process of configuring the Host Controller. 0 = Indicates that software has not completed host controller configuration.
5	**Software Debug (SWDBG)**—R/W. The SWDBG bit must only be manipulated when the controller is in the stopped state. This can be determined by checking the HCHalted bit in the USBSTS register. 1 = Debug mode. In SW Debug mode, the Host Controller clears the Run/Stop bit after the completion of each USB transaction. The next transaction is executed when software sets the Run/Stop bit back to 1. 0 = Normal Mode.
4	**NOTE: Suspend not supported. This bit must be set to 0.** **Force Global Resume (FGR)**—R/W. 1 = Host Controller sends the Global Resume signal on the USB, and sets this bit to 1 when a resume event (connect, disconnect, or K-state) is detected while in global suspend mode. 0 = Software resets this bit to 0 after 20 ms has elapsed to stop sending the Global Resume signal. At that time all USB devices should be ready for bus activity. The 1 to 0 transition causes the port to send a low speed EOP signal. This bit will remain a 1 until the EOP has completed.
3	**NOTE: Suspend not supported. This bit must be set to 0.** **Enter Global Suspend Mode (EGSM)**—R/W. 1 = Host Controller enters the Global Suspend mode. No USB transactions occur during this time. The Host Controller is able to receive resume signals from USB and interrupt the system. Software must ensure that the Run/Stop bit (bit 0) is cleared prior to setting this bit. 0 = Software resets this bit to 0 to come out of Global Suspend mode. Software writes this bit to 0 at the same time that Force Global Resume (bit 4) is written to 0 or after writing bit 4 to 0.

Bit	Description
2	**Global Reset (GRESET)**—R/W. 1 = Global Reset. The Host Controller sends the global reset signal on the USB and then resets all its logic, including the internal hub registers. The hub registers are reset to their power on state. Chip Hardware Reset has the same effect as Global Reset (bit 2), except that the Host Controller does not send the Global Reset on USB. 0 = This bit is reset by the software after a minimum of 10 ms has elapsed as specified in Chapter 7 of the USB Specification.
1	**Host Controller Reset (HCRESET)**—R/W. The effects of HCRESET on Hub registers are slightly different from Chip Hardware Reset and Global USB Reset. The HCRESET affects bits [8,3:0] of the Port Status and Control Register (PORTSC) of each port. HCRESET resets the state machines of the Host Controller including the Connect/Disconnect state machine (one for each port). When the Connect/Disconnect state machine is reset, the output that signals connect/disconnect are negated to 0, effectively signaling a disconnect, even if a device is attached to the port. This virtual disconnect causes the port to be disabled. This disconnect and disabling of the port causes bit 1 (connect status change) and bit 3 (port enable/disable change) of the PORTSC to get set. The disconnect also causes bit 8 of PORTSC to reset. About 64 bit times after HCRESET goes to 0, the connect and low-speed detect will take place, and bits 0 and 8 of the PORTSC will change accordingly. 1 = Reset. When this bit is set, the Host Controller module resets its internal timers, counters, state machines, etc. to their initial value. Any transaction currently in progress on USB is immediately terminated. 0 = Reset by the Host Controller when the reset process is complete.
0	**Run/Stop (RS)**—R/W. When set to 1, the 82801E C-ICH proceeds with execution of the schedule. The 82801E C-ICH continues execution as long as this bit is set. When this bit is cleared, the 82801E C-ICH completes the current transaction on the USB and then halts. The HC Halted bit in the status register indicates when the Host Controller has finished the transaction and has entered the stopped state. The Host Controller clears this bit when the following fatal errors occur: consistency check failure, PCI Bus errors. 1 = Run 0 = Stop

Table 109. Run/Stop, Debug Bit Interaction SWDBG (Bit 5), Run/Stop (Bit 0) Operation

SWDBG (Bit 5)	Run/Stop (Bit 0)	Description
0	0	If executing a command, the Host Controller completes the command and then stops. The 1.0 ms frame counter is reset and command list execution resumes from start of frame using the frame list pointer selected by the current value in the FRNUM register. (While Run/Stop=0, the FRNUM register can be reprogrammed).
0	1	Execution of the command list resumes from Start Of Frame using the frame list pointer selected by the current value in the FRNUM register. The Host Controller remains running until the Run/Stop bit is cleared (by software or hardware).
1	0	If executing a command, the Host Controller completes the command and then stops and the 1.0 ms frame counter is frozen at its current value. All status are preserved. The Host Controller begins execution of the command list from where it left off when the Run/Stop bit is set.
1	1	Execution of the command list resumes from where the previous execution stopped. The Run/Stop bit is set to 0 by the Host Controller when a TD is being fetched. This causes the Host Controller to stop again after the execution of the TD (single step). When the Host Controller has completed execution, the HC Halted bit in the Status Register is set.

When the USB Host Controller is in Software Debug Mode (USBCMD Register bit 5=1), the single stepping software debug operation is as follows:

To Enter Software Debug Mode:

1. The Host Controller Driver (HCD) puts the Host Controller in the Stop state by setting the Run/Stop bit to 0.
2. The HCD puts the Host Controller in Debug Mode by setting the SWDBG bit to 1.
3. The HCD sets up the correct command list and Start Of Frame value for starting point in the Frame List Single Step Loop.
4. The HCD sets the Run/Stop bit to 1.
5. The Host Controller executes next active TD, sets the Run/Stop bit to 0, and stops.
6. The HCD reads the USBCMD register to check if the single step execution is completed (HCHalted=1).
7. The HCD checks results of TD execution. Go to step 4 to execute the next TD or step 8 to end Software Debug mode.
8. The HCD ends Software Debug mode by setting the SWDBG bit to 0.
9. The HCD sets up normal command list and Frame List table.
10. The HCD sets the Run/Stop bit to 1 to resume normal schedule execution.

In Software Debug mode, when the Run/Stop bit is set, the Host Controller starts. When a valid TD is found, the Run/Stop bit is reset. When the TD is finished, the HCHalted bit in the USBSTS register (bit 5) is set.

The SW Debug mode skips over inactive TDs and only halts after an active TD has been executed. When the last active TD in a frame has been executed, the Host Controller waits until the next SOF is sent and then fetches the first TD of the next frame before halting.

This HCHalted bit can also be used outside of Software Debug mode to indicate when the Host Controller has detected the Run/Stop bit and has completed the current transaction. Outside of the Software Debug mode, setting the Run/Stop bit to 0 always resets the SOF counter so that when the Run/Stop bit is set the Host Controller starts over again from the frame list location pointed to by the Frame List Index (see FRNUM Register description) rather than continuing where it stopped.

10.2.2 USBSTA—USB Status Register

I/O Offset:	Base + (02–03h)	Attribute:	R/WC
Default Value:	0020h	Size:	16 bits

This register indicates pending interrupts and various states of the Host Controller. The status resulting from a transaction on the serial bus is not indicated in this register. Software sets a bit to 0 in this register by writing a 1 to it.

Bit	Description
15:6	Reserved.
5	**Host Controller Halted**—RO. 1 = The Host Controller has stopped executing as a result of the Run/Stop bit being set to 0, either by software or by the Host Controller hardware (debug mode or an internal error). Default. 0 = Host Controller executing normally.
4	**Host Controller Process Error**—R/WC. 1 = The Host Controller has detected a fatal error. This indicates that the Host Controller suffered a consistency check failure while processing a Transfer Descriptor. An example of a consistency check failure would be finding an illegal PID field while processing the packet header portion of the TD. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further schedule execution. A hardware interrupt is generated to the system. 0 = Software resets this bit to 0 by writing a 1 to the bit position.
3	**Host System Error**—R/WC. 1 = A serious error occurred during a host system access involving the Host Controller module. In a PCI system, conditions that set this bit to 1 include PCI Parity error, PCI Master Abort, and PCI Target Abort. When this error occurs, the Host Controller clears the Run/Stop bit in the Command register to prevent further execution of the scheduled TDs. A hardware interrupt is generated to the system. 0 = Software resets this bit to 0 by writing a 1 to the bit position.
2	**NOTE: Suspend not supported. This bit must be set to 0.** **Resume Detect (RSM_DET)**—R/WC. 1 = The Host Controller received a “RESUME” signal from a USB device. This is only valid if the Host Controller is in a global suspend state (bit 3 of Command register = 1). 0 = Software resets this bit to 0 by writing a 1 to the bit position.
1	**USB Error Interrupt**—R/WC. 1 = Completion of a USB transaction resulted in an error condition (e.g., error counter underflow). If the TD on which the error interrupt occurred also had its IOC bit set, both this bit and Bit 0 are set. 0 = Software resets this bit to 0 by writing a 1 to the bit position.
0	**USB Interrupt (USBINT)**—R/WC. 1 = The Host Controller sets this bit when the cause of an interrupt is a completion of a USB transaction whose Transfer Descriptor had its IOC bit set. Also set when a short packet is detected (actual length field in TD is less than maximum length field in TD), and short packet detection is enabled in that TD. 0 = Software resets this bit to 0 by writing a 1 to the bit position.

10.2.3 USBINTR—Interrupt Enable Register

I/O Offset:	Base + (04–05h)	Attribute:	R/W
Default Value:	0000h	Size:	16 bits

This register enables and disables reporting of the corresponding interrupt to the software. When a bit is set and the corresponding interrupt is active, an interrupt is generated to the host. Fatal errors (Host Controller Processor Error-bit 4, USBSTS Register) cannot be disabled by the host controller. Interrupt sources that are disabled in this register still appear in the Status Register to allow the software to poll for events.

Bit	Description
15:4	Reserved.
3	**Short Packet Interrupt Enable**—R/W. 1 = Enabled. 0 = Disabled.
2	**Interrupt On Complete (IOC) Enable**—R/W. 1 = Enabled. 0 = Disabled.
1	**Resume Interrupt Enable**—R/W. 1 = Enabled. 0 = Disabled.
0	**Time-out/CRC Interrupt Enable**—R/W. 1 = Enabled. 0 = Disabled.

10.2.4 FRNUM—Frame Number Register

I/O Offset:	Base + (06–07h)	Attribute:	R/W (Writes must be Word Writes)
Default Value:	0000h	Size:	16 bits

Bits [10:0] of this register contain the current frame number which is included in the frame SOF packet. This register reflects the count value of the internal frame number counter. Bits [9:0] are used to select a particular entry in the Frame List during scheduled execution. This register is updated at the end of each frame time.

This register must be written as a word. Byte writes are not supported. This register cannot be written unless the Host Controller is in the STOPPED state as indicated by the HCHalted bit (USBSTS register). A write to this register while the Run/Stop bit is set (USBCMD register) is ignored.

Bit	Description
15:11	Reserved.
10:0	**Frame List Current Index/Frame Number**—R/W. Provides the frame number in the SOF Frame. The value in this register increments at the end of each time frame (approximately every 1 ms). In addition, bits [9:0] are used for the Frame List current index and correspond to memory address signals [11:2].

10.2.5 FRBASEADD—Frame List Base Address

I/O Offset:	Base + (08–0Bh)	Attribute:	R/W
Default Value:	Undefined	Size:	32 bits

This 32-bit register contains the beginning address of the Frame List in the system memory. HCD loads this register prior to starting the schedule execution by the Host Controller. When written, only the upper 20 bits are used. The lower 12 bits are written as zero (4-Kbyte alignment). The contents of this register are combined with the frame number counter to enable the Host Controller to step through the Frame List in sequence. The two least significant bits are always 00. This requires DWord alignment for all list entries. This configuration supports 1024 Frame List entries.

Bit	Description
31:12	**Base Address**—R/W. These bits correspond to memory address signals [31:12], respectively.
11:0	Reserved.

10.2.6 SOFMOD—Start of Frame Modify Register

I/O Offset:	Base + (0Ch)	Attribute:	R/W
Default Value:	40h	Size:	8 bits

This 1-byte register is used to modify the value used in the generation of SOF timing on the USB. Only the 7 least significant bits are used. When a new value is written into these 7 bits, the SOF timing of the next frame will be adjusted. This feature can be used to adjust out any offset from the clock source that generates the clock that drives the SOF counter. This register can also be used to maintain real time synchronization with the rest of the system so that all devices have the same sense of real time. Using this register, the frame length can be adjusted across the full range required by the USB specification. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. It may be reprogrammed by USB system software at any time. Its value will take effect from the beginning of the next frame. This register is reset upon a Host Controller Reset or Global Reset. Software must maintain a copy of its value for reprogramming if necessary.

Bit	Description
7	Reserved.
6:0	**SOF Timing Value**—R/W. Guidelines for the modification of frame time are contained in Chapter 7 of the USB Specification. The SOF cycle time (number of SOF counter clock periods to generate a SOF frame length) is equal to 11936 + value in this field. The default value is decimal 64 which gives a SOF cycle time of 12000. For a 12 MHz SOF counter clock input, this produces a 1 ms Frame period. The following table indicates what SOF Timing Value to program into this field for a certain frame period. **Frame Length (# 12 MHz Clocks) (decimal)** — **SOF Reg. Value (decimal)** 11936 — 0 11937 — 1 . — . . — . 11999 — 63 12000 — 64 12001 — 65 . — . . — . 12062 — 126 12063 — 127

10.2.7 PORTSC[0,1]—Port Status and Control Register

I/O Offset: Port 0: Base + (10-11h) Port 1: Base + (12-13h) Attribute:R/W (Word writes only)
Default Value: 0080h Size:16 bits

After a Power-up reset, Global reset, or Host Controller reset, the initial conditions of a port are: no device connected, Port disabled, and the bus line status is 00 (single-ended zero).

Bit	Description
15:13	Reserved—RO.
12	**Suspend**—R/W. This bit should not be written to a 1 if global suspend is active (bit 3=1 in the USBCMD register). Bit 2 and bit 12 of this register define the hub states as follows: **Bits [12,2]** **Hub State** X0 Disable 01 Enable 11 Suspend When in suspend state, downstream propagation of data is blocked on this port, except for single-ended 0 resets (global reset and port reset). The blocking occurs at the end of the current transaction, if a transaction was in progress when this bit was written to 1. In the suspend state, the port is sensitive to resume detection. Note that the bit status does not change until the port is suspended and that there may be a delay in suspending a port if there is a transaction currently in progress on the USB. 1 = Port in suspend state. 0 = Port not in suspend state. **NOTE:** Normally, if a transaction is in progress when this bit is set, the port will be suspended when the current transaction completes. However, in the case of a specific error condition (out transaction with babble), the 82801E C-ICH may issue a start-of-frame, and then suspend the port.
11	**Overcurrent Indicator**—R/WC. Set by hardware 1 = Overcurrent pin has gone from inactive to active on this port. 0 = Software clears this bit by writing a 1 to the bit position.
10	**Overcurrent Active**—RO. This bit is set and cleared by hardware. 1 = Indicates that the overcurrent pin is active (low). 0 = Indicates that the overcurrent pin is inactive (high).
9	**Port Reset**—RO. 1 = Port is in Reset. When set, the port is disabled and sends the USB Reset signaling. 0 = Port is not in Reset.
8	**Low Speed Device Attached (LS)**—RO. Writes have no effect. 1 = Low speed device is attached to this port. 0 = Full speed device is attached.
7	Reserved—RO. Always read as 1.
6	**Resume Detect (RSM_DET)—**R/W. Software sets this bit to a 1 to drive resume signaling. The Host Controller sets this bit to a 1 if a J-to-K transition is detected for at least 32 microseconds while the port is in the Suspend state. The 82801E C-ICH then reflects the K-state back onto the bus as long as the bit remains a 1 and the port is still in the suspend state (bit 12,2 are 11). Writing a 0 (from 1) causes the port to send a low speed EOP. This bit will remain a 1 until the EOP has completed. 1 = Resume detected/driven on port. 0 = No resume (K-state) detected/driven on port.
5:4	**Line Status**—RO. These bits reflect the D+ (bit 4) and D- (bit 5) signals lines' logical levels. These bits are used for fault detect and recovery as well as for USB diagnostics. This field is updated at EOF2 time (See Chapter 11 of the USB Specification).

Bit	Description
3	**Port Enable/Disable Change**—R/WC. For the root hub, this bit gets set only when a port is disabled due to disconnect on that port or due to the appropriate conditions existing at the EOF2 point (See Chapter 11 of the USB Specification). 1 = Port enabled/disabled status has changed. 0 = No change. Software clears this bit by writing a 1 to the bit location.
2	**Port Enabled/Disabled (PORT_EN)—**R/W. Ports can be enabled by host software only. Ports can be disabled by either a fault condition (disconnect event or other fault condition) or by host software. Note that the bit status does not change until the port state actually changes and that there may be a delay in disabling or enabling a port if there is a transaction currently in progress on the USB. 1 = Enable. 0 = Disable.
1	**Connect Status Change**—R/WC. Indicates that a change has occurred in the port's Current Connect Status (see bit 0). The hub device sets this bit for any changes to the port device connect status, even if system software has not cleared a connect status change. If, for example, the insertion status changes twice before system software has cleared the changed condition, hub hardware will be setting" an already-set bit (i.e., the bit will remain set). However, the hub transfers the change bit only once when the Host Controller requests a data transfer to the Status Change endpoint. System software is responsible for determining state change history in such a case. 1 = Change in Current Connect Status. 0 = No change. Software clears this bit by writing a 1 to the bit location.
0	**Current Connect Status**—RO. This value reflects the current state of the port, and may not correspond directly to the event that caused the Connect Status Change bit (Bit 1) to be set. 1 = Device is present on port. 0 = No device is present.

SMBus Controller Registers (D31:F3) 11

11.1 PCI Configuration Registers (SMBUS—D31:F3)

Table 110. PCI Configuration Registers (SMBUS—D31:F3)

Offset	Mnemonic	Register Name/Function	Default Value	Attribute
00h–01h	VID	Vendor ID	8086h	RO
02h–03h	DID	Device ID	2453h	RO
04h–05h	CMD	Command Register	0000h	RO, R/W
06h–07h	STA	Device Status	0280h	RO, R/WC
08h	RID	Revision ID	See Note	RO
09h	PI	Programming Interface	00h	RO
0Ah	SCC	Sub Class Code	05h	RO
0Bh	BCC	Base Class Code	0Ch	RO
20h–23h	SMB_BASE	SMBus Base Address Register	00000001h	R/W
2Ch–2Dh	SVID	Subsystem Vendor ID	00h	RO
2Eh–2Fh	SID	Subsystem ID	00h	RO
3Ch	INTR_LN	Interrupt Line	00h	R/W
3Dh	INTR_PN	Interrupt Pin	02h	RO
40h	HOSTC	Host Configuration	00h	R/W

NOTE: See the *Intel 82801E Communications I/O Controller Hub (C-ICH) Specification Update* (order number 273645) for Revision ID information.

Note: Registers that are not shown should be treated as Reserved (See "PCI Configuration Map" on page 162 for details).

11.1.1 VID—Vendor Identification Register (SMBUS—D31:F3)

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

Bit	Description
15:0	**Vendor ID Value—**RO. This is a 16 bit value assigned to Intel

11.1.2 DID—Device Identification Register (SMBUS—D31:F3)

Address: 02–03h Attributes: RO
Default Value: 2453h Size: 16 bits

Bit	Description
15:0	**Device ID value—**RO.

11.1.3 CMD—Command Register (SMBUS—D31:F3)

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

Bit	Description
15:10	Reserved.
9	**Fast Back to Back Enable (FBE)**—RO. Reserved as 0.
8	**SERR# Enable (SERREN)**—RO. Reserved as 0.
7	**Wait Cycle Control (WCC)**—RO. Reserved as 0.
6	**Parity Error Response (PER)**—RO. Reserved as 0.
5	**VGA Palette Snoop (VPS)**—RO. Reserved as 0.
4	**Postable Memory Write Enable (PMWE)**—RO. Reserved as 0.
3	**Special Cycle Enable (SCE)**—RO. Reserved as 0.
2	**Bus Master Enable (BME)**—RO. Reserved as 0.
1	**Memory Space Enable (MSE)**—RO. Reserved as 0.
0	**I/O Space Enable (IOSE)—**R/W. 0 = Disable. 1 = Enables access to the SM Bus I/O space registers as defined by the Base Address Register.

11.1.4 STA—Device Status Register (SMBUS—D31:F3)

Address: 06–07h Attributes: RO, R/WC
Default Value: 0280h Size: 16 bits

Bit	Description
15	**Detected Parity Error (DPE)**—RO. Reserved as 0.
14	**Signaled System Error (SSE)**—RO. Reserved as 0.
13	**Received Master Abort (RMA)**—RO. Reserved as 0.
12	**Received Target Abort (RTA)**—RO. Reserved as 0.
11	**Signaled Target-Abort Status (STA)**—R/WC. 1 = Function is targeted with a transaction that the 82801E C-ICH terminates with a target abort. 0 = Software resets STA to 0 by writing a 1 to this bit location.
10:9	**DEVSEL# Timing Status (DEVT)**—RO. This 2-bit field defines the timing for DEVSEL# assertion for positive decode. 01 = Medium timing.
8	**Data Parity Error Detected**—RO. Reserved as 0.
7	**Fast Back-to-Back Capable**—RO. Reserved as 1.
6	**User Definable Features (UDF)**—RO. Reserved as 0.
5	**66 MHz Capable**—RO. Reserved as 0.
4:0	Reserved.

11.1.5 RID—Revision ID Register (SMBUS—D31:F3)

Offset Address: 08h Attribute: RO
Default Value: See bit description Size: 8 bits

Bit	Description
7:0	**Revision Identification Number.** 8-bit value that indicates the revision number for the SMBus Controller. Refer to the Specification Update for the value of the Revision ID Register

11.1.6 PI—Programming Interface (SMBUS—D31:F3)

Address Offset: 09h Attribute: RO
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**Programming Interface Value—**RO. 80h = The 1b in bit 7 indicates that this IDE controller is capable of bus master operation.

11.1.7 SCC—Sub Class Code Register (SMBUS—D31:F3)

Address Offset: 0Ah Attributes: RO
Default Value: 05h Size: 8 bits

Bit	Description
7:0	**Sub Class Code**—RO. 05h = SM Bus serial controller

11.1.8 BCC—Base Class Code Register (SMBUS—D31:F3)

Address Offset: 0Bh Attributes: RO
Default Value: 0Ch Size: 8 bits

Bit	Description
7:0	**Base Class Code**—RO. 0Ch = Serial controller.

11.1.9 SMB_BASE—SMBus Base Address Register (SMBUS—D31:F3)

Address Offset: 20–23h Attribute: R/W
Default Value: 00000001h Size: 32-bits

Bit	Description
31:16	Reserved.
15:4	**Base Address—**R/W. Provides the 16-bit system I/O base address for the 82801E C-ICH SMB logic.
3:1	Reserved.
0	**IO Space Indicator**—RO. This read-only bit is always 1, indicating that the SMB logic is I/O mapped.

11.1.10 SVID—Subsystem Vendor ID (SMBUS—D31:F2/F4)

Address Offset: 2Ch–2Dh Attribute: RO
Default Value: 00h Size: 16 bits
Lockable: No Power Well: Core

Bit	Description
15:0	**Subsystem Vendor ID (SVID)—**RO. The SVID register, in combination with the Subsystem ID (SID) register, enables the operating system (OS) to distinguish subsystems from each other. The value returned by reads to this register is the same as that which was written by BIOS into the IDE_SVID register.

11.1.11 SID—Subsystem ID (SMBUS—D31:F2/F4)

Address Offset:	2Eh–2Fh	Attribute:	RO
Default Value:	00h	Size:	16 bits
Lockable:	No	Power Well:	Core

Bit	Description
15:0	**Subsystem ID (SID)—**R/Write-Once. The SID register, in combination with the SVID register, enables the operating system (OS) to distinguish subsystems from each other. The value returned by reads to this register is the same as that which was written by BIOS into the IDE_SID register.

11.1.12 INTR_LN—Interrupt Line Register (SMBUS—D31:F3)

Address Offset:	3Ch	Attributes:	R/W
Default Value:	00h	Size:	8 bits

Bit	Description
7:0	**Interrupt line—**R/W. This data is not used by the 82801E C-ICH. It is to communicate to software the interrupt line that the interrupt pin is connected to PIRQB#.

11.1.13 INTR_PN—Interrupt Pin Register (SMBUS—D31:F3)

Address Offset:	3Dh	Attributes:	RO
Default Value:	02h	Size:	8 bits

Bit	Description
7:0	**Interrupt PIN—**RO. 02h = Indicates that the 82801E C-ICH SMBus Controller will drive PIRQB# as its interrupt line.

11.1.14 HOSTC—Host Configuration Register (SMBUS—D31:F3)

Address Offset:	40h	Attribute:	R/W
Default Value:	00h	Size:	8 bits

Bit	Description
7:3	Reserved.
2	**I^2C Enable (I2C_EN)—**R/W. 0 = SMBus behavior. 1 = The 82801E C-ICH is enabled to communicate with I^2C devices. This will change the formatting of some commands.
1	**SMBus to SMI Enable (SMB_SMI_EN)—**R/W. 0 = SMBus interrupts will not generate an SMI#. 1 = Any source of an SMB interrupt will instead be routed to generate an SMI#. This bit will only take effect if the INTREN bit is set in I/O space.This bit needs to be set for SMBALERT# to be enabled.
0	**SMBus Host Enable (HST_EN)**—R/W. 0 = Disable the SMBus Host Controller. 1 = Enable. The SMB Host Controller interface is enabled to execute commands. The INTREN bit needs to be enabled for the SMB Host Controller to interrupt or SMI#. Note that the SMB Host Controller will not respond to any new requests until all interrupt requests have been serviced.

