



Intel[®] 7500 Chipset

Datasheet

March 2010



THIS SPECIFICATION IS PROVIDED "AS IS" WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. Intel disclaims all liability, including liability for infringement of any proprietary rights, relating to use of information in this specification. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted herein.

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, life sustaining, critical control or safety systems, or in nuclear facility 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® 7500 chipset and Intel® Itanium® processor 9300 series, ICH9/ICH10, Intel® Xeon® processor 7500 series-based platform and Intel Itanium processor 9300 series-based platforms may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

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

Intel® Virtualization Technology requires a computer system with an enabled Intel® processor, BIOS, virtual machine monitor (VMM) and, for some uses, certain computer system software enabled for it. Functionality, performance or other benefits will vary depending on hardware and software configurations and may require a BIOS update. Software applications may not be compatible with all operating systems. Please check with your application vendor.

No computer system can provide absolute security under all conditions. Intel® Trusted Execution Technology (Intel® TXT) requires a computer system with Intel® Virtualization Technology, an Intel TXT-enabled processor, chipset, BIOS, Authenticated Code Modules and an Intel TXT-compatible measured launched environment (MLE). The MLE could consist of a virtual machine monitor, an OS or an application. In addition, Intel TXT requires the system to contain a TPM v1.2, as defined by the Trusted Computing Group and specific software for some uses. For more information, see <http://www.intel.com/technology/security/>.

Copies of documents which have an order 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 Web Site.

Intel, Xeon, Itanium, and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

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

Copyright © 2010, Intel Corporation.



Contents

| | | |
|----------|---|----|
| 1 | Introduction | 19 |
| 1.1 | System Configuration Overview | 19 |
| 1.2 | Feature Summary | 19 |
| 1.2.1 | Features By Segment based on PCI Express Ports | 20 |
| 1.2.2 | Non-Legacy IOH | 21 |
| 1.2.3 | Intel® QuickPath Interconnect Features | 21 |
| 1.2.4 | PCI Express* Features | 21 |
| 1.2.5 | Enterprise South Bridge Interface (ESI) Features | 21 |
| 1.2.6 | Controller Link (CL) | 22 |
| 1.2.7 | Intel® I/OAT Gen3 | 22 |
| 1.2.8 | Intel® Virtualization Technology for Directed I/O (Intel® VT-d), Second Revision | 22 |
| 1.2.9 | Reliability, Availability, Serviceability (RAS) Features | 22 |
| 1.2.10 | Power Management Support | 23 |
| 1.2.11 | Security | 23 |
| 1.2.12 | Other | 23 |
| 1.3 | Terminology | 23 |
| 1.4 | Related Documents | 26 |
| 2 | Platform Topology | 27 |
| 2.1 | Introduction | 27 |
| 2.2 | IOH