11.2 SMBus I/O Registers

Table 111. SMB I/O Registers

Offset	Mnemonic	Register Name/Function	Default	Access
00h	HST_STS	Host Status	00h	R/W
02h	HST_CNT	Host Control	00h	R/W
03h	HST_CMD	Host Command	00h	R/W
04h	XMIT_SLVA	Transmit Slave Address	00h	R/W
05h	HST_D0	Host Data 0	00h	R/W
06h	HST_D1	Host Data 1	00h	R/W
07h	BLOCK_DB	Block Data Byte	00h	R/W
08h	—	Reserved	00h	RO
09h	RCV_SLVA	Receive Slave Address	44h	R/W
0Ah	SLV_DATA	Slave Data	0000h	R/W
0Bh–0Dh	—	Reserved	00h	RO
0Eh	SMLINK_PIN_CTL	SMLINK Pin Control	See Register Description	R/W
0Fh	SMBUS_PIN_CTL	SMbus Pin Control	See Register Description	R/W

11.2.1 HST_STS—Host Status Register

Register Offset:	00h	Attribute:	R/WC
Default Value:	00h	Size:	8-bits

All status bits are set by hardware and cleared by the software writing a one to the particular bit position. Writing a zero to any bit position has no effect.

Bit	Description
7	**Byte Done Status (BYTE_DONE_STA)—**R/WC. 1 = The 82801E C-ICH has received a byte (for Block Read commands) or if it has completed transmission of a byte (for Block Write commands). This bit will be set even on the last byte of the transfer. It will not be set when transmission is due to the Alert On LAN* heartbeat. 0 = Cleared by writing a 1 to the bit position. **NOTE:** The 82801E C-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 bytes or more. The 82801E C-ICH will 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. 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 if interrupts are used or not. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.
6	**In Use Status (INUSE_STA)—**R/WC (special). This bit is used as semaphore among various independent software threads that may need to use the **82801E C-ICH**'s SMBus logic and has no other effect on Hardware. 0 = After a full PCI reset, a read to this bit returns a 0. 1 = After the first read, subsequent reads will return a 1. A write of a 1 to this bit will reset the next read value to 0. Writing a 0 to this bit has no effect. Software can poll this bit until it reads a 0, and will then own the usage of the host controller.
5	**SMBus Alert Status (SMBALERT_STA)—**R/WC. 0 = Interrupt or SMI# was not generated by SMBALERT#. 1 = The source of the interrupt or SMI# was the SMBALERT# signal. This bit is only cleared by software writing a 1 to the bit position or by RSMRST# going low. If the signal is programmed as a GPIO, then this bit will never be set.
4	**Interrupt/SMI# was Failed Bus Transaction (FAILED)—**R/WC. 1 = The source of the interrupt or SMI# was a failed bus transaction. This bit is set in response to the KILL bit being set to terminate the host transaction. 0 = Cleared by writing a 1 to the bit position.
3	**Bus Error (BUS_ERR)**—R/WC. 1 = The source of the interrupt of SMI# was a transaction collision. 0 = Cleared by writing a 1 to the bit position.

Bit	Description
2	**Device Error (DEV_ERR)**—R/WC. 1 = The source of the interrupt or SMI# was due to one of the following: • Illegal Command Field, • Unclaimed Cycle (host initiated), • Host Device Time-out Error.] 0 = Software resets this bit by writing a 1 to this location. The 82801E C-ICH will then deassert the interrupt or SMI#.
1	**Interrupt/SMI# was Successful Completion (INTR)—**R/WC (special). This bit can only be set by termination of a command. INTR is not dependent on the INTREN bit of the Host Controller Register (offset 02h); it is only dependent on the termination of the command. If the INTREN bit is not set, the INTR bit will be set, although the interrupt will not be generated. Software can poll the INTR bit in this non-interrupt case. 1 =The source of the interrupt or SMI# was the successful completion of its last command. 0 =Software resets this bit by writing 1 to this location. The 82801E C-ICH then deasserts the interrupt or SMI#.
0	**Host Busy (HOST_BUSY)—**RO. 1 = Indicates that the 82801E C-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 Register. The Block Data Byte register can be accessed when this bit is set only when the SMB_CMD bits in the Host Control register are programmed for Block command or I^2C Read command. This is necessary in order to check the BYTE_DONE_STS bit. 0 = Cleared by the 82801E C-ICH when the current transaction is completed.

11.2.2 HST_CNT—Host Control Register

Register Offset: 02h Attribute: R/W
Default Value: 00h Size: 8-bits

Bit	Description
7	Reserved.
6	**START—**WO. 1 = Writing a 1 to this bit initiates the command described in the SMB_CMD field. All registers should be setup prior to writing a 1 to this bit position. 0 = This bit will always return 0 on reads. The HOST_BUSY bit in the Host Status register (offset 00h) can be used to identify when the 82801E C-ICH has finished the command.
5	**LAST BYTE—**WO. This bit is used for Block Read commands. 1 = Software sets this bit to indicate that the next byte will be the last byte to be received for the block. This causes the 82801E C-ICH to send a NACK (instead of an ACK) after receiving the last byte. **NOTE:** Once the SECOND_TO_STS bit in TCO2_STS register (Device 31, Function 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 82801E C-ICH from running some of the SM Bus commands (Block Read/Write, I^2C Read, Block I^2C Write).

Bit	Description
4:2	**SMBus Command (SMB_CMD)—**R/W. The bit encoding below indicates which command the 82801E C-ICH is to perform. If enabled, the 82801E C-ICH generates an interrupt or SMI# when the command has completed. If the value is for a non-supported or reserved command, the 82801E C-ICH sets the device error (DEV_ERR) status bit and generates an interrupt when the START bit is set. The 82801E C-ICH performs no command and does not operate until DEV_ERR is cleared. 000 = Quick: The slave address and read/write value (bit 0) are stored in the transmit slave address register. 001 = Byte: This command uses the transmit slave address and command registers. Bit 0 of the slave address register determines if this is a read or write command. 010 = Byte Data: This command uses the transmit slave address, command, and DATA0 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, the DATA0 register will contain the read data. 011 = Word Data: This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, after the command completes, the DATA0 and DATA1 registers will contain the read data. 100 = Process Call: This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. After the command completes, the DATA0 and DATA1 registers will contain the read data. 101 = Block: This command uses the transmit slave address, command, DATA0 registers, and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block reads, the count is received and stored in the DATA0 register. Bit 0 of the slave address register selects if this is a read or write command. For writes, data is retrieved from the first n (where n is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register. 110 = I^2C Read: This command uses the transmit slave address, command, DATA0, DATA1 registers, and the Block Data Byte register. The read data is stored in the Block Data Byte register. The 82801E C-ICH will continue reading data until the NAK is received. 111 = Reserved
1	**KILL—**R/W. 1 = When set, kills the current host transaction taking place, sets the FAILED status bit, and asserts the interrupt (or SMI#). This bit, once set, must be cleared by software to allow the SMBus Host Controller to function normally. 0 = Normal SMBus Host Controller functionality.
0	**INTREN—**R/W. 1 = Enable the generation of an interrupt or SMI# upon the completion of the command. 0 = Disable.

11.2.3 HST_CMD—Host Command Register

Register Offset: 03h Attribute: R/W
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**Host Command**—R/W. This eight bit field is transmitted by the host controller in the command field of the SMBus protocol during the execution of any command.

11.2.4 XMIT_SLVA—Transmit Slave Address Register

Register Offset: 04h Attribute: R/W
Default Value: 00h Size: 8 bits

This register is transmitted by the host controller in the slave address field of the SMBus protocol.

Bit	Description
7:1	**ADDRESS—**R/W. 7-bit address of the targeted slave.
0	**Read/Write Select—**R/W. Direction of the host transfer. 0 = Write 1 = Read

11.2.5 HST_D0—Data 0 Register

Register Offset: 05h Attribute: R/W
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**DATA0/COUNT—**R/W. This field contains the eight bit data sent in the DATA0 field of the SMBus protocol. For block write commands, this register reflects the number of bytes to transfer. This register should be programmed to a value between 1 and 32 for block counts. A count of 0 or a count above 32 will result in unpredictable behavior. The host controller does not check or log illegal block counts. **NOTE:** The 82801E C-ICH uses the HST_D0 register (Dev 31, Func 3, Offset 05h) as the byte count register instead of depending on the LAST_BYTE bit in the Host Control register (Dev 31, Func 3, Offset 02h:[bit-5]) to end the transaction. The transaction will stop prematurely if HST_D0 contains a number smaller than the intended transaction. There is no workaround for 10-bit addressing of I^2C devices. Designers can use the SM Bus read command for 7-bit addressing of I^2C devices. See the *Intel® 82801E Communications I/O Controller Hub (C-ICH) Specification Update* for information about steppings affected by this issue.

11.2.6 HST_D1—Data 1 Register

Register Offset: 06h Attribute: R/W
Default Value: 00h Size: 8 bits

Bit	Description
7:0	**DATA1—**R/W. This eight bit register is transmitted in the DATA1 field of the SMBus protocol during the execution of any command.

11.2.7 BLOCK_DB—Block Data Byte Register

Register Offset:	07h	Attribute:	R/W
Default Value:	00h	Size:	8 bits

Bit	Description
7:0	**Block Data Byte—**R/W. For Block Writes, software writes the first byte to this register as part of the setup for this command. After the 82801E C-ICH has sent the Address, Command, and Byte Count fields, it will send the byte in the Block Data Byte register. After the byte has been sent, the 82801E C-ICH sets the BYTE_DONE_STS bit in the Host Status register. If there are more bytes to send, the software then writes in the next byte to the Block Data Byte register and software also clears the BYTE_DONE_STS bit. The 82801E C-ICH then sends the next byte. During the time from when a byte has been transmitted to when the next byte has been loaded, the 82801E C-ICH inserts wait-states on the SMBus/I^2C. A similar process will be used for Block Reads. After receiving the byte count (which goes in the DATA 0 register), the first “data byte” goes in the Block Data Byte register and the 82801E C-ICH generates an SMI# or interrupt (depending on configuration). The interrupt or SMI# handler then reads the byte and clears the BYTE_DONE_STS bit. This frees room for the next byte. During the time from when a byte is read to when the BYTE_DONE_STS bit is cleared, the 82801E C-ICH inserts wait-states on the SMBus/I^2C.

11.2.8 RCV_SLVA—Receive Slave Address Register

Register Offset:	09h	Attribute:	R/W
Default Value:	44h	Size:	8 bits
Lockable:	No	Power Well:	Resume

Bit	Description
7	Reserved.
6:0	**SLAVE_ADDR—**R/W. This field is the slave address that the 82801E C-ICH decodes for read and write cycles. The default is not 0 so the SMBus Slave Interface can respond even before the processor comes up (or if the processor is dead). This register is cleared by RSMRST#, but not by PCIRST#. **NOTE:** When an external micro-controller accesses the SM Bus slave interface over the SM Link, a translation in the address is needed to accommodate the least significant bit used for read/ write control. For example, the 82801E C-ICH slave address (RCV/SLVA) is left at 44h (default), the external micro-controller would use an address of 88h/89h (write/read).

11.2.9 SLV_DATA—Receive Slave Data Register

Register Offset:	0Ah	Attribute:	RO
Default Value:	0000h	Size:	16 bits
Lockable:	No	Power Well:	Resume

This register contains the 16-bit data value written by the external SMBus master. The CPU can then read the value from this register. This register is reset by RSMRST#, but not PCIRST#.

Bit	Description
15:8	**DATA_MSG1:** Data Message Byte 1—RO. See Section 3.17.5 for a discussion of this field.
7:0	**DATA_MSG0:** Data Message Byte 0—RO. See Section 3.17.5 for a discussion of this field.

11.2.10 SMLINK_PIN_CTL—SMLINK Pin Control Register

Register Offset:	0Eh	Attribute:	Read/Write
Default Value:	See Below	Size:	8 bits

Note: This register is in the resume well and is reset by RSMRST#.

Bit	Description
7:3	Reserved.
2	**SMLINK Clock Pin Control (SMLINK_CLK_CTL)—**R/W. 1 = No functional impact on the SMLINK[0] pin. (default) 0 = 82801E C-ICH will drive the SMLINK[0] pin low, independent of the what the other SMLINK logic would otherwise indicate for the SMLINK[0] pin.
1	**SMLINK[1] Pin Current Status (SMLINK[1]_CUR_STA)—**RO. This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMLINK[1] pin. This allows software to read the current state of the pin. 1 = SMLINK[1] pin is high 0 = SMLINK[1] pin is low
0	**SMLINK[0] Pin Current Status (SMLINK[0]_CUR_STA)—**RO. This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMLINK[0] pin. This allows software to read the current state of the pin. 1 = SMLINK[0] pin is high 0 = SMLINK[0] pin is low

11.2.11 SMBUS_PIN_CTL—SMBus Pin Control Register

Register Offset:	0Fh	Attribute:	Read/Write
Default Value:	See Below	Size:	8 bits

Note: This register is in the resume well and is reset by RSMRST#.

Bit	Description
7:3	Reserved.
2	**SMBCLK Pin Control (SMBCLK_CTL)**—R/W. 1 = No functional impact on the SMBCLK pin. (default) 0 = 82801E C-ICH drives the SMBCLK pin low, independent of the what the other SMB logic would otherwise indicate for the SMBCLK pin.
1	**SMBDATA Pin Current Status (SMBDATA_CUR_STA)**—RO. This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMBDATA pin. This allows software to read the current state of the pin. 1 = SMBDATA pin is high 0 = SMBDATA pin is low
0	**SMBCLK Pin Current Status (SMBCLK_CUR_STA)**—RO. This read-only bit has a default value that is dependent on an external signal level. This pin returns the value on the SMBCLK pin. This allows software to read the current state of the pin. 1 = SMBCLK pin is high 0 = SMBCLK pin is low

Serial I/O Unit 12

The SIU is similar to currently available Super I/O controllers. It is specifically designed for integration into the 82801E C-ICH. It is connected externally via the LPC bus and consists of two UARTS, a Serial Interrupt Controller and the LPC interface.

Important: **The integrated Serial I/O unit's LPC bus is NOT internally connected; it is routed to external pins. The SIU LPC bus pins must be connected to the 82801E C-ICH LPC bus pins. Connect SIU_LAD[3:0] to LAD[3:0]; SIU_LFRAME# to LFRAME#; and SIU_LDRQ# to LDRQ*x*. Refer to the design guide for more information.**

12.1 Features

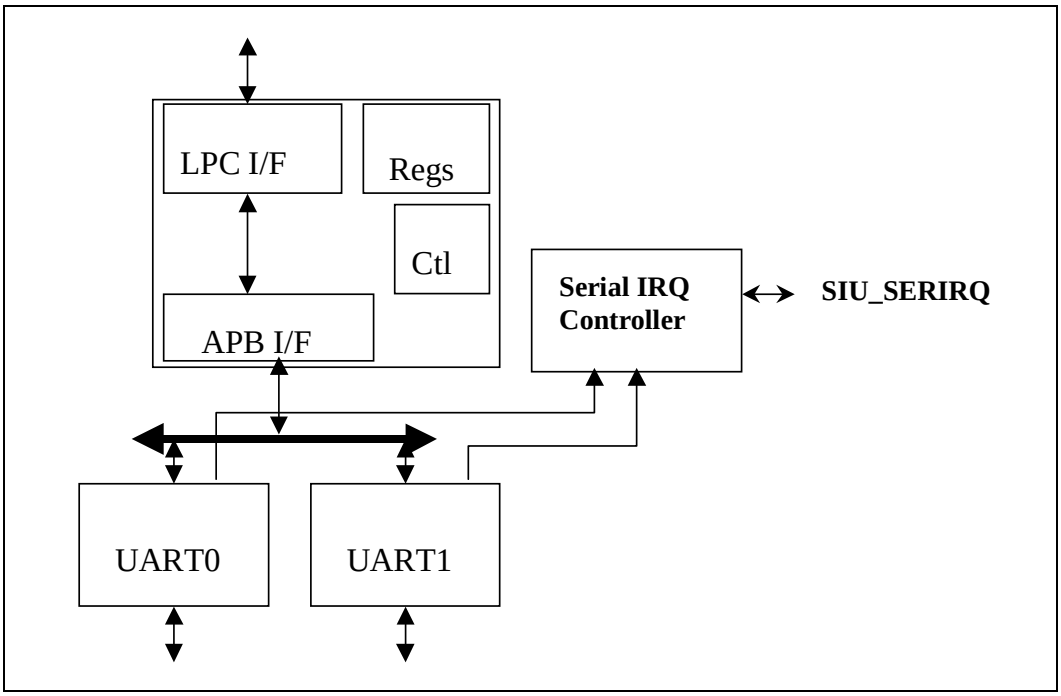
LPC Interface

- Multiplexed Command, Address and Data Bus
- 8-Bit I/O Transfers
- 8-Bit DMA Transfers
- 16-Bit Address Qualification for I/O transactions
- Serial IRQ Interface Compatible with Serialized IRQ Support for PCI Systems

Serial Port

- Two Full Function 16550 Compatible Serial Ports
- Configurable I/O addresses and interrupts
- 16-Byte FIFOs
- Supports up to 115Kbps
- Programmable Baud Rate Generator
- Modem Control Circuitry
- 14.7456 MHz and 48 MHz supported for UART baud clock input

Figure 28. SIU Block Diagram



12.2 Pin Description

12.2.1 Universal Asynchronous Receive And Transmit (UART0, UART1)

Table 112. Universal Asynchronous Receive And Transmit (UART0, UART1) (Sheet 1 of 2)

Signal Name	Type	Description
UART_CLK	I	Input clock to the SIU. This clock is passed to the baud clock generation logic of each UART in the SIU.
SIU0_RXD, SIU1_RXD	I	SERIAL INPUTs for UART0 and UART1: Serial data input from device pin to the receive port.
SIU0_TXD, SIU1_TXD	O	SERIAL OUTPUT for UART0 and UART1: Serial data output to the communication peripheral/modem or data set. Upon reset, the TXD pins will be set to MARKING condition (logic ‘1’ state).

Table 112. Universal Asynchronous Receive And Transmit (UART0, UART1) (Sheet 2 of 2)

Signal Name	Type	Description
SIU0_CTS#, SIU1_CTS#	I	CLEAR TO SEND: Active low, this pin indicates that data can be exchanged between the 82801E C-ICH and the external interface. These pins have no effect on the transmitter. **NOTE:** These pins could be used as Modem Status Input whose condition can be tested by the processor by reading bit 4 (CTS) of the Modem Status register (MSR). Bit 4 is the complement of the CTS# signal. Bit 0 (DCTS) of the MSR indicates whether the CTS# input has changed state since the previous reading of the MSR. When the CTS bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU0_DSR#, SIU1_DSR#	I	DATA SET READY for UART0 and UART1: Active low, this pin indicates that the external agent is ready to communicate with 82801E C-ICH UARTs. These pins have no effect on the transmitter. **NOTE:** These pins could be used as Modem Status Input whose condition can be tested by the processor by reading bit 5 (DSR) of the Modem Status register. Bit 5 is the complement of the DSR# signal. Bit 1 (DDSR) of the Modem status register (MSR) indicates whether the DSR# input has changed state since the previous reading of the MSR. When the DSR bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU0_DCD#, SIU1_DCD#	I	DATA CARRIER DETECT for UART0 and UART1: Active low, this pin indicates that data carrier has been detected by the external agent. **NOTE:** These pins are Modem Status Input whose condition can be tested by the processor by reading bit 7 (DCD) of the Modem Status register (MSR). Bit 7 is the complement of the DCD# signal. Bit 3 (DDCD) of the MSR indicates whether the DCD# input has changed state since the previous reading of the MSR. When the DCD bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU0_RI# SIU1_RI#	I	RING INDICATOR for UART0 and UART1: Active low, this pin indicates that a telephone ringing signal has been received by the external agent. **NOTE:** These pins are Modem Status Input whose condition can be tested by the processor by reading bit 6 (RI) of the Modem Status register (MSR). Bit 6 is the complement of the RI# signal. Bit 2 (TERI) of the MSR indicates whether the DCD# input has changed state since the previous reading of the MSR. When the RI bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU0_DTR#, SIU1_DTR#	O	DATA TERMINAL READY for UART0 and UART1: When low these pins informs the modem or data set that 82801E C-ICH UART0 and UART1 are ready to establish a communication link. The DTR#x(x=0,1) output signals can be set to an active low by programming the DTRx (x-0,1) (bit0) of the Modem control register to a logic ‘1’. A Reset operation sets this signal to its inactive state (logic ‘1’). LOOP mode operation holds this signal in its inactive state.
SIU0_RTS#, SIU1_RTS#	O	REQUEST TO SEND for UART0 and UART1: When low these pins informs the modem or data set that 82801E C-ICH UART0 and UART1 are ready to establish a communication link. The RTS#x(x=0,1) output signals can be set to an active low by programming the RTSx (x-0,1) (bit1) of the Modem control register to a logic ‘1’. A Reset operation sets this signal to its inactive state (logic ‘1’). LOOP mode operation holds this signal in its inactive state.

12.2.2 SIU LPC Interface

Table 113. SIU Interface

Signal Name	Type	Description
SIU_LCLK	I	**SIU LPC clock input to SIU:** 33 MHz LPC clock.
SIU_RESET#	I	**SIU RESET:** This signal should be tied to PCI RESET
SIU_LAD[3:0]	I/O	**SIU LPC Multiplexed Command, Address, Data:** Internal pull-ups are provided.
SIU_LFRAME#	I	**SIU LPC Frame:** Indicates the start of an LPC cycle, or an abort.
SIU_LDRQ#	O	**SIU LPC Serial DMA/Master Request Output:** Used by SIU devices to indicate a DMA request. **NOTE:** These signals have weak internal pull-up resistors to avoid external glue.
SIU_SERIRQ	I/O	**SIU Serial IRQ input:** This pin receives the serial interrupt protocol from external devices. Pull up if unused.

12.3 Functional Description

12.3.1 Host Processor Interface (LPC)

The host processor communicates with the SIU via the LPC bus. Access is through a series of read/write registers and accomplished through I/O cycles or DMA transfers. All registers are 8 bits wide. The SIU registers include global configuration space and device specific regions accessed by setting the Logical Device Number in the SIU Configuration Register 07H (SCR7). See Table 114, “Address Map” on page 334 below.

Table 114. Address Map

Address	Block Name	Logical Device
04Eh	Configuration Index	
04Fh	Configuration Data	
Base+(0h-7h)	Serial Port Com 1	04H
Base+(0h-7h)	Serial Port Com 2	05H

See Section 12.7, “Configuration” on page 358 for configuration register descriptions and setting the base address.

12.4 LPC Interface

The LPC interface is used to control all the logical blocks on the SIU. LPC bus signals use PCI 33 MHz electrical signal characteristics. Refer to the *Low Pin Count (LPC) Interface Specification Rev 1.0.*

12.4.1 LPC Cycles

The following cycle types are supported by the LPC protocol.

Table 115. Supported LPC Cycle Types

Cycle Type	Transfer Size
I/O Write	1 Byte
I/O Read	1 Byte
DMA Write	1 Byte
DMA Read	1 Byte

The SIU ignores cycles that it does not support.

12.4.1.1 I/O Read and Write Cycles

The SIU is the target for I/O cycles. I/O cycles are initiated by the host for register or FIFO accesses and will generally have minimal Sync times.

Data transfers are assumed to be exactly 1-byte. If the CPU requested a 16- or 32-bit transfer, the host must break it up into 8-bit transfers.

See the *Low Pin Count (LPC) Interface Specification* for the sequence of cycles for the I/O Read and Write cycles.

12.4.1.2 DMA Read and Write Cycles

DMA read cycles involve the transfer of data from the host (main memory) to the SIU. DMA write cycles involve the transfer of data from the SIU to the host (main memory). Data will be coming from or going to a FIFO and will have minimal Sync times.

See the *Low Pin Count (LPC) Interface Specification* for the field definitions and the sequence of the DMA Read and Write cycles.

12.4.1.3 DMA Protocol

DMA on the LPC bus is handled through the use of the SIU_LDRQ# from the SIU and special encodings on LAD[3:0] from the host.

See the *Low Pin Count (LPC) Interface Specification* for DMA protocol (request, acknowledge, terminal count).

12.4.1.4 DMA Arbitration

The SIU does not have to arbitrate internally, even though it supports more than one DMA channel. When more than one DMA requests is pending, it sends one request out, then the other.

Priority and arbitration for DMA channels is performed at the 82801E C-ICH DMA controller.

12.4.2 Reset Policy

The following rules govern the reset policy:

SIU_RESET# is tied to the PCI bus reset.

When SIU_RESET# goes active (low):

- The host drives the LFRAME# signal high, tristates the LAD[3:0] signals, and ignores the LDRQ# signal.
- The SIU ignores LFRAME#, tristates the LAD[3:0] pins and drives the SIU_LDRQ# signal inactive (high).

Note: LPC bus signals from SIU are tied to primary LPC interface external to the 82801E C-ICH device. Host LPC and SIU LPC names are used interchangeably throughout.

12.4.3 LPC Transfers

12.4.3.1 I/O Transfers

These will generally be used for register or FIFO accesses, and will generally have minimal Sync times. The minimum number of wait-states between bytes is 1. Data transfers are assumed to be exactly 1 byte. The host is responsible for breaking up larger data transfers into 8 bit cycles.

Table 116. I/O Sync Bits Description

Bits	Indication
0000	Sync Achieved with no error.
0101	Indicates that Sync not Achieved yet, but the part is driving the bus.
0110	Indicates that Sync not Achieved yet, but the part is driving the bus, and expect long Sync
1010	Special Case: peripheral indicating errors.

12.4.3.2 DMA Transfers

DMA on LPC is handled through the use of the LDRQ# lines from peripherals and special encodings on LAD[3:0] from the host. Channels 0 - 3 are 8 bit channels. Channels 5 - 7 are 16 bit channels. Channel 4 is reserved as a generic bus master request (see “DMA Operation (D31:F0)” on page 58).

SIU_LDRQ# is synchronous with SIU_LCLK. The SIU uses the following serial encoding sequence:

- Starts the sequence by asserting SIU_LDRQ# low (start bit). SIU_LDRQ# is high during idle conditions.

- The next 3 bits contain the encoded DMA channel number (MSB first).
- The next bit (ACT) indicates whether the request for the indicated DMA channel is active or inactive. The ACT bit will be a 1(high) to indicate if it is active and 0 (low) if it is inactive. The case where ACT is low will be rare, and is only used to indicate that a previous request for that channel is being abandoned.
- After the active/inactive indication, the SIU_LDRQ# signal must go high for at least 1 clock. After that one clock, SIU_LDRQ# signal can be brought low to the next encoding sequence.

If another DMA channel also needs to request a transfer, another sequence can be sent on SIU_LDRQ#. For example, if an encoded request is sent for channel 0, and then channel 2 needs a transfer before the cycle for channel 0 is run on the interface, the peripheral can send the encoded request for channel 2. This allows multiple DMA agents behind an I/O device to request use of the LPC interface, and the I/O device does not need to self-arbitrate before sending the message.

DMA requests can be de-asserted in two fashions: on error conditions by sending an LDRQ# message with the 'ACT' bit set to '0', or normally through a SYNC field during the DMA transfer.

Note: Software can initiate a FIFO flush to a UART transmit or receive FIFO (bit 2 and bit 1 in FCR) to abandon the respective DMA request. The abandoned request will generate an LDRQ# message with the ACT bit cleared to 0.

There may be some special cases where the SIU desires to abandon a DMA transfer. In these cases, the SIU wishes to stop further DMA activity. It may do so by sending an SIU_LDRQ# message with the ACT bit as '0'. However, since the DMA request was seen by the host, there is no guarantee that the cycle hasn't been granted and will shortly run on LPC. Therefore, the SIU must take into account that a DMA cycle may still occur and should not respond to this cycle, in which case the host will abort it.

12.5 Logical Device 4 & 5: Serial Ports (UARTs)

This section describes the Universal Asynchronous Receiver/Transmitter (UART) serial port used for the two UARTs integrated into the SIU. The UART can be controlled via DMA or programmed I/O. The basic programming model is the same for both UARTs with the only difference being the Logical Device Number assigned to each.

12.5.1 Overview

The serial port consists of a UART which supports all the functions of a standard 16550 UART including hardware flow control interface.

The UART performs serial-to-parallel conversion on data characters received from a peripheral device or a modem and parallel-to-serial conversion on data characters received from the processor. The processor can read the complete status of the UART at any time during the functional operation. Available status information includes the type and condition of the transfer operations being performed by the UART and any error conditions (parity, overrun, framing, or break interrupt).

The serial port can operate in either FIFO or non-FIFO mode. In FIFO mode, a 16-byte transmit FIFO holds data from the processor to be transmitted on the serial link and a 16-byte Receive FIFO buffers data from the serial link until read by the processor.

Each UART includes a programmable baud rate generator which is capable of dividing the baud clock input by divisors of 1 to (2^{16} -1) and producing a 16X clock to drive the internal transmitter and receiver logic. Each UART has complete modem control capability and a processor interrupt system. Interrupts can be programmed to the user's requirements, minimizing the computing required to handle the communications link. Each UART can operate in a polled or an interrupt driven environment as configured by software.

The baud rate generator input is a function of the UART_CLK and a configurable pre-divide of 1, 8, or 26. See also SIU Configuration (address 29h) in Table 124. The output of the baud rate generator is 16 times the baud rate.

Table 117. Baud Rate Examples

Desired Baud Rate	UART Clock Frequency Divisor		
	1.8432 MHz	14.7456 MHz	48 MHz
1200	96	768	2500
2400	48	384	1250
4800	24	192	625
7200	16	128	417
9600	12	96	312
19200	6	48	156
38400	3	24	78
56000	2	16	54
115200	1	8	26

12.5.1.1 UART Feature List

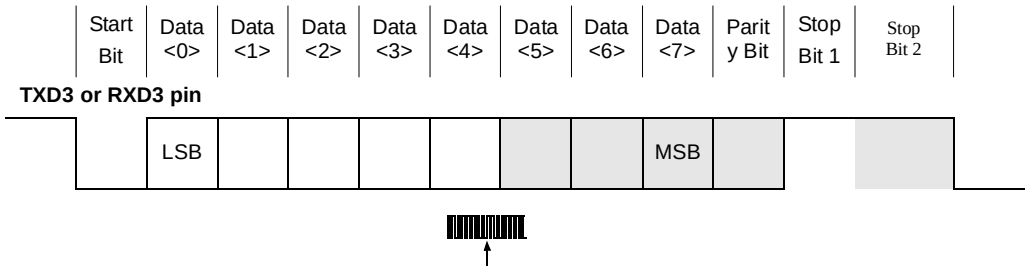
- Functionally compatible with National Semiconductor's PC16550D
- Adds or deletes standard asynchronous communications bits (start, stop, and parity) to or from the serial data
- Independently controlled transmit, receive, line status and data set interrupts
- Programmable baud rate generator allows division of clock by 1 to (2^{16} -1) and generates an internal 16X clock
- Modem control functions (CTS#, RTS#, DSR#, DTR#, RI#, and DCD#)
- Fully programmable serial-interface characteristics:
- 5, 6, 7 or 8-bit characters
- Even, odd, or no parity detection
- 1, 1-1/2, or 2 stop bit generation
- Baud rate generation (up to 115kbps)
- False start bit detection
- 16-byte Receive FIFO
- Complete status reporting capability
- Line break generation and detection

- Internal diagnostic capabilities include:
- Loopback controls for communications link fault isolation
- Break, parity, overrun, and framing error simulation
- Fully prioritized interrupt system controls
- Two separate DMA requests for transmit and receive data services

12.5.2 UART Operational Description

The format of a UART data frame is shown in Figure 29.

Figure 29. Example UART Data Frame



Receive data sample counter frequency = 16x bit frequency, each bit is sampled three times in the middle. Shaded bits are optional and can be programmed by users.

Each data frame is between 7 bits and 12 bits long depending on the size of data programmed, if parity is enabled and if two stop bits is selected. The frame begins with a start bit that is represented by a high to low transition. Next, either 5 to 8 bits of data are transmitted, beginning with the least significant bit. An optional parity bit follows, which is set if even parity is enabled and an odd number of ones exist within the data byte, or if odd parity is enabled and the data byte contains an even number of ones. The data frame ends with one, one and a half or two stop bits as programmed by the user, which is represented by one or two successive bit periods of a logic one.

The unit is disabled upon reset, the user needs to enable the unit by setting bit 6 of Interrupt Enable Register. When the unit is enabled, the receiver starts looking for the start bit of a frame; the transmitter starts transmitting data to the transmit data pin if there is data available in the transmit FIFO. Transmit data can be written to the FIFO before the unit is enabled. When the unit is disabled, the transmitter/receiver finishes the current byte being transmitted/received if it is in the middle of transmitting/receiving a byte and stops transmitting/receiving more data.

An SIU_RESET# to the SIU will force the internal register and output signals on the serial port to the values listed in Table 118.

Table 118. SIU Signal Reset States

Register/Signal	Reset Control	Reset State
Interrupt Enable Register	RESET	All bits are low.
Interrupt ID Register	RESET	Bit 0 is forced high. Bits 1-3 and 6-7 are forced low. Bits 4-5 are permanently low.
Line Control Register	RESET	All bits are forced low.
Line Status Register	RESET	Bits 0-4,7 are forced low. Bits 5 and 6 are forced high.
Modem Control Register	RESET	Bits 0,1,2,3,4 are forced low. Bits 5,6,7 are permanently low.
Modem Status Register	RESET/Modem signal, read MSR for bits 3-0.	Low
SIU0_TXD SIU1_TXD	RESET	High
DMA_TXRQ	RESET	Low
DMA_RXRQ	RESET/ clear LINE STATUS REG	Low
SIU_SERIRQ	RESET/ clear LINE STATUS REG	Low
SIU0_RTS# SIU1_RTS#	RESET	High
SIU0_DTR# SIU1_DTR#	RESET	High

12.5.3 Internal Register Descriptions

There are 12 registers in the UART. These registers share eight address locations in the I/O address space. Table 119 shows the registers and their addresses as offsets of a base address. Note that the state of the Divisor Latch Bit (DLAB), which is the MOST significant bit of the Serial Line Control Register, affects the selection of certain of the UART registers. The DLAB bit must be set high by the system software to access the Baud Rate Generator Divisor Latches.

Table 119. Internal Register Descriptions

UART Register Addresses (Base + offset)	DLAB Bit Value	Register Accessed
Base	0	Receive BUFFER (read only)
Base	0	Transmit BUFFER (write only)
Base + 01H	0	Interrupt Enable (R/W)
Base + 02H	X	Interrupt I.D. (read only)
Base + 02H	X	FIFO Control (write only)
Base + 03H	X	Line Control (R/W)
Base + 04H	X	Modem Control (R/W)
Base + 05H	X	Line Status (Read only)
Base + 06H	X	Modem Status (Read only)
Base + 07H	X	Scratch Pad (R/W)
Base	1	Divisor Latch (Lower Byte, R/W)
Base + 01H	1	Divisor Latch (Upper Byte, R/W)

12.5.3.1 Receive Buffer Register (RBR)

In non-FIFO mode, this register holds the character received by the UART's Receive Shift Register. If fewer than 8 bits are received, the bits are right-justified and the leading bits are zeroed. Reading the register empties the register and resets the Data Ready (DR) bit in the Line Status Register to 0. Other (error) bits in the Line Status Register are not cleared. In FIFO mode, this register latches the value of the data byte at the top of the FIFO.

Register Offset: Base (DLAB=0) Attribute: RO
Default Value: 00H Size: 8 bits

Bit Number	Bit Mnemonic	Function
7:0	RB[7:0]	Data byte received, least significant bit first

12.5.3.2 Transmit Holding Register (THR)

This register holds the next data byte to be transmitted. When the Transmit Shift Register becomes empty, the contents of the Transmit Holding Register are loaded into the shift register and the Transmit Data Request (TDRQ) bit in the Line Status Register is set to 1.

Register Offset: Base (DLAB=0) Attribute: WO
Default Value: 00H Size: 8 bits

Bit Number	Bit Mnemonic	Function
7:0	TB[7:0]	Data byte transmitted, least significant bit first

In FIFO mode, writing to THR puts data to the top of the FIFO. The data at the bottom of the FIFO is loaded to the shift register when it is empty.

12.5.3.3 Interrupt Enable Register (IER)

This register enables five types of interrupts which independently activate the INT signal and set a value in the Interrupt Identification Register. Each of the five interrupt types can be disabled by resetting the appropriate bit of the IER register. Similarly, by setting the appropriate bits, selected interrupts can be enabled. Receiver time out interrupt can be configured to be separated from the receive data available interrupt (using the bit5: COMP) to avoid interrupt controller and DMA controller serving the receive FIFO at the same time.

An error interrupt is also supported in the unit when DMA requests are enabled. This interrupt is generated when bit 7 of LSR is set to 1, since no receive DMA request is generated when the receive FIFO has an error. The error interrupt is used to tell the processor to handle the data in the receive FIFO through programmed I/O. Note that the user needs to read LSR first to check if an interrupt is the error interrupt before checking IIR for the source of an interrupt if DMA requests are enabled. If DMA requests are disabled, user does not need to check for the error interrupt since it happens only when DMA requests are enabled.

The use of bit 5 to bit 4 is different from the register definition of standard 16550.

Register Offset: Base +01H (DLAB=0) Attribute: Read/Write
Default Value: 00H Size: 8 bits

Bit Number	Bit Mnemonic	Function
7:6	RSVD	RSVD = 0
5	COMP	**Compatibility Enable**. 0 = Bit 0 of this register also controls RTOIE and bit 4 is RSVD. 1 = Bit 4 of this register controls RTOIE.
4	RTOIE	**Receiver Time Out Interrupt Enable**. 0 = Receiver data Time out interrupt disabled 1 = Receiver data Time out interrupt enabled
3	MIE	**Modem Interrupt Enable**. 0 = Modem Status interrupt disabled 1 = Modem Status interrupt enabled
2	RLSE	**Receiver Line Status Interrupt Enable**. 0 = Receiver Line Status interrupt disabled 1 = Receiver Line Status interrupt enabled
1	TIE	**Transmit Data Request Interrupt Enable**. 0 = Transmit FIFO Data Request interrupt disabled 1 = Transmit FIFO Data Request interrupt enabled
0	RAVIE	**Receiver Data Available Interrupt Enable**. When BIT 5 = 1 0 = Receiver Data Available (Trigger level reached) interrupt disabled 1 = Receiver Data Available (Trigger level reached) interrupt enabled When BIT 5 = 0 the following additional functionality is used. 0 = Receiver data Time Out Interrupt also disabled 1 = Receiver data Time Out Interrupt enabled

Note: User needs to make sure that DMAE (F0h bit 0) and TIE and RAVIE are not set to 1s at the same time to avoid DMA controller and programmed I/O accessing the same FIFO.

12.5.3.4 Interrupt Identification Register (IIR)

In order to minimize software overhead during data character transfers, the UART prioritizes interrupts into four levels (listed in Table 120) and records these in the Interrupt Identification Register. The Interrupt Identification Register (IIR) stores information indicating that a prioritized interrupt is pending and the source of that interrupt.

Table 120. Interrupt Conditions

Priority Level	Interrupt Origin
1 (highest)	Receiver Line Status: one or more error bits were set.
2	Received Data is available. In FIFO mode, trigger level was reached; in non-FIFO mode, RBR has data.
2	Receiver Time out occurred. It happens in FIFO mode only, when there is data in the receive FIFO but no activity for a time period.
3	Transmitter requests data. In FIFO mode, the transmit FIFO is half or more than half empty; in non-FIFO mode, THR is read already.
4	Modem Status: one or more of the modem input signals has changed state

Register Offset: Base +02H Attribute: RO
Default Value: 01H Size: 8 bits

Bit Number	Bit Mnemonic	Function
7:6	FIFOES[1:0]	**FIFO Mode Enable Status**. 00 = Non-FIFO mode is selected 01 = Reserved. 10 = Reserved. 11 = FIFO mode is selected (TRFIFOE = 1)
5:4	—	Reserved.
3	TOD (IID3)	**Time Out Detected**. 0 = No time out interrupt is pending 1 = Time out interrupt is pending. (FIFO mode only)
2:1	IID[2:1]	**Interrupt Source Encoded**. 00 = Modem Status (CTS, DSR, RI, DCD modem signals changed state) 01 = Transmit FIFO requests data 10 = Received Data Available 11 = Receive error (Overrun, parity, framing, break, FIFO error)
0	IP#	**Interrupt Pending**. 0 = Interrupt is pending. (Active low) 1 = No interrupt is pending

Table 121. Interrupt Identification Register Decode

Interrupt ID bits				Interrupt SET/RESET Function			
3	2	1	0	**Priority**	**Type**	**Source**	**RESET Control**
0	0	0	1	-	None	No Interrupt is pending.	–
0	1	1	0	Highest	Receiver Line Status	Overrun Error, Parity Error, Framing Error, Break Interrupt.	Reading the Line Status Register.
0	1	0	0	Second Highest	Received Data Available.	Non-FIFO mode: Receive Buffer is full.	Non-FIFO mode: Reading the Receiver Buffer Register.
						FIFO mode: Trigger level was reached.	FIFO mode: Reading bytes until Receiver FIFO drops below trigger level or setting RESETRF bit in FCR register.
1	1	0	0	Second Highest	Character Timeout indication.	FIFO Mode only: At least 1 character is in receiver FIFO and there was no activity for a time period.	Reading the Receiver FIFO or setting RESETRF bit in FCR register.
0	0	1	0	Third Highest	Transmit FIFO Data Request	Non-FIFO mode: Transmit Holding Register Empty	Reading the IIR Register (if the source of the interrupt) or writing into the Transmit Holding Register.
						FIFO mode: Transmit FIFO has half or less than half data.	Reading the IIR Register (if the source of the interrupt) or writing to the Transmitter FIFO.
0	0	0	0	Fourth Highest	Modem Status	Clear to Send, Data Set Ready, Ring Indicator, Received Line Signal Detect	Reading the modem status register

12.5.3.5 FIFO Control Register (FCR)

FCR is a write only register that is located at the same address as the IIR (IIR is a read only register). FCR enables/disables the transmitter/receiver FIFOs, clears the transmitter/receiver FIFOs, and sets the receiver FIFO trigger level.

Register Offset: Base +02H Attribute: WO
Default Value: 00H Size: 8 bits

Bit Number	Bit Mnemonic	Function
7:6	ITL[1:0]	**Interrupt Trigger Level**. When the number of bytes in the receiver FIFO equals the interrupt trigger level programmed into this field and the Received Data Available Interrupt is enabled (via IER), an interrupt is generated and appropriate bits are set in the IIR. The receive DMA request is generated too when trigger level is reached and trigger level is >1. 00 = 1 byte or more in FIFO causes interrupt 01 = RSVD 10 = 8 bytes or more in FIFO causes interrupt and DMA request 11 = RSVD
5:3	–	Reserved.
2	RESETTF	**Reset Transmitter FIFO**. When RESETTF is set to 1, the transmitter FIFO counter logic is set to 0, effectively clearing all the bytes in the FIFO. The TDRQ bit in LSR are set and IIR shows a transmitter requests data interrupt if the TIE bit in the IER register is set. The transmitter shift register is not cleared; it completes the current transmission. After the FIFO is cleared, RESETTF is automatically reset to 0. 0 = Writing 0 has no effect 1 = The transmitter FIFO is cleared (FIFO counter set to 0). After clearing, bit is automatically reset to 0
1	RESETRF	**Reset Receiver FIFO**. When RESETRF is set to 1, the receiver FIFO counter is reset to 0, effectively clearing all the bytes in the FIFO. The DR bit in LSR is reset to 0. All the error bits in the FIFO and the FIFOE bit in LSR are cleared. Any error bits, OE, PE, FE or BI, that had been set in LSR are still set. The receiver shift register is not cleared. If IIR had been set to Received Data Available, it is cleared. After the FIFO is cleared, RESETRF is automatically reset to 0. 0 = Writing 0 has no effect 1 = The receiver FIFO is cleared (FIFO counter set to 0). After clearing, bit is automatically reset to 0
0	TRFIFOE	**Transmit and Receive FIFO Enable**. TRFIFOE enables/disables the transmitter and receiver FIFOs. When TRFIFOE = 1, both FIFOs are enabled (FIFO Mode). When TRFIFOE = 0, the FIFOs are both disabled (non-FIFO Mode). Writing a 0 to this bit clears all bytes in both FIFOs. When changing from FIFO mode to non-FIFO mode and vice versa, data is automatically cleared from the FIFOs. This bit must be 1 when other bits in this register are written or the other bits are not programmed. 0 = FIFOs are disabled 1 = FIFOs are enabled

12.5.3.6 Line Control Register (LCR)

In the Line Control Register (LCR), the system programmer specifies the format of the asynchronous data communications exchange. The serial data format consists of a start bit (logic 0), five to eight data bits, an optional parity bit, and one or two stop bits (logic 1). The LCR has bits for accessing the Divisor Latch and causing a break condition. The programmer can also read the contents of the Line Control Register. The read capability simplifies system programming and eliminates the need for separate storage in system memory.

Register Offset:	Base +03H	Attribute:	Read/Write
Default Value:	00H	Size:	8 bits

Bit Number	Bit Mnemonic	Function
7	DLAB	**Divisor register access bit**. This bit is the Divisor Latch Access Bit. It must be set high (logic 1) to access the Divisor Latches of the Baud Rate Generator during a READ or WRITE operation. It must be set low (logic 0) to access the Receiver Buffer, the Transmit Holding Register, or the Interrupt Enable Register. 0 = access Transmit Holding register (THR), Receive Buffer Register (RBR) and Interrupt Enable Register. 1 = access Divisor Latch Registers (DLL and DLM)
6	SB	**Set break**. This bit is the set break control bit. It causes a break condition to be transmitted to the receiving UART. When SB is set to a logic 1, the serial output (TXD) is forced to the spacing (logic 0) state and remains there until SB is set to a logic 0. This bit acts only on the TXD pin and has no effect on the transmitter logic. This feature enables the processor to alert a terminal in a computer communications system. If the following sequence is executed, no erroneous characters will be transmitted because of the break: • Load 00H in the Transmit Holding register in response to a TDRQ interrupt • After TDRQ goes high (indicating that 00H is being shifted out), set the break bit before the parity or stop bits reach the TXD pin • Wait for the transmitter to be idle (TEMT = 1) and clear the break bit when normal transmission has to be restored During the break, the transmitter can be used as a character timer to accurately establish the break duration. In FIFO mode, wait for the transmitter to be idle (TEMT=1) to set and clear the break bit. 0 = no effect on TXD output 1 = forces TXD output to 0 (space)
5	STKYP	**Sticky Parity**. This bit is the “sticky parity” bit, which can be used in multiprocessor communications. When PEN and STKYP are logic 1, the bit that is transmitted in the parity bit location (the bit just before the stop bit) is the complement of the EPS bit. If EPS is 0, then the bit at the parity bit location will be transmitted as a 1. In the receiver, if STKYP and PEN are 1, then the receiver compares the bit that is received in the parity bit location with the complement of the EPS bit. If the values being compared are not equal, the receiver sets the Parity Error bit in LSR and causes an error interrupt if line status interrupts were enabled. For example, if EPS is 0, the receiver expects the bit received at the parity bit location to be 1. If it is not, then the parity error bit is set. By forcing the bit value at the parity bit location, rather than calculating a parity value, a system with a master transmitter and multiple receivers can identify some transmitted characters as receiver addresses and the rest of the characters as data. If PEN = 0, STKYP is ignored. 0 = no effect on parity bit 1 = Forces parity bit to be opposite of EPS bit value

4	EPS	**Even Parity Select**. This bit is the even parity select bit. When PEN is a logic 1 and EPS is a logic 0, an odd number of logic ones is transmitted or checked in the data word bits and the parity bit. When PEN is a logic 1 and EPS is a logic 1, an even number of logic ones is transmitted or checked in the data word bits and parity bit. If PEN = 0, EPS is ignored. 0 = sends or checks for odd parity 1 = sends or checks for even parity
3	PEN	**Parity Enable**. This is the parity enable bit. When PEN is a logic 1, a parity bit is generated (transmit data) or checked (receive data) between the last data word bit and Stop bit of the serial data. (The parity bit is used to produce an even or odd number of ones when the data word bits and the parity bit are summed.) 0 = no parity function 1 = allows parity generation and checking
2	STB	**Stop Bits**. This bit specifies the number of stop bits transmitted and received in each serial character. If STB is a logic 0, one stop bit is generated in the transmitted data. If STB is a logic 1 when a 5-bit word length is selected via bits 0 and 1, then 1 and one half stop bits are generated. If STB is a logic 1 when either a 6, 7, or 8-bit word is selected, then two stop bits are generated. The receiver checks the first stop bit only, regardless of the number of stop bits selected. 0 = 1 stop bit 1 = 2 stop bits, except for 5-bit character then 1-1/2 bits
1:0	WLS[1:0]	**Word Length Select**. The Word Length Select bits specify the number of data bits in each transmitted or received serial character. 00 = 5-bit character (default) 01 = 6-bit character 10 = 7-bit character 11 = 8-bit character

12.5.3.7 Line Status Register (LSR)

This register provides status information to the processor concerning the data transfers. Bits 5 and 6 show information about the transmitter section. The rest of the bits contain information about the receiver.

In non-FIFO mode, three of the LSR register bits, parity error, framing error, and break interrupt, show the error status of the character that has just been received. In FIFO mode, these three bits of status are stored with each received character in the FIFO. LSR shows the status bits of the character at the top of the FIFO. When the character at the top of the FIFO has errors, the LSR error bits are set and are not cleared until software reads LSR, even if the character in the FIFO is read and a new character is now at the top of the FIFO.

Bits 1 through 4 are the error conditions that produce a receiver line status interrupt when any of the corresponding conditions are detected and the interrupt is enabled. These bits are not cleared by reading the erroneous byte from the FIFO or receive buffer. They are cleared only by reading LSR. In FIFO mode, the line status interrupt occurs only when the erroneous byte is at the top of the FIFO. If the erroneous byte being received is not at the top of the FIFO, an interrupt is generated only after the previous bytes are read and the erroneous byte is moved to the top of the FIFO.

Register Offset:	Base +05H	Attribute:	RO
Default Value:	60H	Size:	8 bits

Bit Number	Bit Mnemonic	Function
7	FIFOE	**FIFO Error Status**. In non-FIFO mode, this bit is 0. In FIFO Mode, FIFOE is set to 1 when there is at least one parity error, framing error, or break indication for any of the characters in the FIFO. Note that a processor read to the Line Status register does not reset this bit. FIFOE is reset when all error bytes have been read from the FIFO. If DMA requests are enabled (IER bit7 is set to 1) and FIFOE is set to 1, no receive DMA request is generated even the receive FIFO reaches the trigger level and the error interrupt is generated. If DMA requests are not enabled (IER bit7 is set to 0), FIFOE set to 1 does not generate interrupt. 0 = No FIFO or no errors in receiver FIFO 1 = At least one character in receiver FIFO has errors
6	TEMT	**Transmitter Empty**. TEMT is set to a logic 1 when the Transmit Holding register and the Transmitter Shift register are both empty. It is reset to a logic 0 when either the Transmit Holding register or the transmitter shift register contains a data character. In FIFO mode, TEMT is set to 1 when the transmitter FIFO and the Transmit Shift register are both empty.
5	TRDQ	**Transmit Data Request**. TDRQ indicates that the UART is ready to accept a new character for transmission. In addition, this bit causes the UART to issue an interrupt to the processor when the transmit data request interrupt enable is set high and generates the DMA request to DMA controller to ask for data. The TDRQ bit is set to a logic 1 when a character is transferred from the Transmit Holding register into the Transmit Shift register. The bit is reset to logic 0 concurrently with the loading of the Transmit Holding register by the processor. In FIFO mode, TDRQ is set to 1 when the transmit FIFO is empty or the RESETTF bit in FCR has been set to 1. It is cleared when at least one byte is written to the transmit FIFO. If more than 16 characters are loaded into the FIFO, the excess characters are lost. 0 = Processor has loaded the Transmit Holding Register 1 = Transmit FIFO is empty (FIFO mode) or a character has transferred from the Transmit Holding register into the Transmit Shift register

4	BI	**Break Interrupt**. BI is set to a logic 1 when the received data input is held in the spacing (logic 0) state for longer than a full word transmission time (that is, the total time of Start bit + data bits + parity bit + stop bits). The Break indicator is reset when the processor reads the Line Status Register. In FIFO mode, only one character (equal to 00H), is loaded into the FIFO regardless of the length of the break condition. BI shows the break condition for the character at the top of the FIFO, not the most recently received character. 0 = No break signal has been received 1 = Break signal occurred
3	FE	**Framing Error**. FE indicates that the received character did not have a valid stop bit. FE is set to a logic 1 when the bit following the last data bit or parity bit is detected as a logic 0 bit (spacing level). If the Line Control register had been set for two stop bit mode, the receiver does not check for a valid second stop bit. The FE indicator is reset when the processor reads the Line Status Register. The UART will resynchronize after a framing error. To do this it assumes that the framing error was due to the next start bit, so it samples this “start” bit twice and then takes in the “data”. In FIFO mode, FE shows a framing error for the character at the top of the FIFO, not for the most recently received character. 0 = No Framing error 1 = Invalid stop bit has been detected
2	PE	**Parity Error**. PE indicates that the received data character does not have the correct even or odd parity, as selected by the even parity select bit. The PE is set to a logic 1 upon detection of a parity error and is reset to a logic 0 when the processor reads the Line Status register. In FIFO mode, PE shows a parity error for the character at the top of the FIFO, not the most recently received character. 0 = No Parity error 1 = Parity error has occurred
1	OE	**Overrun Error**. In non-FIFO mode, OE indicates that data in the receiver buffer register was not read by the processor before the next character was transferred into the receiver buffer register, thereby destroying the previous character. In FIFO mode, OE indicates that all 16 bytes of the FIFO are full and the most recently received byte has been discarded. The OE indicator is set to a logic 1 upon detection of an overrun condition and reset when the processor reads the Line Status register. 0 = No data has been lost 1 = Received data has been lost
0	DR	**Data Ready**. Bit 0 is set to a logic 1 when a complete incoming character has been received and transferred into the receiver buffer register or the FIFO. In non-FIFO mode, DR is reset to 0 when the receive buffer is read. In FIFO mode, DR is reset to a logic 0 if the FIFO is empty (last character has been read from RBR) or the RESETRF bit is set in FCR. 0 = No data has been received 1 = Data is available in RBR or the FIFO

12.5.3.8 Modem Control Register (MCR)

This 8-bit register controls the interface with the modem or data set (or a peripheral device emulating a modem). The contents of the Modem Control register are described below:

Register Offset: Base +04H Attribute: Read/Write
Default Value: 00H Size: 8 bits

Bit Number	Bit Mnemonic	Function
7:5	0	Reserved.
4	LOOP	**Loop Back Test Mode**. This bit provides a local Loopback feature for diagnostic testing of the UART. When LOOP is set to a logic 1, the following will occur: The transmitter serial output is set to a logic 1 state. The OUT2# signal is forced to a logic 1 state. The receiver serial input is disconnected from the pin. The output of the Transmitter Shift register is “looped back” into the receiver shift register input. The four modem control inputs (CTS#, DSR#, DCD#, and RI#) are disconnected from the pins and the modem control output pins (RTS# and DTR#) are forced to their inactive state. • Coming out of the loopback test mode may result in unpredictable activation of the delta bits (bits 3:0) in the Modem Status Register (MSR). It is recommended that MSR be read once to clear the delta bits in the MSR. The lower four bits of the Modem Control register are connected to the upper four Modem Status register bits: • DTR = 1 forces DSR to a 1 • RTS = 1 forces CTS to a 1 • OUT1 = 1 forces RI to a 1 • OUT2= 1 forces DCD to a 1 In the diagnostic mode, data that is transmitted is immediately received. This feature allows the processor to verify the transmit and receive data paths of the UART. The transmit, receive and modem control interrupts are operational, except the modem control interrupts are activated by Control register bits, not the modem control inputs. A break signal can also be transferred from the transmitter section to the receiver section in loopback mode. 0 = Normal UART operation 1 = Test mode UART operation
3	OUT2	**OUT2# Signal Control**. This bit controls the OUT2# output. When the OUT2 bit is set, OUT2# is asserted low. When the OUT2 bit is cleared, OUT2# is deasserted (set high). Outside of the UART module, the OUT2# signal is used to connect the UART's interrupt output to the Interrupt Controller unit. 0 = OUT2# signal is 1, which disables the UART interrupt. 1 = OUT2# signal is 0
2	OUT1	**Test Bit**. This bit is used only in Loopback test mode. See (LOOP) Above.
1	RTS	**Request to Send**. This bit controls the Request to Send (RTS#) output pin. Bit 1 affects the RTS# output in a manner identical to that described below for the DTR bit. 0 = RTS# pin is 1 1 = RTS# pin is 0
0	DTR	**Data Terminal Ready**. This bit controls the Data Terminal Ready output. When bit 0 is set to a logic 1, the DTR# output is force to a logic 0. When bit 0 is reset to a logic 0, the DTR# output pin is forced to a logic 1. • The DTR# output of the UART may be applied to an EIA inverting line driver (such as the DS1488) to obtain the proper polarity input at the succeeding modem or data set. 0 = DTR# pin is 1 1 = DTR# pin is 0

12.5.3.9 Modem Status Register (MSR)

This 8 bit register provides the current state of the control lines from the modem or data set (or a peripheral device emulating a modem) to the processor. In addition to this current state information, four bits of the Modem Status register provide change information. These bits, 3:0, are set to a logic 1 when a control input from the Modem changes state. They are reset to a logic 0 when the processor writes ones to the bits of the Modem Status register.

When bits 0, 1, 2, or 3 are set to logic 1, a Modem Status interrupt is generated if bit 3 of the Interrupt Enable Register is set.

Register Offset:	Base +06H	Attribute:	RO
Default Value:	00H	Size:	8 bits

Bit Number	Bit Mnemonic	Function
7	DCD	**Data Carrier Detect**. This bit is the complement of the Data Carrier Detect (DCD#) input. This bit is equivalent to bit OUT2 of the Modem Control register if LOOP in the MCR is set to 1. 0 = DCD# pin is 1 1 = DCD# pin is 0
6	RI	**Ring Indicator**. This bit is the complement of the ring Indicator (RI#) input. This bit is equivalent to bit OUT1 of the Modem Control register if LOOP in the MCR is set to 1. 0 = RI# pin is 1 1 = RI# pin is 0
5	DSR	**Data Set Ready**. This bit is the complement of the Data Set Ready (DSR#) input. This bit is equivalent to bit DTR of the Modem Control register if LOOP in the MCR is set to 1. 0 = DSR# pin is 1 1 = DSR# pin is 0
4	CTS	**Clear to Send**. This bit is the complement of the Clear to Send (CTS#) input. This bit is equivalent to bit RTS of the Modem Control register if LOOP in the MCR is set to 1. 0 = CTS# pin is 1 1 = CTS# pin is 0
3	DDCD	**Delta Data Carrier Detect**. 0 = No change in DCD# pin since last read of MSR 1 = DCD# pin has changed state
2	TERI	**Trailing Edge Ring Indicator**. 0 = RI# pin has not changed from 0 to 1 since last read of MSR 1 = RI# pin has changed from 0 to 1
1	DDSR	**Delta Data Set Ready**. 0 = No change in DSR# pin since last read of MSR 1 = DSR# pin has changed state
0	DCTS	**Delta Clear To Send**. 0 = No change in CTS# pin since last read of MSR 1 = CTS# pin has changed state

12.5.3.10 Scratchpad Register (SCR)

This 8-bit read/write register has no effect on the UART. It is intended as a scratchpad register for use by the programmer.

Register Offset: Base +07H Attribute: Read/Write
Default Value: 00H Size: 8 bits

Bit Number	Bit Mnemonic	Function
7	SP[7:0]	No effect on UART functionality

12.5.3.11 Programmable Baud Rate Generator

The UART contains a programmable Baud Rate Generator that is capable of taking the UART_CLK input and dividing it by any divisor from 1 to ($2^{16} - 1$). The output frequency of the Baud Rate Generator is 16 times the baud rate. Two 8-bit latches store the divisor in a 16-bit binary format. These Divisor Latches must be loaded during initialization to ensure proper operation of the Baud Rate Generator. If both Divisor Latches are loaded with 0, the 16X output clock is stopped. Upon loading either of the Divisor latches, a 16-bit baud counter is immediately loaded. This prevents long counts on initial load. Access to the Divisor latch can be done with a word write.

The baud rate of the data shifted in/out of the UART is given by:

BaudRate = UART_CLK(MHz)/[16X Divisor]

For example, if UART_CLK is 14.7456MHz and the divisor is 96, the baud rate is 9600.

A divisor value of 0 in the Divisor Latch Register is not allowed. The reset value of the divisor is 02.

12.5.3.12 Divisor Latch Register Low (DLL)

Register Offset: Base (DLAB=1) Attribute: Read/Write
Default Value: 02H Size: 8 bits

Bit Number	Bit Mnemonic	Function
7:0	BR[7:0]	Low byte compare value to generate baud rate

12.5.3.13 Divisor Latch Register High (DLH)

Register Offset: Base + 1 (DLAB=1) Attribute: Read/Write
Default Value: 00H Size: 8 bits

Bit Number	Bit Mnemonic	Function
7:0	BR[15:8]	High byte compare value to generate baud rate

12.5.4 FIFO Operation

12.5.4.1 FIFO Interrupt Mode Operation

12.5.4.1.1 Receiver Interrupt

When the Receive FIFO and receiver interrupts are enabled (FCR[0]=1 and IER[0]=1), receiver interrupts occur as follows:

- The receive data available interrupt is invoked when the FIFO has reached its programmed trigger level. The interrupt is cleared when the FIFO drops below the programmed trigger level.
- The IIR receive data available indication also occurs when the FIFO trigger level is reached, and like the interrupt, the bits are cleared when the FIFO drops below the trigger level.
- The receiver line status interrupt (IIR = C6H), as before, has the highest priority. The receiver data available interrupt (IIR=C4H) is lower. The line status interrupt occurs only when the character at the top of the FIFO has errors.
- The data ready bit (DR in LSR register) is set to 1 as soon as a character is transferred from the shift register to the Receive FIFO. This bit is reset to 0 when the FIFO is empty.

12.5.4.1.2 Character Timeout Interrupt

When the receiver FIFO and receiver time out interrupt are enabled, a character timeout interrupt occurs when all of the following conditions exist:

- At least one character is in the FIFO.
- The last received character was longer than four continuous character times ago (if two stop bits are programmed the second one is included in this time delay).
- The most recent processor read of the FIFO was longer than four continuous character times ago.
- The receiver FIFO trigger level is greater than one.

The maximum time between a received character and a timeout interrupt is 160 ms at 300 baud with a 12-bit receive character (i.e., one start, eight data, one parity, and two stop bits).

When a time out interrupt occurs, it is cleared and the timer is reset when the processor reads one character from the receiver FIFO. If a timeout interrupt has not occurred, the timeout timer is reset after a new character is received or after the processor reads the receiver FIFO.

12.5.4.1.3 Transmit Interrupt

When the transmitter FIFO and transmitter interrupt are enabled (FCR[0]=1, IER[1]=1), transmit interrupts occur as follows:

- The Transmit Data Request interrupt occurs when the transmit FIFO is half empty or more than half empty. The interrupt is cleared as soon as the Transmit Holding Register is written (1 to 16 characters may be written to the transmit FIFO while servicing the interrupt) or the IIR is read.

12.5.4.2 FIFO Polled Mode Operation

With the FIFOs enabled (TRFIFOE bit of FCR set to 1), setting IER[3:0] to all zeros puts the serial port in the FIFO polled mode of operation. Since the receiver and the transmitter are controlled separately, either one or both can be in the polled mode of operation. In this mode, software checks receiver and transmitter status via the LSR. As stated in the register description:

- LSR[0] is set as long as there is one byte in the receiver FIFO.
- LSR[1] through LSR[4] specify which error(s) has occurred for the character at the top of the FIFO. Character error status is handled the same way as interrupt mode. The IIR is not affected since IER[2] = 0.
- LSR[5] indicates when the transmitter FIFO needs data.
- LSR[6] indicates that both the transmitter FIFO and shift register are empty.
- LSR[7] indicates whether there are any errors in the receiver FIFO.

12.5.5 DMA Requests

The UART has two DMA requests, one for transmit data service and one for receive data service. DMA requests are generated in FIFO mode only.

When the transmit FIFO is half or more than half empty, the transmit DMA request is generated if the DMA request enable bit in the Logical Device Register (F0h) is set to 1. The DMA controller then writes data to the FIFO. For each DMA request, DMA controller can send 8 bytes of data to the FIFO. The number of bytes to be transmitted is programmed in the DMA controller.

When the receive FIFO reaches its trigger level (except the case when trigger level is set to 1) and no errors within these entries, the receive DMA request is generated if the DMA request enable bit in the Logical Device Register (F0h) is set to 1. The DMA controller then reads data from the FIFO. For each DMA request, DMA controller can read 8 bytes of data from the FIFO. The number of bytes to be read is programmed in the DMA controller.

When DMA requests are enabled (bit 0 in F0h configuration space is set to 1) and there is an error written to the receive FIFO, if the trigger level is not reached, then the receive DMA request is disabled and the error interrupt is generated; if the trigger level has been reached and the receive DMA request is 1, then the receive DMA request is set to 0 immediately and the error interrupt is set to 1. When all the errors in the receive FIFO are read out, the receive DMA request is generated again as its trigger level is reached. Note that the programmer needs to make sure that the DMA controller has finished the previous receive DMA request before responding to the error interrupt.

In non-FIFO mode, the data is written to THR or read from RBR through programmed I/O.

12.5.5.1 Trailing Bytes in the Receive FIFO

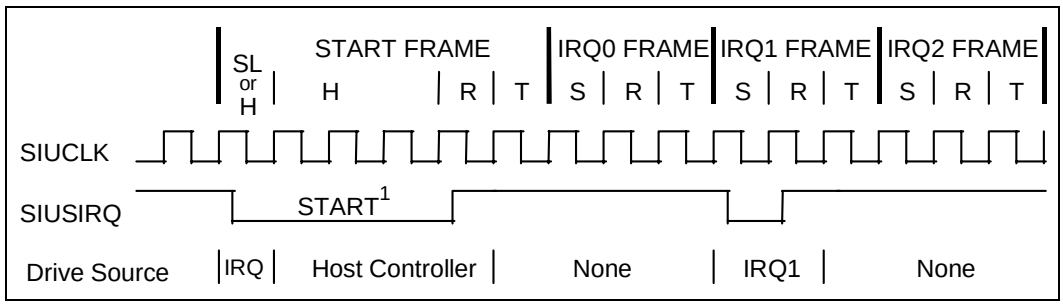
When the entries in the receive FIFO is less than trigger level and there is no more data coming in, these bytes are called trailing bytes. In this case no receive DMA request is generated. To read out the trailing bytes, user needs to wait for time out interrupt and then reads one byte at a time until the FIFO is empty (poll the Line Status Register bit 0) through programmed I/O. Note that the time out interrupt must be enabled and the trigger level must be greater than one.

12.6 Serial IRQ

The SIU supports the serial interrupt to transmit interrupt information to the host system. The serial interrupt scheme adheres to the Serial IRQ Specification.

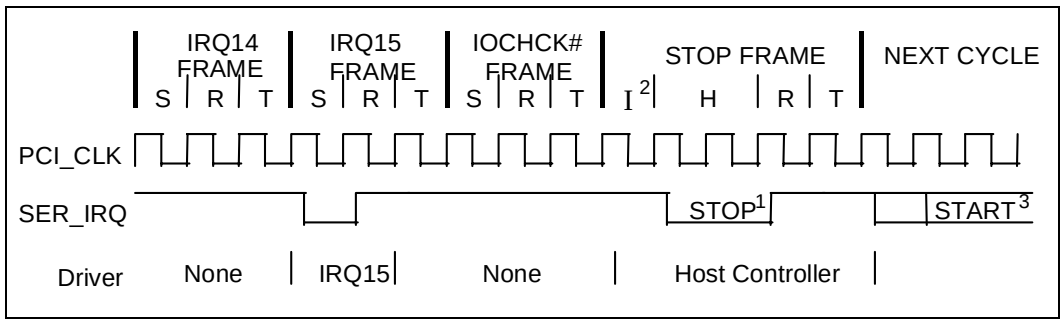
12.6.1 Timing Diagrams For SIU_SERIRQ Cycle

Figure 30. Start Frame Timing with Source Sampled a Low Pulse on IRQ1



NOTES:
1. H=Host Control; R=Recovery; T=Turn-Around; SL=Slave Control; S=Sample
2. Start Frame pulse can be 4-8 clocks wide depending on the location of the device in the PCI bridge hierarchy in a synchronous bridge design.

Figure 31. Stop Frame Timing with Host Using 17 SIU_SERIRQ Sampling Period



NOTES:
1. H=Host Control; R=Recovery; T=Turn-Around; S=Sample; I=Idle
2. Stop pulse is 2 clocks wide for Quiet mode, 3 clocks wide for Continuous mode.
3. There may be none, one or more Idle states during the Stop Frame.
4. The next SIU_SERIRQ cycle's Start Frame pulse may or may not start immediately after the turn-around clock of the Stop Frame.

12.6.1.1 SIU_SERIRQ Cycle Control

There are two modes of operation for the SIU_SERIRQ Start Frame.

1. **Quiet (Active) Mode**: Any device may initiate a Start Frame by driving the SIU_SERIRQ low for one clock, while the SIU_SERIRQ is Idle. After driving low for one clock the SIU_SERIRQ is immediately tri-stated without at any time driving high. A Start Frame may not be initiated while the SIU_SERIRQ is Active. The SIU_SERIRQ is Idle between Stop and Start Frames. The SIU_SERIRQ is Active between Start and Stop Frames. This mode of operation allows the SIU_SERIRQ to be Idle when there are no IRQ/Data transitions which should be most of the time.

Once a Start Frame has been initiated the Host Controller will take over driving the SIU_SERIRQ low in the next clock and will continue driving the SIU_SERIRQ low for a programmable period of three to seven clocks. This makes a total low pulse width of four to eight clocks. Finally, the Host Controller will drive the SIU_SERIRQ back high for one clock, then tri-state.

Any SIU_SERIRQ Device (i.e., the SIU) which detects any transition on an IRQ/Data line for which it is responsible must initiate a Start Frame in order to update the Host Controller unless the SIU_SERIRQ is already in an SIU_SERIRQ Cycle and the IRQ/Data transition can be delivered in that SIU_SERIRQ Cycle.

2. **Continuous (Idle) Mode**: Only the Host controller can initiate a Start Frame to update IRQ/Data line information. All other SIU_SERIRQ agents become passive and may not initiate a Start Frame. SIU_SERIRQ will be driven low for four to eight clocks by Host Controller. This mode has two functions. It can be used to stop or idle the SIU_SERIRQ or the Host Controller can operate SIU_SERIRQ in a continuous mode by initiating a Start Frame at the end of every Stop Frame.

   An SIU_SERIRQ mode transition can only occur during the Stop Frame. **Upon reset, SIU_SERIRQ bus is defaulted to Continuous mode, therefore only the Host controller can initiate the first Start Frame. Slaves must continuously sample the Stop Frames pulse width to determine the next SIU_SERIRQ Cycle's mode.**

12.6.1.2 SIU_SERIRQ Data Frame

Once a Start Frame has been initiated, the SIU will watch for the rising edge of the Start Pulse and start counting IRQ/Data Frames from there. Each IRQ/Data Frame is three clocks: Sample phase, Recovery phase, and Turn-around phase. During the Sample phase the SIU drives the SIU_SERIRQ low, if and only if, its last detected IRQ/Data value was low. If its detected IRQ/Data value is high, SIU_SERIRQ is left tri-stated. During the Recovery phase the SIU drives the SIU_SERIRQ high, if and only if, it had driven the SIU_SERIRQ low during the previous Sample Phase. During the Turn-around Phase the SIU tri-states the SIU_SERIRQ. The SIU will drive the SIU_SERIRQ line low at the appropriate sample point if its associated IRQ/Data line is low, regardless of which device initiated the Start Frame.

The Sample Phase for each IRQ/Data follows the low to high transition of the Start Frame pulse by a number of clocks equal to the IRQ/Data Frame times three, minus one. (e.g., The IRQ5 Sample clock is the sixth IRQ/Data Frame, (6 x 3) - 1 = 17th clock after the rising edge of the Start Pulse).

Table 122. SIU_SERIRQ Sampling Periods (Sheet 1 of 2)

SIU_SERIRQ PERIOD	SIGNAL SAMPLED	# OF CLOCKS PAST START
1	Not Used	2
2	IRQ1	5
3	IRQ2	8
4	IRQ3	11
5	IRQ4	14
6	IRQ5	17
7	IRQ6	20
8	IRQ7	23
9	IRQ8	26
10	IRQ9	29

Table 122. SIU_SERIRQ Sampling Periods (Sheet 2 of 2)

SIU_SERIRQ PERIOD	SIGNAL SAMPLED	# OF CLOCKS PAST START
11	IRQ10	32
12	IRQ11	35
13	IRQ12	38
14	IRQ13	41
15	IRQ14	44
16	IRQ15	47

SIU_SERIRQ Period 14 is used to transfer IRQ13. Logical devices 4 (Ser Port 1) and 5 (Ser Port 2) shall have IRQ13 as a choice for their primary interrupt.

12.6.1.3 Stop Cycle Control

Once all IRQ/Data Frames have completed the Host Controller will terminate SIU_SERIRQ activity by initiating a Stop Frame. Only the Host Controller can initiate the Stop Frame. A Stop Frame is indicated when the SIU_SERIRQ is low for two or three clocks. If the Stop Frame's low time is two clocks then the next SIU_SERIRQ Cycle's sampled mode is the Quiet mode; and any SIU_SERIRQ device may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse. If the Stop Frame's low time is three clocks then the next SIU_SERIRQ Cycle's sampled mode is the Continuous mode; and only the Host Controller may initiate a Start Frame in the second clock or more after the rising edge of the Stop Frame's pulse.

12.6.1.4 Latency

Latency for IRQ/Data updates over the SIU_SERIRQ bus in bridge-less systems with the minimum Host supported IRQ/Data Frames of seventeen, will range up to 96 clocks (2.88 µs with a 33 MHz PCI Bus). If one or more PCI to PCI Bridge is added to a system, the latency for IRQ/Data updates from the secondary or tertiary buses will be a few clocks longer for synchronous buses, and approximately double for asynchronous buses.

12.6.1.5 EOI/ISR Read Latency

Any serialized IRQ scheme has a potential implementation issue related to IRQ latency. IRQ latency could cause an EOI or ISR Read to precede an IRQ transition that it should have followed. This could cause a system fault. The host interrupt controller is responsible for ensuring that these latency issues are mitigated. The recommended solution is to delay EOIs and ISR Reads to the interrupt controller by the same amount as the SIU_SERIRQ Cycle latency in order to ensure that these events do not occur out of order.

12.6.1.6 Reset and Initialization

The SIU_SERIRQ bus uses SIU_LRESET# as its reset signal. The SIU_SERIRQ pin is tri-stated by all agents while SIU_LRESET# is active. With reset, SIU_SERIRQ Slaves are put into the (continuous) IDLE mode. The Host Controller is responsible for starting the initial SIU_SERIRQ Cycle to collect system's IRQ/Data default values. The system then follows with the Continuous/Quiet mode protocol (Stop Frame pulse width) for subsequent SIU_SERIRQ Cycles. It is Host Controller's responsibility to provide the default values to the Interrupt controller and other system logic before the first SIU_SERIRQ Cycle is performed. For SIU_SERIRQ system suspend,

insertion, or removal application, the Host controller should be programmed into Continuous (IDLE) mode first. This is to guarantee SIU_SERIRQ bus is in IDLE state before the system configuration changes.

12.7 Configuration

The configuration of the SIU is very flexible and is based on the configuration architecture implemented in typical Plug-and-Play components. The SIU is designed for motherboard applications in which the resources required by their components are known. With its flexible resource allocation architecture, the SIU allows the BIOS to assign resources at POST.

12.7.1 Configuration Port Address Selection

The SIU configuration port addresses for INDEX and DATA are fixed at 4Eh/4Fh.

See also “LPC_EN—LPC I/F Enables (LPC I/F—D31:F0)” on page 221.

12.7.2 Primary Configuration Address Decoder

After a PCI Reset (SIU_LRESET# pin asserted) or Power On Reset the SIU is in the Run Mode with the two UARTs disabled. They may be configured through two standard Configuration I/O Ports (INDEX and DATA) by placing the SIU into Configuration Mode.

The BIOS uses these configuration ports to initialize the logical devices at POST. The INDEX and DATA ports are only valid when the SIU is in Configuration Mode.

The INDEX and DATA ports are effective only when the chip is in the Configuration State. When the SIU is not in the Configuration State, reads return FFh and write data is ignored.

12.7.2.1 Entering the Configuration State

The device enters the Configuration State by the following contiguous sequence:

Write 80H to Configuration Port
Write 86H to Configuration Port

12.7.2.2 Exiting the Configuration State

The device exits the Configuration State by the following contiguous sequence:

Write 68H to Configuration Port
Write 08H to Configuration Port

12.7.2.3 Configuration Sequence

To program the configuration registers, the following sequence must be followed:

1. Enter Configuration Mode
2. Configure the Configuration Registers
3. Exit Configuration Mode.

12.7.2.4 Configuration Mode

The system sets the logical device information and activates desired logical devices through the INDEX and DATA ports. In configuration mode, the INDEX PORT is located at the CONFIG PORT address and the DATA PORT is at INDEX PORT address + 1.

The desired configuration registers are accessed in two steps:

1. Write the index of the Logical Device Number Configuration Register (i.e., 07) to the INDEX PORT and then write the number of the desired logical device to the DATA PORT.
2. Write the address of the desired configuration register within the logical device to the INDEX PORT and then write or read the configuration register through the DATA PORT.

Note: If accessing the Global Configuration Registers, step (a) is not required.

The chip returns to the RUN State.

Note: Only two states are defined (Run and Configuration). In the Run State the chip will always be ready to enter the Configuration State.

12.7.3 SIU Configuration Registers Summary

Table 123. Configuration Registers Summary

Global Configuration Registers			
Index	**Type**	**Default**	**Configuration Register**
07h	R/W	00h	Logical Device Number
20h	R	00h	Device ID
21h	R	00h	Device Rev
28h	R/W	00h	SIU I/F (wait states)
29h	R/W	00h	SIRQ Configuration
2Eh	R/W	TBD	Test Mode Configuration Register
Logical Device 4 Registers (Serial Port 0)			
30h	R/W	00h	Enable
60h	R/W	00h	Base I/O Address MSB
61h	R/W	00h	Base I/O Address LSB
70h	R/W	00h	Primary Interrupt Select
74h	R/W	04h	DMA Receive Channel Select
75h	R/W	04h	DMA Transmit Channel Select
F0h	R/W	00h	Vendor Specific Configuration
Logical Device 5 Registers (Serial Port 1)			
30h	R/W	00h	Enable
60h	R/W	00h	Base I/O Address MSB
61h	R/W	00h	Base I/O Address LSB
70h	R/W	00h	Primary Interrupt Select
74h	R/W	04h	DMA Receive Channel Select
75h	R/W	04h	DMA Transmit Channel Select
F0h	R/W	00h	Vendor Specific Configuration

NOTE: Reserved registers are read-only, reads return 0.

12.7.3.1 Global Control/Configuration Registers [00h — 2Fh]

The chip-level (global) registers lie in the address range [00h-2Fh]. The design MUST use all 8 bits of the ADDRESS Port for register selection. All unimplemented registers and bits ignore writes and return zero when read.

The INDEX PORT is used to select a configuration register in the chip. The DATA PORT is then used to access the selected register. These registers are accessible only in the Configuration Mode.

Table 124. Global Control Registers

Register	Address (Type)	Description
Logical Device #' Default = 00h	07h (R/W)	**Logical Device Select**: A write to this register selects the current logical device. This allows access to the control and configuration registers for each logical device.
Device ID Default = 00h	20h (R)	**Device ID:** A read only register which provides the Device ID.
Device Rev Default = 00h	21h (R)	**Device Rev:** A read only register which provides device revision information.
SIU Interface Default = 02h	28h (R/W bits 7:2, 0 R- bit 1)	Bit 1 – LPC Bus Wait States 1 = long wait states (sync 6) 0 = Not supported Bit 7:2, 0 – RSVD = 0
SIU Configuration Default = 00h	29h (R/W bits 3:2, 0 R- bit 1)	Bit 0 – SIRQ Enable 1 = enabled; participates in interrupt generation 0 = disabled; serial interrupts disabled Bit 1 – IRQ Mode (Read only, Writes ignored) 1 = Continuous mode 0 = Quiet mode Bit 3:2 – UART_CLK pre-divide UART_CLK input 00 Divide by 1 1.8432 MHz 01 Divide by 8 14.7456 MHz 10 Divide by 26 48 MHz 11 Reserved. Bit 7:4 – RSVD = 0

12.7.3.2 Logical Device Configuration Registers [30h — FFh]

Used to access the registers that are assigned to each logical unit. This chip supports two logical units and has two sets of logical device registers. The two logical devices are UART0 and UART1. A separate set (bank) of control and configuration registers exists for each logical device and is selected with the Logical Device # Register.

The INDEX PORT is used to select a specific logical device register. These registers are then accessed through the DATA PORT.

The Logical Device registers are accessible only when the device is in the Configuration State. The logical register addresses are shown in the tables below.

Table 125. Logical Device 4 (Serial Port 0) (Sheet 1 of 2)

Logical Device Register	Address	Description

Enable
Default = 00h
	30h	Bits[7:1] Reserved, set to zero.
		Bit[0]
		1 = Enable the logical device currently selected through the Logical Device # register.
		0 = Logical device currently selected is inactive

I/O Base Address Default = 00h	60-61h	Registers 60h (MSB) and 61h (LSB) set the base address for the device. Note that the decode is on 8 Byte boundaries. 82801E C-ICH Comm Decode Ranges: 3F8 – 3FF (COM 1) 2F8 – 2FF (COM 2) 220 – 227 228 – 22F 238 – 23F 2E8 – 2EF (COM 4) 338 – 33F 3E8 – 3EF (COM 3)
Primary Interrupt Select Default = 00h	70h (R/W)	Bits[3:0] selects which interrupt level is used for the primary Interrupt. 00 = No interrupt selected 01 = IRQ1 02 = IRQ2 03 = IRQ3 04 = IRQ4 05 = IRQ5 06 = IRQ6 07 = IRQ7 08 = IRQ8 09 = IRQ9 0A = IRQ10 0B = IRQ11 0C = IRQ12 0D = IRQ13 0E = IRQ14 0F = IRQ15 Bits[7:4] Reserved. **NOTE:** An Interrupt is activated by setting this register to a non-zero value and setting any combination of bits 0-3 in the corresponding UART IER and the OUT2 bit in the MCR

Table 125. Logical Device 4 (Serial Port 0) (Sheet 2 of 2)

DMA Receive Channel Select Default = 04h	74h (R/W)	Bits[3:0] select the receive FIFO DMA Channel. 00h = DMA 0 01h = DMA 1 02h = DMA 2 03h = DMA 3 04-07h = RSVD
DMA Transmit Channel Select Default = 04h	75h (R/W)	Bits[3:0] select the transmit FIFO DMA Channel. 00h = DMA 0 01h = DMA 1 02h = DMA 2 03h = DMA 3 04-07h = RSVD
DMA Enable Default = 00h	F0h (R/W)	Bit[0]: 0 = DMA requests disabled 1 = DMA requests enabled Bits[7:1] - RSVD

Table 126. Logical Device 5 (Serial Port 1) (Sheet 1 of 2)

Logical Device Register	Address	Description

Enable
Default = 00h
	30h	Bits[7:1] Reserved, set to zero.
		Bit[0]:
		1 = Enable the logical device currently selected through the Logical Device # register.
		0 = Logical device currently selected is inactive

I/O Base Address Default = 00h	60-61h	Registers 60h (MSB) and 61h (LSB) set the base address for the device. Note that the decode is on 8 Byte boundaries. 82801E C-ICH Comm Decode Ranges: 3F8 – 3FF (COM 1) 2F8 – 2FF (COM 2) 220 – 227 228 – 22F 238 – 23F 2E8 – 2EF (COM 4) 338 – 33F 3E8 – 3EF (COM 3)

Table 126. Logical Device 5 (Serial Port 1) (Sheet 2 of 2)

Logical Device Register	Address	Description
Primary Interrupt Select Default = 00h	70h (R/W)	Bits[3:0] selects which interrupt level is used for the primary Interrupt. 00 = No interrupt selected 01 = IRQ1 02 = IRQ2 03 = IRQ3 04 = IRQ4 05 = IRQ5 06 = IRQ6 07 = IRQ7 08 = IRQ8 09 = IRQ9 0A = IRQ10 0B = IRQ11 0C = IRQ12 0D = IRQ13 0E = IRQ14 0F = IRQ15 Bits[7:4] Reserved. **NOTE:** An Interrupt is activated by setting this register to a non-zero value and setting any combination of bits 0-3 in the corresponding UART IER and the OUT2 bit in the MCR
DMA Receive Channel Select Default = 04h	74h (R/W)	Bits[3:0] select the receive FIFO DMA Channel. 00h = DMA 0 01h = DMA 1 02h = DMA 2 03h = DMA 3 04-07h = RSVD
DMA Transmit Channel Select Default = 04h	75h (R/W)	Bits[3:0] select the transmit FIFO DMA Channel. 00h = DMA 0 01h = DMA 1 02h = DMA 2 03h = DMA 3 04-07h = RSVD
DMA Enable Default = 00h	F0h (R/W)	Bit[0]: 1 = DMA requests enabled 0 = DMA requests disabled Bits[7:1] - RSVD

Signal Description A

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

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

The following notations are used to describe the signal type:

I	Input Pin
O	Output Pin
OD	Open Drain Output Pin
I/O	Bidirectional Input/Output Pin

A.1 Hub Interface to Host Controller

Table 127. Hub Interface Signals

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

A.2 Link to LAN Connect

Table 128. LAN Connect Interface Signals

Name	Type	Description
LAN0_CLK **LAN1_CLK**	I	**LAN Interface Clock:** This signal is driven by the LAN Connect component. The frequency range is 0.8 MHz to 50 MHz.
LAN0_RSTSYNC **LAN1_RSTSYNC**	O	**LAN Reset/Sync:** The LAN Connect component's Reset and Sync signals are multiplexed onto this pin.
LAN0_RXD[2:0] **LAN1_RXD[2:0]**	I	**Received Data:** The LAN Connect component uses these signals to transfer data and control information to the integrated LAN Controller. These signals have integrated weak pull-up resistors.
LAN0_TXD[2:0] **LAN1_TXD[2:0]**	O	**Transmit Data**: The integrated LAN Controller uses these signals to transfer data and control information to the LAN Connect component.

A.3 EEPROM Interface

Table 129. EEPROM Interface Signals

Name	Type	Description
EE0_CS **EE1_CS**	O	**EEPROM Chip Select:** These signals are chip-select signals to the EEPROMs.
EE0_DIN **EE1_DIN**	I	**EEPROM Data In:** These signals transfer data from the EEPROMs to the 82801E C-ICH. These signals have an integrated pull-up resistor.
EE0_DOUT **EE1_DOUT**	O	**EEPROM Data Out:** These signals transfer data from the 82801E C-ICH to the EEPROMs.
EE0_SHCLK **EE1_SHCLK**	O	**EEPROM Shift Clock:** These signals are the serial shift clock output to the EEPROMs.

A.4 Firmware Hub Interface

Table 130. Firmware Hub Interface Signals

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

A.5 PCI Interface

Table 131. PCI Interface Signals (Sheet 1 of 3)

Name	Type	Description
AD[31:0]	I/O	**PCI Address/Data:** AD[31:0] is a multiplexed address and data bus. During the first clock of a transaction, AD[31:0] contain a physical address (32 bits). During subsequent clocks, AD[31:0] contain data. The 82801E C-ICH drives all 0s on AD[31:0] during the address phase of all PCI Special Cycles.
C/BE[3:0]#	I/O	**Bus Command and Byte Enables:** The command and byte enable signals are multiplexed on the same PCI pins. During the address phase of a transaction, C/BE[3:0]# define the bus command. During the data phase, C/BE[3:0]# define the Byte Enables. **C/BE[3:0]# — Command Type** 0000 — Interrupt Acknowledge 0001 — Special Cycle 0010 — I/O Read 0011 — I/O Write 0110 — Memory Read 0111 — Memory Write 1010 — Configuration Read 1011 — Configuration Write 1100 — Memory Read Multiple 1101 — DAC Mode Address to be latched (target only) 1110 — Memory Read Line 1111 — Memory Write and Invalidate All command encodings not shown are reserved. The 82801E C-ICH does not decode reserved values, and therefore will not respond when a PCI master generates a cycle using one of the reserved values. As a target, the 82801E C-ICH can support DAC mode addressing for 44 bits.

Table 131. PCI Interface Signals (Sheet 2 of 3)

Name	Type	Description
DEVSEL#	I/O	**Device Select:** The 82801E C-ICH asserts DEVSEL# to claim a PCI transaction. As an output, the 82801E C-ICH asserts DEVSEL# when a PCI master peripheral attempts an access to an internal 82801E C-ICH address or an address destined for the hub interface (main memory or AGP). As an input, DEVSEL# indicates the response to an 82801E C-ICH-initiated transaction on the PCI bus. DEVSEL# is tri-stated from the leading edge of PCIRST#. DEVSEL# remains tri-stated by the 82801E C-ICH until driven by a target device.
FRAME#	I/O	**Cycle Frame:** The current Initiator drives FRAME# to indicate the beginning and duration of a PCI transaction. While the initiator asserts FRAME#, data transfers continue. When the initiator deasserts FRAME#, the transaction is in the final data phase. FRAME# is an input to the 82801E C-ICH when the 82801E C-ICH is the target, and FRAME# is an output from the 82801E C-ICH when the 82801E C-ICH is the Initiator. FRAME# remains tri-stated by the 82801E C-ICH until driven by an Initiator.
IRDY#	I/O	**Initiator Ready:** IRDY# indicates the 82801E C-ICH's ability, as an Initiator, to complete the current data phase of the transaction. It is used in conjunction with TRDY#. A data phase is completed on any clock both IRDY# and TRDY# are sampled asserted. During a write, IRDY# indicates the 82801E C-ICH has valid data present on AD[31:0]. During a read, it indicates the 82801E C-ICH is prepared to latch data. IRDY# is an input to the 82801E C-ICH when the 82801E C-ICH is the Target and an output from the 82801E C-ICH when the 82801E C-ICH is an Initiator. IRDY# remains tri-stated by the 82801E C-ICH until driven by an Initiator.
TRDY#	I/O	**Target Ready:** TRDY# indicates the 82801E C-ICH's ability as a Target to complete the current data phase of the transaction. TRDY# is used in conjunction with IRDY#. A data phase is completed when both TRDY# and IRDY# are sampled asserted. During a read, TRDY# indicates that the 82801E C-ICH, as a Target, has placed valid data on AD[31:0]. During a write, TRDY# indicates the 82801E C-ICH, as a Target is prepared to latch data. TRDY# is an input to the 82801E C-ICH when the 82801E C-ICH is the Initiator and an output from the 82801E C-ICH when the 82801E C-ICH is a Target. TRDY# is tri-stated from the leading edge of PCIRST#. TRDY# remains tri-stated by the 82801E C-ICH until driven by a target.
STOP#	I/O	**Stop:** STOP# indicates that the 82801E C-ICH, as a Target, is requesting the Initiator to stop the current transaction. STOP# causes the 82801E C-ICH, as an Initiator, to stop the current transaction. STOP# is an output when the 82801E C-ICH is a target and an input when the 82801E C-ICH is an Initiator. STOP# is tri-stated from the leading edge of PCIRST#. STOP# remains tri-stated until driven by the 82801E C-ICH.
PAR	I/O	**Calculated/Checked Parity:** PAR uses "even" parity calculated on 36 bits, AD[31:0] plus C/BE[3:0]#. "Even" parity means that the 82801E C-ICH counts the number of 1s within the 36 bits plus PAR and the sum is always even. The 82801E C-ICH always calculates PAR on 36 bits, regardless of the valid byte enables. The 82801E C-ICH generates PAR for address and data phases and only guarantees PAR to be valid one PCI clock after the corresponding address or data phase. The 82801E C-ICH drives and tri-states PAR identically to the AD[31:0] lines except that the 82801E C-ICH delays PAR by exactly one PCI clock. PAR is an output during the address phase (delayed one clock) for all 82801E C-ICH initiated transactions. PAR is an output during the data phase (delayed one clock) when the 82801E C-ICH is the Initiator of a PCI write transaction, and when it is the target of a read transaction. 82801E C-ICH checks parity when it is the target of a PCI write transaction. If a parity error is detected, the 82801E C-ICH sets the appropriate internal status bits, and has the option to generate an NMI# or SMI#.
PERR#	I/O	**Parity Error:** An external PCI device drives PERR# when it receives data that has a parity error. The 82801E C-ICH drives PERR# when it detects a parity error. The ICH can either generate an NMI# or SMI# upon detecting a parity error (either detected internally or reported via the PERR# signal).

Table 131. PCI Interface Signals (Sheet 3 of 3)

Name	Type	Description
REQ[3:0]# /REQ[5]# /REQ[B]# /GPIO[1]	I	**PCI Requests:** The 82801E C-ICH supports up to four masters on the PCI bus. REQ[5]# is muxed with PC/PCI REQ[B]# (must choose one or the other, but not both). If not used for PCI or PC/PCI, REQ[5]#/REQ[B]# can instead be used as GPIO[1]. **NOTE:** REQ[0]# is programmable to have improved arbitration latency for supporting PCI-based 1394 controllers.
GNT[3:0]# /GNT[5]# /GNT[B]# /GPIO[17]#	O	**PCI Grants:** The 82801E C-ICH supports up to four masters on the PCI bus. GNT[5]# is muxed with PC/PCI GNT[B]# (must choose one or the other, but not both). If not used for PCI or PC/PCI, GNT[5]# can instead be used as a GPIO. Pull-up resistors are not required on these signals. If pullups are used, they should be tied to the Vcc3_3 power rail. GNT[B]#/GNT[5]#/GPIO[17] has an internal pull-up.
PCICLK	I	**PCI Clock:** This is a 33 MHz clock. PCICLK provides timing for all transactions on the PCI Bus.
PCIRST#	O	**PCI Reset:** 82801E C-ICH asserts PCIRST# to reset devices that reside on the PCI bus. The 82801E C-ICH asserts PCIRST# during power-up and when S/W initiates a hard reset sequence through the RC (CF9h) register. The 82801E C-ICH drives PCIRST# inactive a minimum of 1 ms after PWROK is driven active. The 82801E C-ICH drives PCIRST# active a minimum of 1 ms when initiated through the RC register.
PLOCK#	I/O	**PCI Lock:** PLOCK# indicates an exclusive bus operation and may require multiple transactions to complete. 82801E C-ICH asserts PLOCK# when it performs non-exclusive transactions on the PCI bus.
SERR#	I	**System Error:** SERR# can be pulsed active by any PCI device that detects a system error condition. Upon sampling SERR# active, the 82801E C-ICH has the ability to generate an NMI, SMI#, or interrupt.
REQ[A]# /GPIO[0] **REQ[B]#** /REQ[5]# /GPIO[1]	I	**PC/PCI DMA Request [A:B]:** This request serializes ISA-like DMA Requests for the purpose of running ISA-compatible DMA cycles over the PCI bus. This is used by devices such as PCI-based Super I/O or audio codecs that need to perform legacy 8237 DMA but have no ISA bus. When not used for PC/PCI requests, these signals can be used as General Purpose Inputs. Instead, REQ[B]# can be used as the fourth PCI bus request.
GNT[A]# /GPIO[16] **GNT[B]#** /GNT[5]# /GPIO[17]	O	**PC/PCI DMA Acknowledges [A:B]:** This grant serializes an ISA-like DACK# for the purpose of running DMA/ISA master cycles over the PCI bus. This is used by devices such as PCI-based Super I/O or audio codecs which need to perform legacy 8237 DMA but have no ISA bus. When not used for PC/PCI, these signals can be used as General Purpose Outputs. GNTB# can also be used as the fourth PCI bus master grant output. These signal have internal pull-up resistors.

A.6 IDE Interface

Table 132. IDE Interface Signals

Name	Type	Description
PDCS1# **SDCS1#**	O	**Primary and Secondary IDE Device Chip Selects for 100 Range:** These signals are for the ATA command register block. This output signal is connected to the corresponding signal on the primary or secondary IDE connector.
PDCS3# **SDCS3#**	O	**Primary and Secondary IDE Device Chip Select for 300 Range:** These signals are for the ATA control register block. This output signal is connected to the corresponding signal on the primary or secondary IDE connector.
PDA[2:0] **SDA[2:0]**	O	**Primary and Secondary IDE Device Address:** These output signals are connected to the corresponding signals on the primary or secondary IDE connectors. They are used to indicate which byte in either the ATA command block or control block is being addressed.
PDD[15:0] **SDD[15:0]**	I/O	**Primary and Secondary IDE Device Data:** These signals directly drive the corresponding signals on the primary or secondary IDE connector. There is a weak internal pull-down resistor on PDD[7] and SDD[7].
PDDREQ **SDDREQ**	I	**Primary and Secondary IDE Device DMA Request:** These input signals are directly driven from the DRQ signals on the primary or secondary IDE connector. It is asserted by the IDE device to request a data transfer, and used in conjunction with the PCI bus master IDE function. They are not associated with any AT-compatible DMA channel. There is a weak internal pull-down resistor on these signals.
PDDACK# **SDDACK#**	O	**Primary and Secondary IDE Device DMA Acknowledge:** These signals directly drive the DAK# signals on the primary and secondary IDE connectors. Each signal is asserted by the 82801E C-ICH to indicate to the IDE DMA slave devices that a given data transfer cycle (assertion of DIOR# or DIOW#) is a DMA data transfer cycle. This signal is used in conjunction with the PCI bus master IDE function and are not associated with any AT-compatible DMA channel.
PDIOR# **/(PDWSTB** **/PRDMARDY#)** **SDIOR#** **/(SDWSTB** **/SRDMARDY#)**	O	**Primary and Secondary Disk I/O Read (PIO and Non-Ultra DMA):** This is the command to the IDE device that it may drive data on the PDD or SDD lines. Data is latched by the 82801E C-ICH on the deassertion edge of PDIOR# or SDIOR#. The IDE device is selected either by the ATA register file chip selects (PDCS1# or SDCS1#, PDCS3# or SDCS3#) and the PDA or SDA lines, or the IDE DMA acknowledge (PDDAK# or SDDAK#). **Primary and Secondary Disk Write Strobe (Ultra DMA Writes to Disk):** This is the data write strobe for writes to disk. When writing to disk, 82801E C-ICH drives valid data on rising and falling edges of PDWSTB or SDWSTB. **Primary and Secondary Disk DMA Ready (Ultra DMA Reads from Disk):** This is the DMA ready for reads from disk. When reading from disk, 82801E C-ICH deasserts PRDMARDY# or SRDMARDY# to pause burst data transfers.
PDIOW# **/(PDSTOP)** **SDIOW#** **/(SDSTOP)**	O	**Primary and Secondary Disk I/O Write (PIO and Non-Ultra DMA):** This is the command to the IDE device that it may latch data from the PDD or SDD lines. Data is latched by the IDE device on the deassertion edge of PDIOW# or SDIOW#. The IDE device is selected either by the ATA register file chip selects (PDCS1# or SDCS1#, PDCS3# or SDCS3#) and the PDA or SDA lines, or the IDE DMA acknowledge (PDDAK# or SDDAK#). **Primary and Secondary Disk Stop (Ultra DMA):** 82801E C-ICH asserts this signal (PDSTOP, SDSTOP) to terminate a burst.
PIORDY **/(PDRSTB** **/PWDMARDY#)** **SIORDY** **/(SDRSTB** **/SWDMARDY#)**	I	**Primary and Secondary I/O Channel Ready (PIO):** This signal keeps the strobe active (PDIOR# or SDIOR# on reads, PDIOW# or SDIOW# on writes) longer than the minimum width. It adds wait states to PIO transfers. **Primary and Secondary Disk Read Strobe (Ultra DMA Reads from Disk)**: When reading from disk, 82801E C-ICH latches data on rising and falling edges of this signal from the disk. **Primary and Secondary Disk DMA Ready (Ultra DMA Writes to Disk)**: When writing to disk, this is deasserted by the disk to pause burst data transfers.

A.7 LPC Interface

Table 133. LPC Interface Signals

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

A.8 Interrupt Interface

Table 134. Interrupt Signals

Name	Type	Description
SERIRQ	I/O	**Serial Interrupt Request:** This pin implements the serial interrupt protocol.
PIRQ[A:D]#	I/OD	**PCI Interrupt Requests:** In Non-APIC Mode the PIRQ*n*# signals can be routed to interrupts 3:7, 9:12, 14, or 15 as described in the Interrupt Steering section. Each PIRQ*n*# line has a separate Route Control Register. In APIC mode, these signals are connected to the internal I/O APIC in the following fashion: PIRQ[A]# is connected to IRQ16, PIRQ[B]# to IRQ17, PIRQ[C]# to IRQ18, and PIRQ[D]# to IRQ19. This frees the ISA interrupts.
PIRQ[E:F]# **PIRQ[G]#/GPIO[4]** **PIRQ[H]#/GPIO[5]**	I/OD	**PCI Interrupt Requests:** In Non-APIC Mode the PIRQ*n*# signals can be routed to interrupts 3:7, 9:12, 14 or 15 as described in the Interrupt Steering section. Each PIRQ*n*# line has a separate Route Control Register. In APIC mode, these signals are connected to the internal I/O APIC in the following fashion: PIRQ[E]# is connected to IRQ20, PIRQ[F]# to IRQ21, PIRQ[G]# to IRQ22, and PIRQ[H]# to IRQ23. This frees the ISA interrupts. If not needed for interrupts, PIRQ[H:G] can be used as GPIO.
IRQ[14:15]	I	**Interrupt Request 14:15:** These interrupt inputs are connected to the IDE drives. IRQ14 is used by the drives connected to the primary controller and IRQ15 is used by the drives connected to the secondary controller.
APICCLK	I	**APIC Clock:** The APIC clock runs at 33.333 MHz.
APICD[1:0]	I/OD	**APIC Data:** These bidirectional open drain signals are used to send and receive data over the APIC bus. As inputs, the data is valid on the rising edge of APICCLK. As outputs, new data is driven from the rising edge of the APICCLK.

A.9 USB Interface

Table 135. USB Interface Signals

Name	Type	Description
USBP0P **USBP0N** **USBP1P** **USBP1N**	I/O	**Universal Serial Bus Port 1:0 Differential:** These differential pairs are used to transmit Data/Address/Command signals for ports 0 and 1.
OC[1:0]#	I	**Overcurrent Indicators:** These signals set corresponding bits in the USB controllers to indicate that an overcurrent condition has occurred.

A.10 Power Signals

Table 136. Power Signals

Name	Type	Description
PWROK	I	**Power OK:** When asserted, PWROK is an indication to the 82801E C-ICH that core power and PCICLK have been stable for at least 1 ms. PWROK can be driven asynchronously. When PWROK is negated, the 82801E C-ICH asserts PCIRST#.
RSM_PWROK	I	**Resume Well Power OK:** When asserted, this signal is an indication to the 82801E C-ICH that the resume well power has been stable for at least 10 ms. **NOTE:** The 82801E C-ICH does not use the resume well power OK signal.
RSMRST#	I	**Resume Well Reset:** RSMRST# is used for resetting the resume power plane logic. **NOTE:** The 82801E C-ICH does not use the resume well reset signal.
VRMPWRGD	I	**VRM Power Good:** VRMPWRGD should be connected to be the processor's VRM Power Good. This signal should be ANDed with the ATX power supply's PWROK signal.

A.11 Processor Interface

Table 137. Processor Interface Signals (Sheet 1 of 2)

Name	Type	Description
A20M#	O	**Mask A20:** A20M# goes active based on setting the appropriate bit in the Port 92h register, or based on the A20GATE signal. **Speed Strap:** During the reset sequence, 82801E C-ICH drives A20M# high if the corresponding bit is set in the FREQ_STRP register.
CPUSLP#	O	**Processor Sleep:** This signal puts the processor into a state that saves substantial power compared to Stop-Grant state. However, during that time, no snoops occur. **NOTE:** The 82801E C-ICH does not support Sleep states. This signal must be pulled up through an 8.2 KΩ resistor to 3.3 V.
FERR#	I	**Numeric Coprocessor Error:** This signal is tied to the coprocessor error signal on the processor. FERR# is only used if the 82801E C-ICH coprocessor error reporting function is enabled in the General Control Register (Device 31:Function 0, Offset D0, bit 13). When FERR# is asserted, the 82801E C-ICH generates an internal IRQ13 to its interrupt controller unit. It is also used to gate the IGNNE# signal to ensure that IGNNE# is not asserted to the processor unless FERR# is active. FERR# requires an external weak pull-up to ensure a high level when the coprocessor error function is disabled.
IGNNE#	O	**Ignore Numeric Error:** This signal is connected to the ignore error pin on the processor. IGNNE# is only used if the 82801E C-ICH coprocessor error reporting function is enabled in the General Control Register (Device 31:Function 0, Offset D0, bit 13). When FERR# is active, indicating a coprocessor error, a write to the Coprocessor Error Register (F0h) causes the IGNNE# to be asserted. IGNNE# remains asserted until FERR# is negated. If FERR# is not asserted when the Coprocessor Error Register is written, the IGNNE# signal is not asserted. **Speed Strap:** During the reset sequence, 82801E C-ICH drives IGNNE# high if the corresponding bit is set in the FREQ_STRP register.
INIT#	O	**Initialization:** INIT# is asserted by the 82801E C-ICH for 16 PCI clocks to reset the processor. 82801E C-ICH can be configured to support processor BIST. In that case, INIT# will be active when PCIRST# is active.
INTR	O	**Processor Interrupt:** INTR is asserted by the 82801E C-ICH to signal the processor that an interrupt request is pending and needs to be serviced. It is an asynchronous output and normally driven low. **Speed Strap:** During the reset sequence, 82801E C-ICH drives INTR high if the corresponding bit is set in the FREQ_STRP register.
NMI	O	**Non-Maskable Interrupt:** NMI is used to force a non-maskable interrupt to the processor. The 82801E C-ICH can generate an NMI when either SERR# or IOCHK# is asserted. The processor detects an NMI when it detects a rising edge on NMI. NMI is reset by setting the corresponding NMI source enable/disable bit in the NMI Status and Control Register. **Speed Strap:** During the reset sequence, 82801E C-ICH drives NMI high if the corresponding bit is set in the FREQ_STRP register.
SMI#	O	**System Management Interrupt:** SMI# is an active low output synchronous to PCICLK. It is asserted by the 82801E C-ICH in response to one of many enabled hardware or software events.
STPCLK#	O	**Stop Clock Request:** STPCLK# is an active low output synchronous to PCICLK. It is asserted by the 82801E C-ICH in response to one of many hardware or software events. When the processor samples STPCLK# asserted, it responds by stopping its internal clock.

Table 137. Processor Interface Signals (Sheet 2 of 2)

Name	Type	Description
RCIN#	I	**Keyboard Controller Reset Processor:** The keyboard controller can generate INIT# to the processor. This saves the external OR gate with the 82801E C-ICH's other sources of INIT#. When the 82801E C-ICH detects the assertion of this signal, INIT# is generated for 16 PCI clocks.
A20GATE	I	**A20 Gate:** This signal is from the keyboard controller. It acts as an alternative method to force the A20M# signal active. A20GATE eliminates the need for the external OR gate needed with various other PCIsets.
CPUPWRGD	OD	**Processor Power Good:** This signal should be connected to the processor's PWRGOOD input. This is an open-drain output signal (external pull-up resistor required) that represents a logical AND of the 82801E C-ICH's PWROK and VRMPWRGD signals.

A.12 SMBus Interface

Table 138. SMBus Interface Signals

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

A.13 System Management Interface

Table 139. System Management Interface Signals

Name	Type	Description
INTRUDER#	I	**Intruder Detect:** This signal can be set to disable the system if the is opened. This signal's status is readable, so it can be used like a GPI if the Intruder Detection is not needed.
SMLINK[1:0]	I/OD	**System Management Link:** These signals are an SMBus link to an optional external system management ASIC or LAN controller. External pull-ups are required. Note that SMLINK[0] corresponds to an SMBus Clock signal and SMLINK[1] corresponds to an SMBus Data signal.

A.14 Real Time Clock Interface

Table 140. Real Time Clock Interface

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

A.15 Other Clocks

Table 141. Other Clocks

Name	Type	Description
CLK14	I	**Oscillator Clock:** CLK14 is used for 8254 timers and runs at 14.31818 MHz.
CLK48	I	**48 MHz Clock:** CLK48 is used to for the USB controller and runs at 48 MHz.
CLK66 (HLCLK)	I	**66 MHz Clock (HLCLK):** CLK66 is used for the hub interface and runs at 66 MHz.

A.16 Universal Asynchronous Receive and Transmit (UART0, UART1)

Table 142. Universal Asynchronous Receive And Transmit (UART 0, 1) (Sheet 1 of 2)

Signal Name	Type	Description
UART_CLK	I	**Input clock to the SIU**. This clock is passed to the baud clock generation logic of each UART in the SIU.
SIU0_CTS# **SIU1_CTS#**	I	**Clear to Send:** Active low, this pin indicates that data can be exchanged between the 82801E C-ICH and the external interface. These pins have no effect on the transmitter. **NOTE:** These pins could be used as Modem Status Inputs whose condition can be tested by the processor by reading bit 4 (CTS) of the Modem Status register (MSR). Bit 4 is the complement of the CTS# signal. Bit 0 (DCTS) of the MSR indicates whether the CTS# input has changed state since the previous reading of the MSR. When the CTS bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU0_DCD# **SIU1_DCD#**	I	**Data Carrier Detect for UART0 and UART1:** Active low, this pin indicates that data carrier has been detected by the external agent. **NOTE:** These pins are Modem Status Inputs whose condition can be tested by the processor by reading bit 7 (DCD) of the Modem Status register (MSR). Bit 7 is the complement of the DCD# signal. Bit 3 (DDCD) of the MSR indicates whether the DCD# input has changed state since the previous reading of the MSR. When the DCD bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.

Table 142. Universal Asynchronous Receive And Transmit (UART 0, 1) (Sheet 2 of 2)

Signal Name	Type	Description
SIU0_DSR# **SIU1_DSR#**	I	**Data Set Ready for UART0 and UART1:** Active low, this pin indicates that the external agent is ready to communicate with 82801E C-ICH UARTs. These pins have no effect on the transmitter. **NOTE:** These pins could be used as Modem Status Input whose condition can be tested by the processor by reading bit 5 (DSR) of the Modem Status register. Bit 5 is the complement of the DSR# signal. Bit 1 (DDSR) of the Modem status register (MSR) indicates whether the DSR# input has changed state since the previous reading of the MSR. When the DSR bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU0_DTR# **SIU1_DTR#**	O	**Data Terminal Ready for UART0 and UART1:** When low these pins informs the modem or data set that 82801E C-ICH UART0 and UART1 are ready to establish a communication link. The DTR#x(x=0,1) output signals can be set to an active low by programming the DTRx (x-0,1) (bit0) of the Modem control register to a logic ‘1’. A Reset operation sets this signal to its inactive state (logic ‘1’). LOOP mode operation holds this signal in its inactive state.
SIU0_RI# **SIU1_RI#**	I	**Ring Indicator for UART0 and UART1:** Active low, this pin indicates that a telephone ringing signal has been received by the external agent. **NOTE:** These pins are Modem Status Input whose condition can be tested by the processor by reading bit 6 (RI) of the Modem Status register (MSR). Bit 6 is the complement of the RI# signal. Bit 2 (TERI) of the MSR indicates whether the DCD# input has changed state since the previous reading of the MSR. When the RI bit of the MSR changes state an interrupt is generated if the Modem Status Interrupt is enabled.
SIU0_RTS# **SIU1_RTS#**	O	**Request to Send for UART0 and UART1:** When low these pins informs the modem or data set that 82801E C-ICH UART0 and UART1 are ready to establish a communication link. The RTS#x(x=0,1) output signals can be set to an active low by programming the RTSx (x-0,1) (bit1) of the Modem control register to a logic ‘1’. A Reset operation sets this signal to its inactive state (logic ‘1’). LOOP mode operation holds this signal in its inactive state.
SIU0_RXD **SIU1_RXD**	I	**Serial Inputs for UART0 and UART1:** Serial data input from device pin to the receive port.
SIU0_TXD **SIU1_TXD**	O	**Serial Output for UART0 and UART1:** Serial data output to the communication peripheral/modem or data set. Upon reset, the TXD pins will be set to MARKING condition (logic ‘1’ state).

A.17 SIU LPC Interface

Table 143. SIU Interface

Signal Name	Type	Description
SIU_LAD[3:0]	I/O	**SIU LPC Multiplexed Command, Address, Data**: Internal pull-ups are provided.
SIU_LCLK	I	**SIU LPC clock input to SIU**: 33 MHz LPC clock.
SIU_LDRQ#	O	**SIU LPC Serial DMA/Master Request Output**: Used by SIU devices to indicate a DMA request. **NOTE:** These signals have weak internal pull-up resistors to avoid external glue.
SIU_LFRAME#	I	**SIU LPC Frame**: Indicates the start of an LPC cycle, or an abort.
SIU_RESET#	I	**SIU RESET**: This signal should be tied to PCI RESET.
SIU_SERIRQ	I/O	**SIU Serial IRQ Input**: This pin receives the serial interrupt protocol from external devices. Pull up if unused.

A.18 Miscellaneous Signals

Table 144. Miscellaneous Signals

Name	Type	Description
HL[11]	I	No pull-up required. Use a no-stuff or a test point for NAND tree testing.
RTCRST#	I	**RTC Reset:** When asserted, this signal resets register bits in the RTC well and sets the RTC_PWR_STS bit (bit 2 in GEN_PMCON3 register). This signal is also used to enter the test modes documented in the datasheet. **NOTE:** Clearing CMOS in an 82801E C-ICH-based platform can be done by using a jumper on RTCRST# or GPI, or using SAFEMODE strap. Implementations should not attempt to clear CMOS by using a jumper to pull VccRTC low.
SPKR	O	**Speaker:** The SPKR signal is the output of counter 2 and is internally ANDed with Port 61h bit 1 to provide Speaker Data Enable. This signal drives an external speaker driver device, which in turn drives the system speaker. Upon PCIRST#, its output state is 1. **NOTE:** SPKR is sampled at the rising edge of PWROK as a functional strap. See the datasheet for more details.
TP0	I	**Test Point 0:** This signal must have an external pull-up to Vcc3_3.
THRM#	I	**Thermal Alarm:** THRM# is an active low signal generated by external hardware to start the hardware clock throttling mode. This signal can also generate an SMI# or an SCI.
RI#	I	**Ring Indicate:** From the modem interface. This signal can be enabled as a wake event; this is preserved across power failures.
RESERVED1 **RESERVED2**	—	This signal must have an external pull up to Vcc3_3.
SUSCLK	O	**Suspend Clock:** This signal is an output of the RTC generator circuit and is used by other chips for the refresh clock.

Table 144. Miscellaneous Signals

Name	Type	Description
TP1	I	**Test Point 1:** Route to a test point with option to jumper to Vcc1_8. Used for NAND tree testing. Otherwise jumper to Vcc1_8.
TP2	I	**Test Point 2:** Route to a test point with option to jumper to V_{SS}. Used for NAND tree testing. Otherwise jumper to V_{SS}.
TP3	I	**Test Point 3:** Route to a test point with option to jumper to V_{SS}. Used for NAND tree testing. Otherwise jumper to V_{SS}.

A.19 General Purpose I/O

Table 145. General Purpose I/O Signals

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

A.20 Power and Ground

Table 146. Power and Ground Signals

Name	Description
HUBREF	0.9 V reference for the hub interface.
V5REF	Reference for 5 V tolerance on Core well inputs.
VBIAS	RTC well bias voltage. The DC reference voltage applied to this pin sets a current that is mirrored throughout the oscillator and buffer circuitry.
Vcc1_8	1.8 V supply for Core well logic.
Vcc3_3	3.3 V supply for Core well I/O buffers.
VccRTC	3.3 V (can drop to 2.0 V min. in G3 state) supply for the RTC well. This power is not expected to be shut off unless the RTC battery is removed or completely drained. **NOTE:** Implementations should not attempt to clear CMOS by using a jumper to pull VccRTC low. Clearing CMOS in an 82801E C-ICH-based platform can be done by using a jumper on RTCRST# or GPI, or using SAFEMODE strap.
V_CPU_IO	Powered by the same supply as the processor I/O voltage. This supply is used to drive the processor interface outputs.
Vss	Ground.

I/O Register Index B

Table 147. Intel® 82801E C-ICH Fixed I/O Registers (Sheet 1 of 5)

Register Name	Port	Location
Channel 0 DMA Base & Current Address Register	00h	Section 8.2.1, “DMABASE_CA—DMA Base and Current Address Registers” on page 227
Channel 0 DMA Base & Current Count Register	01h	Section 8.2.2, “DMABASE_CC—DMA Base and Current Count Registers” on page 228
Channel 1 DMA Base & Current Address Register	02h	Section 8.2.1, “DMABASE_CA—DMA Base and Current Address Registers” on page 227
Channel 1 DMA Base & Current Count Register	03h	Section 8.2.2, “DMABASE_CC—DMA Base and Current Count Registers” on page 228
Channel 2 DMA Base & Current Address Register	04h	Section 8.2.1, “DMABASE_CA—DMA Base and Current Address Registers” on page 227
Channel 2 DMA Base & Current Count Register	05h	Section 8.2.2, “DMABASE_CC—DMA Base and Current Count Registers” on page 228
Channel 3 DMA Base & Current Address Register	06h	Section 8.2.1, “DMABASE_CA—DMA Base and Current Address Registers” on page 227
Channel 3 DMA Base & Current Count Register	07h	Section 8.2.2, “DMABASE_CC—DMA Base and Current Count Registers” on page 228
Channel 0–3 DMA Command Register Channel 0–3 DMA Status Register	08h	Section 8.2.4, “DMACMD—DMA Command Register” on page 229 Section 8.2.5, “DMASTS—DMA Status Register” on page 229
Channel 0–3 DMA Write Single Mask Register	0Ah	Section 8.2.6, “DMA_WRSMSK—DMA Write Single Mask Register” on page 230
Channel 0–3 DMA Channel Mode Register	0Bh	Section 8.2.7, “DMACH_MODE—DMA Channel Mode Register” on page 230
Channel 0–3 DMA Clear Byte Pointer Register	0Ch	Section 8.2.8, “DMA Clear Byte Pointer Register” on page 231
Channel 0–3 DMA Master Clear Register	0Dh	Section 8.2.9, “DMA Master Clear Register” on page 231
Channel 0–3 DMA Clear Mask Register	0Eh	Section 8.2.10, “DMA_CLMSK—DMA Clear Mask Register” on page 231
Channel 0–3 DMA Write All Mask Register	0Fh	Section 8.2.11, “DMA_WRMSK—DMA Write All Mask Register” on page 232
Aliased at 00h–0Fh	10h–1Fh	
Master PIC ICW1 Init. Cmd Word 1 Register Master PIC OCW2 Op Ctrl Word 2 Register Master PIC OCW3 Op Ctrl Word 3 Register	20h	Section 8.4.2, “ICW1—Initialization Command Word 1 Register” on page 237 Section 8.4.8, “OCW2—Operational Control Word 2 Register” on page 240 Section 8.4.9, “OCW3—Operational Control Word 3 Register” on page 241

Table 147. Intel® 82801E C-ICH Fixed I/O Registers (Sheet 2 of 5)

Register Name	Port	Location
Master PIC ICW2 Init. Cmd Word 2 Register Master PIC ICW3 Init. Cmd Word 3 Register Master PIC ICW4 Init. Cmd Word 4 Register Master PIC OCW1 Op Ctrl Word 1 Register	21h	Section 8.4.3, “ICW2—Initialization Command Word 2 Register” on page 238 Section 8.4.4, “ICW3—Master Controller Initialization Command Word 3 Register” on page 238 Section 8.4.6, “ICW4—Initialization Command Word 4 Register” on page 239 Section 8.4.7, “OCW1—Operational Control Word 1 (Interrupt Mask) Register” on page 239
Aliased at 20h–21h	24h–25h	
Aliased at 20h–21h	28h–29h	
Aliased at 20h–21h	24h–25h	
Aliased at 20h–21h	2Ch–2Dh	
Aliased at 20h–21h	30h–31h	
Aliased at 20h–21h	34h–35h	
Aliased at 20h–21h	38h–39h	
Aliased at 20h–21h	3Ch–3Dh	
Counter 0 Interval Time Status Byte Format Counter 0 Counter Access Port Register	40h	Section 8.3.2, “SBYTE_FMT—Interval Timer Status Byte Format Register” on page 235 Section 8.3.3, “Counter Access Ports Register” on page 236
Counter 1 Interval Time Status Byte Format Counter 1 Counter Access Port Register	41h	Section 8.3.2, “SBYTE_FMT—Interval Timer Status Byte Format Register” on page 235 Section 8.3.3, “Counter Access Ports Register” on page 236
Counter 2 Interval Time Status Byte Format Counter 2 Counter Access Port Register	42h	Section 8.3.2, “SBYTE_FMT—Interval Timer Status Byte Format Register” on page 235 Section 8.3.3, “Counter Access Ports Register” on page 236
Timer Control Word Register Timer Control Word Register Read Back Counter Latch Command	43h	Section 8.3.1, “TCW—Timer Control Word Register” on page 233 Section 8.3.1.1, “RDBK_CMD—Read Back Command” on page 233 Section 8.3.1.2, “LTCH_CMD—Counter Latch Command” on page 234
Aliased at 40h–43h	50h–53h	
NMI Status and Control Register	61h	Section 8.7.1, “NMI_SC—NMI Status and Control Register” on page 254
NMI Enable Register	70h	Section 8.7.2, “NMI_EN—NMI Enable (and Real Time Clock Index)” on page 254
Real-Time Clock (Standard RAM) Index Register	70h	Table 98, “RTC (Standard) RAM Bank” on page 250 Section 8.7.2, “NMI_EN—NMI Enable (and Real Time Clock Index)” on page 254
Real-Time Clock (Standard RAM) Target Register	71h	Table 98, “RTC (Standard) RAM Bank” on page 250
Extended RAM Index Register	72h	
Extended RAM Target Register	73h	

Table 147. Intel® 82801E C-ICH Fixed I/O Registers (Sheet 3 of 5)

Register Name	Port	Location
Aliased at 70h–71h	74h–75h	Aliased if U128E bit in RTC Configuration Register is enabled Section 8.1.24, “RTC_CONF—RTC Configuration Register (LPC I/F—D31:F0)” on page 217
Aliased at 72h–73h or 70h–71h	76h–77h	Aliased to 70h–71h if U128E bit in RTC Configuration Register is enabled Section 8.1.24, “RTC_CONF—RTC Configuration Register (LPC I/F—D31:F0)” on page 217
Channel 2 DMA Memory Low Page Register	81h	Section 8.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 228
Channel 3 DMA Memory Low Page Register	82h	Section 8.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 228
Channel 1 DMA Memory Low Page Register	83h	Section 8.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 228
Reserved Page Registers	84h–86h	
Channel 0 DMA Memory Low Page Register	87h	Section 8.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 228
Reserved Page Register	88h	
Channel 6 DMA Memory Low Page Register	89h	Section 8.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 228
Channel 7 DMA Memory Low Page Register	8Ah	Section 8.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 228
Channel 5 DMA Memory Low Page Register	8Bh	Section 8.2.3, “DMAMEM_LP—DMA Memory Low Page Registers” on page 228
Reserved Page Registers	8Ch–8Eh	
Refresh Low Page Register	8Fh	
Aliased at 81h–8Fh	91h–9Fh (except 92h)	
Fast A20 and INIT Register	92h	Section 8.7.3, “PORT92—Fast A20 and Init Register” on page 255
Slave PIC ICW1 Init. Cmd Word 1 Register Slave PIC OCW2 Op Ctrl Word 2 Register Slave PIC OCW3 Op Ctrl Word 3 Register	A0h	Section 8.4.2, “ICW1—Initialization Command Word 1 Register” on page 237 Section 8.4.8, “OCW2—Operational Control Word 2 Register” on page 240 Section 8.4.9, “OCW3—Operational Control Word 3 Register” on page 241
Slave PIC ICW2 Init. Cmd Word 2 Register Slave PIC ICW3 Init. Cmd Word 3 Register Slave PIC ICW4 Init. Cmd Word 4 Register Slave PIC OCW1 Op Ctrl Word 1 Register	A1	Section 8.4.3, “ICW2—Initialization Command Word 2 Register” on page 238 Section 8.4.4, “ICW3—Master Controller Initialization Command Word 3 Register” on page 238 Section 8.4.6, “ICW4—Initialization Command Word 4 Register” on page 239 Section 8.4.7, “OCW1—Operational Control Word 1 (Interrupt Mask) Register” on page 239
Aliased at A0h–A1h	A4h–A5h	
Aliased at A0h–A1h	A8h–A9h	
Aliased at A0h–A1h	ACh–ADh	

Table 147. Intel® 82801E C-ICH Fixed I/O Registers (Sheet 4 of 5)

Register Name	Port	Location
Aliased at A0h–A1h	B0h–B1h	
Advanced Power Management Control Port Register	B2h	Section 8.8.2.1, “APM_CNT—Advanced Power Management Control Port Register” on page 261
Advanced Power Management Status Port Register	B3h	Section 8.8.2.2, “APM_STS—Advanced Power Management Status Port Register” on page 261
Aliased at A0h–A1h	B4h–B5h	
Aliased at A0h–A1h	B8h–B9h	
Aliased at A0h–A1h	BCh–BDh	
Channel 4 DMA Base & Current Address Register	C0h	Section 8.2.1, “DMABASE_CA—DMA Base and Current Address Registers” on page 227
Aliased at C0h	C1h	
Channel 4 DMA Base & Current Count Register	C2h	Section 8.2.2, “DMABASE_CC—DMA Base and Current Count Registers” on page 228
Aliased at C2h	C3h	
Channel 5 DMA Base & Current Address Register	C4h	Section 8.2.1, “DMABASE_CA—DMA Base and Current Address Registers” on page 227
Aliased at C4h	C5h	
Channel 5 DMA Base & Current Count Register	C6h	Section 8.2.2, “DMABASE_CC—DMA Base and Current Count Registers” on page 228
Aliased at C6h	C7h	
Channel 6 DMA Base & Current Address Register	C8h	Section 8.2.1, “DMABASE_CA—DMA Base and Current Address Registers” on page 227
Aliased at C8h	C9h	
Channel 6 DMA Base & Current Count Register	CAh	Section 8.2.2, “DMABASE_CC—DMA Base and Current Count Registers” on page 228
Aliased at CAh	CBh	
Channel 7 DMA Base & Current Address Register	CCh	Section 8.2.1, “DMABASE_CA—DMA Base and Current Address Registers” on page 227
Aliased at CCh	CDh	
Channel 7 DMA Base & Current Count Register	CEh	Section 8.2.2, “DMABASE_CC—DMA Base and Current Count Registers” on page 228
Aliased at CEh	CFh	
Channel 4–7 DMA Command Register Channel 4–7 DMA Status Register	D0h	Section 8.2.4, “DMACMD—DMA Command Register” on page 229 Section 8.2.5, “DMASTS—DMA Status Register” on page 229
Aliased at D0h	D1h	
Channel 4–7 DMA Write Single Mask Register	D4h	Section 8.2.6, “DMA_WRSMSK—DMA Write Single Mask Register” on page 230
Aliased at D4h	D5h	
Channel 4–7 DMA Channel Mode Register	D6h	Section 8.2.7, “DMACH_MODE—DMA Channel Mode Register” on page 230
Aliased at D6h	D7h	

Table 147. Intel® 82801E C-ICH Fixed I/O Registers (Sheet 5 of 5)

Register Name	Port	Location
Channel 4–7 DMA Clear Byte Pointer Register	D8h	Section 8.2.8, “DMA Clear Byte Pointer Register” on page 231
Aliased at D8h	D9h	
Channel 4–7 DMA Master Clear Register	DAh	Section 8.2.9, “DMA Master Clear Register” on page 231
Aliased at DAh	DBh	
Channel 4–7 DMA Clear Mask Register	DCh	Section 8.2.10, “DMA_CLMSK—DMA Clear Mask Register” on page 231
Aliased at DCh	DEh	
Channel 4–7 DMA Write All Mask Register	DEh	Section 8.2.11, “DMA_WRMSK—DMA Write All Mask Register” on page 232
Aliased at DEh	DFh	
Coprocessor Error Reigster	F0h	Section 8.7.4, “COPROC_ERR—Coprocessor Error Register” on page 255
PIO Mode Command Block Offset for Secondary Drive	170h–177h	See ATA Specification for detailed register description
PIO Mode Command Block Offset for Primary Drive	1F0h–1F7h	See ATA Specification for detailed register description
PIO Mode Control Block Offset for Secondary Drive	376h	See ATA Specification for detailed register description
PIO Mode Control Block Offset for Primary Drive	3F6h	See ATA Specification for detailed register description
Master PIC Edge/Level Triggered Register	4D0h	Section 8.4.10, “ELCR1—Master Controller Edge/Level Triggered Register” on page 242
Slave PIC Edge/Level Triggered Register	4D1h	Section 8.4.11, “ELCR2—Slave Controller Edge/Level Triggered Register” on page 243
Reset Control Register	CF9h	Section 8.7.5, “RST_CNT—Reset Control Register” on page 255

NOTE: When the POS_DEC_EN bit is set, additional I/O ports get positively decoded by the 82801E C-ICH. Refer to through for a listing of these ranges.

Table 148. Intel® 82801E C-ICH Variable I/O Registers (Sheet 1 of 4)

Register Name	Offset	Location
LAN Control/Status Registers (CSR) may be mapped to either I/O space or memory space. LAN CSR at CSR_IO_BASE + Offset or CSR_MEM_BASE + Offset. CSR_MEM_BASE set in Section 6.1.11, "CSR_MEM_BASE CSR—Memory-Mapped Base Address Register (LAN Controller—B1:D8/D9:F0)" on page 174 CSR_IO_BASE set in Section 6.1.12, "CSR_IO_BASE—CSR I/O-Mapped Base Address Register (LAN Controller—B1:D8/D9:F0)" on page 175		
SCB Status Word	01h–00h	Section 6.2.1, "System Control Block Status Word Register" on page 180
SCB Command Word	03h–02h	Section 6.2.2, "System Control Block Command Word Register" on page 181
SCB General Pointer	07h–04h	Section 6.2.3, "System Control Block General Pointer Register" on page 183
PORT	OBh–08h	Section 6.2.4, "PORT Register" on page 183
EEPROM Control Register	0Fh–0Eh	Section 6.2.5, "EEPROM Control Register" on page 184
MDI Control Register	13h–10h	Section 6.2.6, "Management Data Interface (MDI) Control Register" on page 185
Receive DMA Byte Count	17h–14h	Section 6.2.7, "Receive DMA Byte Count Register" on page 185
Early Receive Interrupt	18h	Section 6.2.8, "Early Receive Interrupt Register" on page 186
Flow Control Register	1Ah–19h	Section 6.2.9, "Flow Control Register" on page 186
PMDR	1Bh	Section 6.2.10, "Power Management Driver (PMDR) Register" on page 187
General Control	1Ch	Section 6.2.11, "General Control Register" on page 188
General Status	1Dh	Section 6.2.12, "General Status Register" on page 188
Power Management I/O Registers at PMBASE+Offset PMBASE set in Section 8.1.10, "PMBASE—ACPI Base Address (LPC I/F—D31:F0)" on page 210		
PM1 Status	00–01h	Section 8.8.3.1, "PM1_STS—Power Management 1 Status Register" on page 263
PM1 Enable	02–03h	Section 8.8.3.2, "PM1_EN—Power Management 1 Enable Register" on page 264
PM1 Control	04–07h	Section 8.8.3.3, "PM1_CNT—Power Management 1 Control Register" on page 265
PM1 Timer	08–0Bh	Section 8.8.3.4, "PM1_TMR—Power Management 1 Timer Register" on page 265
Processor Control	10h–13h	Section 8.8.3.5, "PROC_CNT—Processor Control Register" on page 266
General Purpose Event 0 Status	28–29h	Section 8.8.3.6, "GPE0_STS—General Purpose Event 0 Status Register" on page 267
General Purpose Event 0 Enables	2A–2Bh	Section 8.8.3.7, "GPE0_EN—General Purpose Event 0 Enables Register" on page 268
General Purpose Event 1 Status	2C–2D	Section 8.8.3.8, "GPE1_STS—General Purpose Event 1 Status Register" on page 269

Table 148. Intel® 82801E C-ICH Variable I/O Registers (Sheet 2 of 4)

Register Name	Offset	Location
General Purpose Event 1 Enables	2E–2F	Section 8.8.3.9, “GPE1_EN—General Purpose Event 1 Enable Register” on page 270
SMI# Control and Enable	30–31h	Section 8.8.3.10, “SMI_EN—SMI Control and Enable Register” on page 270
SMI Status Register	34–35h	Section 8.8.3.11, “SMI_STS—SMI Status Register” on page 272
Monitor SMI Status	40h	Section 8.8.3.12, “MON_SMI—Device Monitor SMI Status and Enable Register” on page 273
Device Activity Status	44h	Section 8.8.3.13, “DEVACT_STS—Device Activity Status Register” on page 274
Device Trap Enable	48h	Section 8.8.3.14, “DEVTRAP_EN—Device Trap Enable Register” on page 275
Bus Address Tracker	4Ch	Section 8.8.3.15, “BUS_ADDR_TRACK—Bus Address Tracker Register” on page 276
Bus Cycle Tracker	4Eh	Section 8.8.3.16, “BUS_CYC_TRACK—Bus Cycle Tracker Register” on page 276
TCO I/O Registers at TCOBASE + Offset TCOBASE = PMBASE + 40h PMBASE is set in Section 8.1.10, “PMBASE—ACPI Base Address (LPC I/F—D31:F0)” on page 210		
TCO_RLD: TCO Timer Reload and Current Value	00h	Section 8.9.2, “TCO1_RLD—TCO Timer Reload and Current Value Register” on page 277
TCO_TMR: TCO Timer Initial Value	01h	Section 8.9.3, “TCO1_TMR—TCO Timer Initial Value Register” on page 278
TCO_DAT_IN: TCO Data In	02h	Section 8.9.4, “TCO1_DAT_IN—TCO Data In Register” on page 278
TCO_DAT_OUT: TCO Data Out	03h	Section 8.9.5, “TCO1_DAT_OUT—TCO Data Out Register” on page 278
TCO1_STS: TCO Status	04h–05h	Section 8.9.6, “TCO1_STS—TCO1 Status Register” on page 278
TCO2_STS: TCO Status	06h–07h	Section 8.9.7, “TCO2_STS—TCO2 Status Register” on page 280
TCO1_CNT: TCO Control	08h–09h	Section 8.9.8, “TCO1_CNT—TCO1 Control Register” on page 281
TCO2_CNT: TCO Control	0Ah–0Bh	Section 8.9.9, “TCO2_CNT—TCO2 Control Register” on page 282
GPIO I/O Registers at GPIOBASE + Offset GPIOBASE is set in Section 8.1.14, “GPIOBASE—GPIO Base Address (LPC I/F—D31:F0)” on page 212		
GPIO Use Select	00–03h	Section 8.10.2, “GPIO_USE_SEL—GPIO Use Select Register” on page 285
GPIO Input/Output Select	04–07h	Section 8.10.3, “GP_IO_SEL—GPIO Input/Output Select Register” on page 286
GPIO Level for Input or Output	0C–0Fh	Section 8.10.4, “GP_LVL—GPIO Level for Input or Output Register” on page 287
GPIO Blink Enable	18–1Bh	Section 8.10.5, “GPO_BLINK—GPO Blink Enable Register” on page 288
GPIO Signal Invert	2C–2Fh	Section 8.10.6, “GPI_INV—GPIO Signal Invert Register” on page 289

Table 148. Intel® 82801E C-ICH Variable I/O Registers (Sheet 3 of 4)

Register Name	Offset	Location
BMIDE I/O Registers at BM_BASE + Offset BM_BASE is set at Section 9.1.10, “BM_BASE—Bus Master Base Address Register (IDE—D31:F1)” on page 294		
Command Register Primary	00h	Section 9.2.1, “BMIC[P,S]—Bus Master IDE Command Register” on page 301
Status Register Primary	02h	Section 9.2.2, “BMIS[P,S]—Bus Master IDE Status Register” on page 302
Descriptor Table Pointer Primary	04h–07h	Section 9.2.3, “BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register” on page 302
Command Register Secondary	08h	Section 9.2.1, “BMIC[P,S]—Bus Master IDE Command Register” on page 301
Status Register Secondary	0Ah	Section 9.2.2, “BMIS[P,S]—Bus Master IDE Status Register” on page 302
Descriptor Table Pointer Secondary	0Ch–0Fh	Section 9.2.3, “BMID[P,S]—Bus Master IDE Descriptor Table Pointer Register” on page 302
USB I/O Registers at Base Address + Offset **USB Base Address is set at Section 10.1.10, “BASE—Base Address Register (USB—D31:F2)” on page 306**		
USB Command Register	00h–01h	Section 10.2.1, “USBCMD—USB Command Register” on page 311
USB Status Register	02h–03h	Section 10.2.2, “USBSTA—USB Status Register” on page 314
USB Interrupt Enable	04h–05h	Section 10.2.3, “USBINTR—Interrupt Enable Register” on page 315
USB Frame Number	06h–07h	Section 10.2.4, “FRNUM—Frame Number Register” on page 315
USB Frame List Base Address	08h–0Bh	Section 10.2.5, “FRBASEADD—Frame List Base Address” on page 316
USB Start of Frame Modify	0Ch	Section 10.2.6, “SOFMOD—Start of Frame Modify Register” on page 316
Port 0, 2 Status/Control	10h–11h	Section 10.2.7, “PORTSC[0,1]—Port Status and Control Register” on page 317
Port 1, 3 Status/Control	12h–13h	Section 10.2.7, “PORTSC[0,1]—Port Status and Control Register” on page 317
Loop Back Test Data	18h	
SMBus I/O Registers at SMB_BASE + Offset SMB_BASE is set at Section 11.1.9, “SMB_BASE—SMBus Base Address Register (SMBUS—D31:F3)” on page 322		
Host Status	00h	Section 11.2.1, “HST_STS—Host Status Register” on page 325
Host Control	02h	Section 11.2.2, “HST_CNT—Host Control Register” on page 326
Host Command	03h	Section 11.2.3, “HST_CMD—Host Command Register” on page 328
Transmit Slave Address	04h	Section 11.2.4, “XMIT_SLVA—Transmit Slave Address Register” on page 328

Table 148. Intel® 82801E C-ICH Variable I/O Registers (Sheet 4 of 4)

Register Name	Offset	Location
Host Data 0	05h	Section 11.2.5, “HST_D0—Data 0 Register” on page 328
Host Data 1	06h	Section 11.2.6, “HST_D1—Data 1 Register” on page 328
Block Data Byte	07h	Section 11.2.7, “BLOCK_DB—Block Data Byte Register” on page 329
Receive Slave Address	09h	Section 11.2.8, “RCV_SLVA—Receive Slave Address Register” on page 329
Receive Slave Data	0Ah	Section 11.2.9, “SLV_DATA—Receive Slave Data Register” on page 329

Table 149. Serial I/O Unit Registers

Register Name	Offset	Location
Receive Buffer Register (RBR)	Base (DLAB=0)	Section 12.5.3.1, “Receive Buffer Register (RBR)” on page 341
Transmit Holding Register (THR)	Base (DLAB=0)	Section 12.5.3.2, “Transmit Holding Register (THR)” on page 341
Interrupt Enable Register (IER)	Base + 01H (DLAB=0)	Section 12.5.3.3, “Interrupt Enable Register (IER)” on page 342
Interrupt Identification Register (IIR)	Base + 02H	Section 12.5.3.4, “Interrupt Identification Register (IIR)” on page 343
FIFO Control Register (FCR)	Base + 02H	Section 12.5.3.5, “FIFO Control Register (FCR)” on page 345
Line Control Register (LCR)	Base + 03H	Section 12.5.3.6, “Line Control Register (LCR)” on page 346
Line Status Register (LSR)	Base + 05H	Section 12.5.3.7, “Line Status Register (LSR)” on page 348
Modem Control Register (MCR)	Base + 04H	Section 12.5.3.8, “Modem Control Register (MCR)” on page 350
Modem Status Register (MSR)	Base + 06H	Section 12.5.3.9, “Modem Status Register (MSR)” on page 351
Scratchpad Register (SCR)	Base + 07H	Section 12.5.3.10, “Scratchpad Register (SCR)” on page 352
Divisor Latch Register Low (DLL)	Base (DLAB=1)	Section 12.5.3.12, “Divisor Latch Register Low (DLL)” on page 352
Divisor Latch Register High (DLH)	Base +1 (DLAB=1)	Section 12.5.3.13, “Divisor Latch Register High (DLH)” on page 352
Global Control/Configuration Registers [00h — 2Fh]	00h – 2Fh	Section 12.7.3.1, “Global Control/Configuration Registers [00h — 2Fh]” on page 361
Logical Device Configuration Registers [30h — FFh]	30h – FFh	Section 12.7.3.2, “Logical Device Configuration Registers [30h — FFh]” on page 362

Index

#, defined 23
48 MHz Clock 374
66 MHz Clock (HLCLK) 374
82801E C-ICH functions 21
82801E C-ICH integrated LAN Controller
    role of 43
82801E C-ICH integrated LAN controller 43

A

PC/PCI DMA Acknowledges 368
PC/PCI DMA Request 368
A20 Gate 373
A20GATE 373
A20Gate Pass-Through Enable (A20PASSEN) 309
A20M# 372
ACPI_EN 210
AD31-0 366
ADDRESS 328
Address Increment/Decrement Select 230
Address Shifting 60
ADLIB_ACT_STS 274
ADLIB_LPC_EN 221
ADLIB_TRP_EN 275
Advanced Configuration 41
Advanced Programmable Interrupt Controller (APIC) 33
AF Alarm Flag 253
AFTERG3_EN 258
AIE Alarm Interrupt Enable 252
Alert-On-LAN 34
ALT_A20_GATE 255
ALTACC_EN Alternate Access Mode Enable 216
APIC bus 80
APIC bus arbitration 80
APIC Clock 370
APIC Data 245, 370
APIC ID 246
APIC Index 244
APIC_EN 216
APICCLK 370
APICD1-0 370
APM_STS 273
APMC_EN 271
AUDIO_ACT_STS 274
AUDIO_TRP_EN 275
Autoinitialization 60
Autoinitialize Enable 230
Automatic End of Interrupt (AEOI) 239
Automatic End of Interrupt Mode 77
Automatic Rotation Mode (Equal Priority Devices) 76

B

Backward compatible software 41
Base Address 174, 175, 210, 212, 294, 306, 316, 322
Base address of Descriptor table (BADDR) 302
Base and Current Address 227
Base and Current Count 228
Base Class Code 173, 195, 209, 294, 306, 322
Baud rate generator 338, 352
Baud Rate Generator Divisor Latches 340
Binary/BCD Countdown Select 233
BIOS_EN 271
BIOS_RLS BIOS Release 271
BIOS_STS 273
BIOSWE BIOS Write Enable 211
BIOSWR_STS 279
BLE BIOS Lock Enable 211
Block Data Byte 329
BOOT_STS 280
Break Interrupt 349
Buffered Mode (BUF) 239
Bus Command and Byte Enables 366
Bus Master Device Mapping 57
Bus Master Enable (BME) 171, 193, 292, 304
Bus Master IDE Active (ACT) 302
Bus Message Formats 81
BUS_ERR 325
Byte Enables 366
BYTE_DONE_STS 325

C

C/BE3-0# 366
Calculated/Checked Parity 367
CAP_ID Capability ID 177
CAP_LIST Capabilities List 172
CAP_PTR Capabilities Pointer 176
Cascade Mode 76
Cascaded Interrupt Controller IRQ Connection 238
Channel 0 Select 215
Channel 1 Select 215
Channel 2 Select. 215
Channel 3 Select 215
Channel 5 Select 215
Channel 6 Select 215
Channel 7 Select 215
Channel Mask Bits 232
Channel Mask Select 230
Channel Request Status 229
Channel Terminal Count Status 229
Character timeout interrupt 353
Clear Byte Pointer 231
Clear Mask Register 231
Clear to Send 351, 374

CLK14 374
CLK48 374
CLK66 374
CLS Cache Line Size 173
CNA CU Not Active 180
CNA Mask 181
CNF1_LPC_EN 221
CNF2_LPC_EN 221
COMA Decode Range 218
COMA_LPC_EN 222
COMB Decode Range 218
COMB_LPC_EN 222
Compatibility Enable 342
Compatibility Modules 32
Configuration Read 39, 366
Configuration registers 360, 361
Configuration space 39
Configuration Write 39, 366
Configure Flag (CF) 311
Connect Status Change 318
Control/Status Register block 42, 44
    elements 44
    internal control registers 42
COPR_ERR_EN Coprocessor Error Enable 215
COPROC_ERR 255
Count Register Status 235
Countdown Type Status 235
Counter 0 Select 234
Counter 1 Select 234
Counter 2 Select 234
Counter Latch Command 71, 234
Counter Mode Selection 233
Counter OUT Pin State 235
Counter Port 236
Counter Select 233
Counter Selection 234
Counter/timers 69
    commands 70
CPUPWR_FLR CPU Power Failure 257
CPUPWRGD 373
CPUSLP# 372
CPUSLP_EN 257
Crystal Input 374
CSMA/CD unit 43
CUC Command Unit Command 182
Current Connect Status 318
CUS Command Unit Status 181
CX Command Unit (CU) Executed 180
CX Mask 181
Cycle Completion 46
Cycle Frame 367

D

DAC Mode Address 366
Data 185
Data Carrier Detect 351, 374
Data frames 90
    format of 91
Data Parity Error Detected (DPD) 172, 194, 198
Data Parity Errors 44, 47
Data Ready 349
Data Scale 178
Data Select 178
Data Set Ready 351, 375
Data Terminal Ready 350, 375
DATA_MSG0 Data Message Byte 0 329
DATA_MSG1 Data Message Byte 1 329
DATA0/COUNT 328
DATA1 328
Date Alarm 253
DCB_EN DMA Collection Buffer Enable 216
Delivery Mode 249
Delivery Status 248
Delta Clear To Send 351
Delta Data Carrier Detect 351
Delta Data Set Ready 351
Descriptor tables 116
Destination 248
Destination Mode 249
Detected Parity Error (DPE) 172, 194, 198
DEV_ERR 326
DEV_STS DEVSEL# Timing Status 208
DEV_TRAP_EN 273
DEV_TRAP_STS 273
Device Configuration Registers 362
Device ID Value 206, 292, 304, 320
Device Identification Number 171, 192
Device Select 367
DEVMON_STS Device Monitor Status 272
DEVSEL# 367
DEVSEL# Timing Status (DEVT) 293, 305, 321
Direct Memory Access 46
Divisor Latch Bit (DLAB) 340
Divisor Latch Register High (DLH) 352
Divisor register access bit 346
DM Data Mode 252
DMA 58
DMA arbitration 336
DMA Channel Group Enable 229
DMA Channel Select 230
DMA controllers 32
DMA cycle termination 65
DMA Group Arbitration Priority 229
DMA Low Page 228
DMA read cycles 335
DMA transfer 66
    general flow 66
DMA Transfer Mode 230
DMA Transfer Type 230
DMA write cycles 335
Documents online 24
DPE Detected Parity Error 208
DPED Data Parity Error Detected 208
Drive 0 DMA Capable 302
Drive 0 DMA Timing Enable (DTE0) 296
Drive 0 Fast Timing Bank (TIME0) 297
Drive 0 IORDY Sample Point Enable (IE0) 297
Drive 0 Prefetch/Posting Enable (PPE0) 297
Drive 1 DMA Capable 302

Drive 1 DMA Timing Enable (DTE1) 296
Drive 1 Fast Timing Bank (TIME1) 296
Drive 1 IORDY Sample Point Enable (IE1) 296
Drive 1 Prefetch/Posting Enable (PPE1) 296
Drive 1 Timing Register Enable (SITRE) 296
DSE Daylight Savings Enable 252
DT Delivery Type 247
DTE Delayed Transaction Enable 216
Dual Address Cycle (DAC) format 40
Duplex Mode 188
DV Division Chain Select 251
Dynamic Data 178

E

Early Receive Count 186
ECC Error Reporting 34
Edge/Level Bank Select (LTIM) 237
Edge-triggered and level-triggered mode 76
EE0_CS 366
EE0_DIN 366
EE0_DOUT 366
EE0_SHCLK 366
EE1_CS 366
EE1_DIN 366
EE1_DOUT 366
EE1_SHCLK 366
EECS EEPROM Chip Select 184
EEDI EEPROM Serial Data In 184
EEDO EEPROM Serial Data Out 184
EEPROM 41, 42, 48
EEPROM Chip Select 366
EEPROM Data In 366
EEPROM Data Out 366
EEPROM Shift Clock 366
EESK EEPROM Serial Clock 184
Enable Special Mask Mode (ESMM) 241
End of Interrupt Operations 77
Enhanced Universal Serial Bus (USB) Controller 33
Enter Global Suspend Mode (EGSM) 311
EOS End of SMI 271
ER Early Receive 180
ER Mask 181
Error 302
Error reporting 158
Even Parity Select 347
External LAN Connect 41

F

F1_Disable 225
F2_Disable 225
F3_Disable 225
FAILED 325
FAST_PCB0 Fast Primary Drive 0 Base Clock 300
FAST_PCB1 Fast Primary Drive 1 Base Clock 300
FAST_SCB0 Fast Secondary Drive 0 Base Clock 299
FAST_SCB1 Fast Secondary Drive 1 Base Clock 299
FC Full 186
FC Paused 186
FC Paused Low 186
FCP Flow control Pause 180
FCP Mask 181
FDD Decode Range 218
FDD_LPC_EN 221
FERR# 372
FIFO 42
    DMA requests 354
    operation 353
    polled mode 354
FIFO Control Register (FCR) 345
FIFO Error Status 348
FIFO Mode Enable Status 343
Filtering 50
Firmware Hub (FWH) interface. 156
Firmware Hub Signals 366
Fixed ranges 162, 163
Flow control 50
Flow Control Threshold 187
Force Global Resume (FGR) 311
FORCE_THTL 266
FR Frame Received 180
FR Mask 181
Frame List Current Index/Frame Number 315
FRAME# 367
Framing Error 349
Full duplex mode 50
FULL_RST 255
Fully Nested Mode 75
Function Disable 34
FWH_C0_EN 220, 224
FWH_C0_IDSEL 222
FWH_C8_EN 220, 224
FWH_C8_IDSEL 222
FWH_D0_EN 220, 224
FWH_D0_IDSEL 222
FWH_D8_EN 220, 224
FWH_D8_IDSEL 222, 223
FWH_E0_EN 220
FWH_E0_IDSEL 222, 223
FWH_E8_EN 220
FWH_E8_IDSEL 222, 223
FWH_F0_EN 219
FWH_F0_IDSEL 222, 223
FWH_F8_EN 219
FWH_F8_IDSEL 222
FWH3-0 366, 370
FWH4 366, 370

G

GAMEH_LPC_EN 221
GAMEL_LPC_EN 221
GBL _STS 263
GBL_EN 264
GBL_RLS Global Release 265
GBL_SMI_EN 271
GEN1_BASE Generic I/O Decode Range 1 Base 220
GEN1_EN Generic Decode Range 1 Enable 220

GEN2_BASE Generic I/O Decode Range 2 Base 223
GEN2_EN Generic I/O Decode Range 2 Enable 223
General purpose inputs and outputs 33
Global configuration registers 361
Global Reset (GRESET) 312
GNT3-0# 368
GNT5# 368
GNTA# 368
GNTB# 368
GPE0_STS 272
GPE1_STS 272
GPI Route 258
GPI_EN 270
GPI_STS 269
GPIO_EN 212
GPIO_SEL 286
GPIO_USE_SEL 285
GPIO0 368
GPIO1 368
GPIO11 373
GPIO16 368
GPIO17 368
GPIO17# 368
GPIO31-0 377
GPIO4 370
GPIO5 370

H

Handshake responses 139
Header Type 174, 209
HIDE_ISA Hide ISA Bridge 215
HL_STB 365
HL_STB# 365
HL11 376
HL11-0 365
HLCLK 374
HLCOMP 365
Hole Enable (15MB-16MB). 202
Host controller 145
Host Controller Halted 314
Host Controller Process Error 314
Host Controller Reset (HCRESET) 312
Host processor interface 334
Host System Error 314
HOST_BUSY 326
HOURFORM Hour Format 252
HP_PCI_EN 202
HST_EN SMBus Host Enable 323
Hub Architecture 31
Hub interface 35
Hub Interface Compensation 365
Hub Interface Signals 365
Hub Interface Strobe 365
Hub Interface Strobe Complement 365
HUBNMI_STS 279
HUBREF 378
HUBSCI_STS 279
HUBSERR_STS 279
HUBSMI_STS 279

I

I/O Address Base bits 197
I/O Address Limit bits 197
I/O Addressing Capability 197
I/O APIC 79
    differences from the 8259 79
I/O APIC Identification 247
I/O Read 366
I/O Space Enable (IOE) 171, 193
I/O Space Enable (IOSE) 304, 320
I/O Write 366
I2C_EN 323
ICW/OCW select 237
ICW4 Write Required (IC4) 237
IDE controller 113
IDE Decode Enable (IDE) 296
IDE Interface 32
IDEP0_ACT_STS 275
IDEP0_TRP_EN 276
IDEP1_ACT_STS 275
IDEP1_TRP_EN 276
IDES0_ACT_STS 275
IDES0_TRP_EN 276
IDES1_ACT_STS 274
IDES1_TRP_EN 275
IGNNE# 372
Ignore Numeric Error 372
INIT# 372
INIT_NOW 255
Initialization 372
Initialization command words 74
Initiator Ready 367
Input clock to the SIU 374
Instructions, notational conventions 23
INT_LN Interrupt Line 176
INT_PN Interrupt Pin 176
Intel's Adaptive Technology 41, 42
Interesting Packet 187
Internal LAN Master Request Status (LAN_MREQ_STS) 203
Internal PCI Master Request Status (INT_MREQ_STS) 203
Interrupt 302
    not sharable 91
Interrupt Acknowledge 366
Interrupt acknowledge cycle 73
Interrupt Enable 185
Interrupt Enable Register (IER) 342
Interrupt Identification Register (IIR) 342, 343
    Decode 344
Interrupt Input Pin Polarity 248
Interrupt Level Select (L2, L1, L0) 240
Interrupt Line 307, 323
Interrupt Mask 181
Interrupt On Complete (IOC) Enable 315
Interrupt Pending 343
Interrupt pin 307, 323
Interrupt Priority Level 343
Interrupt Request 14-15 370
Interrupt Request Level 238
Interrupt Request Mask 239

Interrupt Source Encoded 343
Interrupt Trigger Level 345
Interrupt Vector Base Address 238
INTR 326, 372
INTRD_DET Intruder Detect 280
INTRD_SEL 282
INTREN 327
Intruder Detect 34, 373
INTRUDER# 373
INUSE_STS 325
IO Space Indicator 322
IOCHK_NMI_EN 254
IOCHK_NMI_STS IOCHK# NMI Source Status 254
IORDY Sample Point (ISP) 296
IOSE I/O Space Enable (IOSE) 292
IRDY# 367
IRQ Number 245
IRQ Pin Assertion Register 87
IRQ Routing 212, 213
IRQ10 ECL 243
IRQ11 ECL 243
IRQ12 ECL 243
IRQ12LEN Mouse IRQ12 Latch Enable 216
IRQ14 ECL 243
IRQ14-15 370
IRQ15 ECL 243
IRQ1LEN Keyboard IRQ1 Latch Enable 215
IRQ3 ECL 242
IRQ4 ECL 242
IRQ5 ECL 242
IRQ6 ECL 242
IRQ7 ECL 242
IRQ9 ECL 243
IRQEN Interrupt Routing Enable 212, 213
IRQF Interrupt Request Flag 253
ISA Enable 201
ISA-Compatible interrupt controller 32

K

KBC_ACT_STS 274
KBC_LPC_EN 221
KBC_TRP_EN 275
Keyboard Controller Reset Processor 373
KILL 327

L

L128LOCK Lower 128-byte Lock 217
LAD3-0 366, 370
LAN Connect Address 185
LAN Connect implementation 40
LAN Connect Register Address 185
LAN Connect Software Reset 188
LAN Controller 33, 40, 42, 169
    acting as master and a slave 169
    Architectural Overview 41
    Carrier-Sense Multiple Access with Collision Detect (CSMA/CD) 41
LAN Controller Architectural Overview
    FIFO subsystem 41
    Parallel subsystem 41
LAN Interface Clock 365
LAN Reset/Sync 365
LAN0_CLK 365
LAN0_RSTSYNC 365
LAN0_RXD2-0 365
LAN0_TXD2-0 365
LAN1_CLK 365
LAN1_RSTSYNC 365
LAN1_RXD2-0 365
LAN1_TXD2-0 365
Latch Count of Selected Counters 234
Latch Status of Selected Counters 234
LDRQ1-0# 370
LEG_ACT_STS 274
LEG_IO_TRP_EN 275
LEGACY_USB_EN 271
LEGACY_USB_STS 273
LFRAME# 366, 370
Line Control Register (LCR) 346
Line Status 317
Line Status Register 341
Link status change 41
Link Status Change Indication 187
Link Status Indication 188
Locked cycles 36
Logical Device Number 337
Loop Back Test Mode 311, 350
Low Pin Count (LPC) Interface 32
Low Speed Device Attached (LS) 317
LPC Bridge function 52, 205
LPC bus 334
    and DMA 335
    I/O transfers 336
    interface 52, 335
LPC Frame 370
LPC Interface Decoders 57
LPC Multiplexed Command, Address, Data 370
LPC Serial DMA/Master Request Inputs 370
LPT Decode Range 218
LPT_LPC_EN 222

M

Magic Packet 187
Management Data Interface (MDI) Control 42
Management interface 51
Mask 248
Mask A20 372
Master Abort Mode 201
Master Clear 231
Master Latency Count 197
Master Latency Timer (MLT) 197
Master/Slave in Buffered Mode 239
Max Packet (MAXP) 311
Maximum Redirection Entries 247
MC_LPC_EN 221
MCSMI_EN Microcontroller SMI Enable 271

MCSMI_STS Microcontroller SMI# Status 272
MDI Management Data Interrupt 180
Measurements, defined 23
Memory Address Base 199
Memory Address Limit 199
Memory maps 161
    register-set level 161
Memory ranges 166
Memory Read 366
Memory Read Line 366
Memory Read Line (MR Line) command 45
Memory Read Multiple 366
Memory Read Multiple (MR Multiple) command 45
Memory Space Enable (MSE) 171, 193
Memory Write 366
Memory Write (MW) command 45
Memory Write and Invalidate 366
Memory Write and Invalidate (MWI) command 45
Microprocessor Mode 239
MIDI Decode Range 219
MIDI_ACT_STS 274
MIDI_LPC_EN 221
MIDI_TRP_EN 275
MLTC Master Latency Timer Count 174
Mode Selection Status 235
Modem Interrupt Enable 342
MON_TRAP_BASE 260
MON4_FWD_EN 259
MON4_MASK 260
MON5_FWD_EN 259
MON5_MASK 260
MON6_FWD_EN 259
MON6_MASK 260
MON7_FWD_EN 259
MON7_MASK 260
MSS Decode Range 219
MSS_LPC_EN 221
Multi-function Device 174, 209
Multi-Transaction Timer Count Value 203
MWIE Memory Write and Invalidate Enable 171

N

NEWCENTURY_STS 279
NMI 372
NMI_EN 254
NMI_NOW 281
NMI2SMI_EN 279, 281
NO_REBOOT 217
Non-Maskable Interrupt 372
Normal End of Interrupt 77
Numeric Coprocessor Error 372
NXT_PTR Next Item Pointer 177

O

OC1-0# 371
OCW1 Interrupt Mask 239
OCW2 Select 240
OCW3 Select 241
On-chip functional units 21
Online help 24
Opcode 185
Oscillator Clock 374
OUT2# Signal Control 350
Overcurrent Active 317
Overcurrent Indicator 317
Overcurrent Indicators 371
Overrun Error 349

P

Packet field formats 135
PAR 367
Parallel subsystem 42
Parity Enable 347
Parity Error 349, 367
Parity Error Response 193
Parity Error Response Enable 201
Parity errors 38
Pass Through State (PSTATE) 309
PCB0 300
PCB1 300
PCI Address/Data 366
PCI Bus Master 45
PCI bus master interface 42
PCI Clock 368
PCI configuration write cycles 35
    characteristics of 35
PCI expansion 62
PCI Grants 368
PCI Interface 31
PCI Interrupt Enable (USBPIRQEN) 308
PCI Interrupt Requests 370
PCI Lock 368
PCI Master Request Status (PCI_MREQ_STS) 203
PCI Power Management 41
PCI Requests 368
PCI Reset 368
PCI_DAC_EN 202
PCI_SERR_EN 254
PCICLK 368
PCIRST# 368
PCIRST# signal 48
PCI-to-PCI bridge 36
PDA2-0 369
PDCS1# 369
PDCS3# 369
PDD15-0 369
PDDACK# 369
PDDREQ 369
PDIOR# 369
PDIOW# 369
PDRSTB 369
PDSTOP 369
PDWSTB 369
PER Parity Error Response 171, 207
PER_SMI_SEL 257
Performing an abort 56

PERIODIC_EN 270
PERIODIC_STS 272
PERR# 367
PF Periodic Interrupt Flag 253
PIE Periodic Interrupt Enable 252
PIORDY 369
PIRQA-D# 370
PIRQAE_ACT_STS 274
PIRQBF_ACT_STS 274
PIRQCG_ACT_STS 274
PIRQDH_ACT_STS 274
PIRQE-H# 370
PLOCK# 368
PM1_STS_REG 272
PME Enable 178
PME_EN 268
PME_STS 267
Pointer Field 183
Poll Mode 76
Poll Mode Command 241
Polled mode operation 354
PORT 42
Port Enable/Disable Change 318
Port Enabled/Disabled (PORT_EN) 318
PORT Function Selection 183
Port Reset 317
PORT0EN 309
PORT1EN 309
POS_DEC_EN Positive Decode Enable 216
Power management registers 256
Power OK 371
PRBTNOR_STS Power Button Override Status 263
PRDMARDY# 369
Prefetch capabilities 36
Prefetchable Memory Address Base 200
Prefetchable Memory Address Limit 200
PRIM_SIG_MODE 299
Primary Bus Number 196
Primary Disk DMA Ready 369
Primary Disk I/O Read (PIO and Non-Ultra DMA) 369
Primary Disk I/O Write 369
Primary Disk Read Strobe 369
Primary Disk Stop 369
Primary Disk Write Strobe 369
Primary Drive 0 Cycle Time (PCT0) 299
Primary Drive 0 Synchronous DMA Mode Enable (PSDE0) 298
Primary Drive 1 Cycle Time (PCT1) 299
Primary Drive 1 IORDY Sample Point (PISP1) 297
Primary Drive 1 Recovery Time (PRCT1) 297
Primary Drive 1 Synchronous DMA Mode Enable (PSDE1) 298
Primary I/O Channel Ready 369
Primary IDE Device Address 369
Primary IDE Device Chip Selects 369
Primary IDE Device Data 369
Primary IDE Device DMA Acknowledge 369
Primary IDE Device DMA Request 369
Primary Master Channel Cable Reporting 300
Primary Slave Channel Cable Reporting 300
Priority resolution 58
Process call 147
Processor Interrupt 372
Processor Power Good 373
Processor Present Indicator 34
Processor Sleep 372
Product literature, ordering 24
Programming Interface Value 293, 321
PRQ 247
PWDMARDY# 369
PWR_FLR Power Failure 258
PWRBTN__STS 263
PWRBTN_EN 264
PWRBTN_LVL 257
PWRGOOD input 97
PWROK 371
PWROK_FLR PWROK Failure 257

Q

Queue heads 127

R

RCIN 373
RCIN# 373
Read Accesses 44, 45
Read Align 47
Read Back Command 71, 234
Read/Write Control (RWC) 301
Read/Write Select 233
Read/Write Selection Status 235
Ready 185
Real Time Clock 33, 92
Receive DMA Byte Count 185
Receive FIFO 353
Received Data 365
Received Master Abort (RMA) 194, 198
Received Master-Abort Status (RMA) 293, 305
Received System Error (SSE) 194, 198
Received Target Abort (RTA) 172, 194, 198
Receiver Data Available Interrupt Enable 342
Receiver interrupts 353
Receiver Line Status Interrupt Enable 342
Receiver Time Out Interrupt Enable 342
Recovery Time (RCT) 296
Redirection Entry Clear 246
Redirection table 248
REF_TOGGLE Refresh Cycle Toggle 254
Register address map
  Serial I/O Unit 334
Register Read Command 241
Registers 21
  Global configuration 361
  Hub Interface to PCI 191
  IDE Controller 291
  LAN Controller 169
  Logical device configuration 362
  LPC Interface 205

Memory map 161
SMBus Controller 319
UART 340
USB Controller 303
Remote IRR 248
Remote read message 85
REQ3-0# 368
REQ5# 368
REQA# 368
REQB# 368
Request to Send 350, 375
RESERVED1 376
RESERVED2 376
Reset
and Serial I/O unit 336
Reset Receiver FIFO 345
Reset states
SIU 340
Reset Transmitter FIFO 345
Resource Indicator 210, 212
Resource Type Indicator (RTE) 294, 306
Resume Detect (RSM_DET) 314, 317
Resume Interrupt Enable 315
Resume Well Power OK 371
Resume Well Reset 371
Retry 44
Revision Identification Number 173, 195, 209, 293, 321
RI# 376
RI_EN 268
RI_STS 267
Ring Indicate 376
Ring Indicator 351, 375
RMA Master Abort Status 172, 208
RNR Mask 181
RNR Receive Not Ready 180
Rotate and EOI Codes (R, SL, EOI) 240
Rotating Priority 59
RS Rate Select 251
RSM_PWROK 371
RSMRST# 371
RST_CPU 255
RTA Received Target Abort 208
RTC Reset 376
RTC_EN RTC Event Enable 264
RTC_INDX Real Time Clock Index Address 254
RTC_PWR_STS 258
RTC_STS 263
RTCRST# 376
RTCX1 374
RTCX2 374
RUC Receive Unit Command 182
Run/Stop (RS) 312
RUS Receive Unit Status 181
RW 328

S

SAFE_MODE 217
SB16 Decode Range 219
SB16_LPC_EN 221
SCB General Pointer 183
SCB1 300
SCBO 300
SCI_EN 265
SCI_IRQ_SEL 210
SDA2-0 369
SDCS1# 369
SDCS3# 369
SDD15-0 369
SDDACK# 369
SDDREQ 369
SDIOR# 369
SDIOW# 369
SDRSTB 369
SDSTOP 369
SDWSTB 369
SEC_SIG_MODE 299
SECOND_TO_STS 280
Secondary Bus Number 196
Secondary Disk DMA Ready 369
Secondary Disk I/O Read (PIO and Non-Ultra DMA) 369
Secondary Disk I/O Write 369
Secondary Disk Read Strobe 369
Secondary Disk Stop 369
Secondary Disk Write Strobe 369
Secondary Drive 0 Cycle Time (SCT0) 298
Secondary Drive 0 Synchronous DMA Mode Enable (SSDE0) 298
Secondary Drive 1 Cycle Time (SCT1) 298
Secondary Drive 1 IORDY Sample Point (SISP1) 297
Secondary Drive 1 Recovery Time (SRCT1) 297
Secondary Drive 1 Synchronous DMA Mode Enable (SSDE1) 298
Secondary I/O Channel Ready 369
Secondary IDE Device Address 369
Secondary IDE Device Chip Selects 369
Secondary IDE Device Data 369
Secondary IDE Device DMA Acknowledge 369
Secondary IDE Device DMA Request 369
Secondary Master Channel Cable Reporting 300
Secondary Slave Channel Cable Reporting 300
SENDNOW 281
Serial Bus Release Number 308
Serial I/O Unit 34
and DMA arbitration 336
and Reset 336
configuration 358
DMA transfers 336
register address map 334
Reset states 340
serial encoding sequence 336
Serial Input 375
Serial Interrupt Request 370
Serial interrupts 355
Serial Line Control Register 340
Serial Output 375
SERIRQ 370
SERIRQ_SMI_STS 272

SERR# 36, 368
SERR# Due to Delayed Transaction Timeout (SERR_DTT). 204
SERR# Due to Received Target Abort (SERR_RTA). 204
SERR# Enable 201
SERR# Enable (SERR_EN) 171, 193
SERR# enable on Delayed Transaction Timeout (SERR_DTT_EN) 204
SERR# enable on receiving target abort (SERR_RTA_EN) 204
SERR#_NMI_STS SERR# NMI Source Status 254
SERR_DTT SERR# Due to Delayed Transaction Timeout 214
SERR_DTT_EN SERR# on Delayed Transaction Timeout Enable 214
SERR_EN 207
SERR_RTA SERR# Due to Received Target Abort 214
SERR_RTA_EN SERR# on Received Target Abort Enable 214
Set break 346
SET Update Cycle Inhibit 252
SFPW Start Frame Pulse Width 213
Short messages 82
Short Packet Interrupt Enable 315
SI Software Generated Interrupt 181
Signal Description 365
  EEPROM Interface 366
  Firmware Hub Interface 366
  GPIO 377
  Hub Interface 365
  IDE Interface 369
  Interrupt Interface 370
  LAN Connect 365
  LPC Interface 370, 376
  Miscellaneous 376
  Other Clocks 374
  PCI Interface 366
  Power and Ground 378
  Power Signals 371
  Processor Interface 372
  Real Time Clock Interface 374
  SMBus Interface 373
  System Management Interface 373
  UARTs 374
  USB Interface 371
Signaled System Error (SSE) 172
Signaled Target Abort (STA) 194
Signaled Target-Abort Status (STA) 293, 305, 321
signals 21
Signals, notational conventions 23
Single or Cascade (SNGL) 237
SIORDY 369
SIRQEN Serial IRQ Enable 213
SIRQMD Serial IRQ Mode Select 213
SIRQSZ Serial IRQ Frame Size 213
SIU LPC clock input to SIU 376
SIU LPC Frame 376
SIU LPC Multiplexed Command, Address, Data 376
SIU LPC Serial DMA/Master Request Output 376
SIU RESET 376
SIU Serial IRQ Input 376
SIU_LAD3-0 376
SIU_LCLK 376
SIU_LDRQ# 335, 376
SIU_LFRAME# 376
SIU_RESET# 336, 376
SIU_SERIRQ 376
  Continuous (Idle) Mode 356
  Data frame 356
  Initialization 357
  latency 357
  Quiet (Active) Mode 355
  Reset 357
  sampling periods 356
  stop cycle 357
SIU0_CTS# 374
SIU0_DCD# 374
SIU0_DSR# 375
SIU0_DTR# 375
SIU0_RI# 375
SIU0_RTS# 375
SIU0_RXD 375
SIU0_TXD 375
SIU1_CTS# 374
SIU1_DCD# 374
SIU1_DSR# 375
SIU1_DTR# 375
SIU1_RI# 375
SIU1_RTS# 375
SIU1_RXD 375
SIU1_TXD 375
Slave Identification Code 239
Slave write cycle format 153
SLAVE_ADDR 329
SLP_EN 265
SLP_SMI_EN 271
SLP_SMI_STS 273
SLP_TYP 265
SMB_CMD 327
SMB_FOR_BIOS 225
SMB_SMI_EN 323
SMB_WAK_STS SMBus Wake Status 267
SMBALERT# 373
SMBALERT_STS 325
SMBCLK 373
SMBDATA 373
SMBus 34
SMBus Alert 373
SMBus Clock 373
SMBus Data 373
SMBus Host 34
SMBUS_SMI_STS 272
SMI at End of Pass-through Enable (SMIATENDPS) 309
SMI Caused by End of Pass-through (SMIBYENDPS) 308
SMI Caused by Port 60 Read (TRAPBY60R) 309
SMI Caused by Port 60 Write (TRAPBY60W) 308
SMI Caused by Port 64 Read (TRAPBY64R) 308
SMI Caused by Port 64 Write (TRAPBY64W) 308
SMI Caused by USB Interrupt (SMIBYUSB) 308
SMI on Port 60 Reads Enable (60REN) 309

SMI on Port 60 Writes Enable (60WEN) 309
SMI on Port 64 Reads Enable (64REN) 309
SMI on Port 64 Writes Enable (64WEN) 309
SMI on USB IRQ (USBSMIEN) 309
SMI# 372
SMLINK1-0 373
SOF Timing Value 316
Software commands 61
Software Debug (SWDBG) 311
Speaker 376
Special Cycle 366
Special Fully Nested Mode (SFNM) 75, 239
Special Mask Mode (SMM) 241
Specific Rotation Mode (Specific Priority) 76
Specifications, links to 25
Speed 188
Speed Strap 372
SPKR 376
SPKR_DAT_EN 254
SQWE Square Wave Enable 252
SRDMARDY# 369
SSE Signaled System Error 208
STA Signaled Target Abort 208
START 326
START field 57
Start frame 90
    IRQ protocol and its effects 90
Start/Stop Bus Master (START) 301
Statistical counters 190
Sticky Parity 346
Stop 367
Stop Bits 347
Stop Clock Request 372
STOP# 367
STPCLK# 372
Sub Class Code 173, 195, 209, 294, 306, 322
Subordinate Bus Number 196
SUSCLK 376
Suspend 317
Suspend Clock 376
SW_TCO_SMI 279
SWDMARDY# 369
SWI Software Interrupt 180
SWSMI_RATE_SEL 257
SWSMI_TMR_EN Software SMI# Timer Enable 271
SWSMI_TMR_STS 273
SYNC field 68, 134
SYNC protocol 55
SYS_RST 255
System Control Block (SCB) 42
System Error 45, 368
System Management Interrupt 372
System Management Link 373
System Power States 98

T

Target Ready 367
Target-Disconnect 44
TCO Timer 34
TCO_EN 270
TCO_INT_EN TCO Interrupt Enable 211
TCO_INT_SEL TCO Interrupt Select 211
TCO_INT_STS 279
TCO_MESSAGE 282
TCO_STS 272
TCO_TMR_HLT TCO Timer Halt 281
TCOSCI_EN 268
TCOSCI_STS 267
Technical support 24
Terminal count 67
Test Bit 350
Test Point 376, 377
Thermal Alarm 376
THRM# 376
THRM#_POL 268
THRM_DTY 266
THRM_EN 268
THRM_STS Thermal Interrupt Status 267
THRMOR_STS Thermal Interrupt Override Status 267
Throttling 102
THT_EN 266
THTL_DTY 266
THTL_STS Throttle Status 266
TIM_CNT2_EN Timer Counter 2 Enable 254
Time Out Detected 343
TIMEOUT 279
Timeout/CRC Interrupt Enable 315
TMR_VAL 265
TMR2_OUT_STS Timer Counter 2 OUT Status 254
TMROF_EN Timer Overflow Interrupt Enable 264
TMROF_STS Timer Overflow Status 263
Token packets 137
Top-Block Swap mode 167
Total Cost of Ownership 34, 51
TP0 376
TP1 377
TP2 377
TP3 377
Trailing Edge Ring Indicator 351
Transfer descriptors 123
Transfer queues 130
Transmit and Receive FIFO Enable 345
Transmit Data 365
Transmit Data Request 341, 348
Transmit Data Request Interrupt Enable 342
Transmit Holding Register 341
Transmit interrupt 353
Transmit Shift Register 341
Transmitter Empty 348
Traversal options 132
    breadth first 132
    depth-first 132
TRDY# 367
Trigger Mode 248
Type 0 configuration cycle 39
Type 1 configuration cycle 39

U

U128E Upper 128-byte Enable 217
U128LOCK Upper 128-byte Lock 217
UART
    Baud rates 338
    Data frame 339
    FIFO support 337
    Functional description 337
    functions supported 337
    operation 339
    registers 340
UART_CLK 374
UART0,1 332
UF Update-ended Flag 253
UIE Update-ended Interrupt Enable 252
UIP Update In Progress 251
Ultra ATA/33 protocol 119
Ultra ATA/66 protocol 121
Units of measure, defined 23
Universal Serial Bus Port 1
    0 Differential 371
Update cycle 92
URL 24
USB Error Interrupt 314
USB host controller 122
USB Interrupt (USBINT) 314
USB keyboard 142
USB1_EN 268
USB1_STS 267
USBP0N 371
USBP0P 371
USBP1N 371
USBP1P 371

V

V_CPU_IO 378
V5REF 378
Variable ranges 162, 165
VBIAS 378
Vcc1_8 378
Vcc3_3 378
VccRTC 378
Vendor ID Value 206, 292, 303, 319
Vendor Identification Number 192
Verify mode 67
VGA Enable 201
VLAN Support 51
VRM Power Good 371
VRMPWRGD 371
VRT Valid RAM and Time Bit 253
Vss 378

W

WAK_STS 263
WDSTATUS Watchdog Status 282
Wired for Management (WfM) Rev 2.0 41
Word Length Select 347
World Wide Web 24
WR_PingPong_EN 300
Write Accesses 44, 45
Write byte/word access 146
www.intel.com 24

X

Xoff 186
Xon 187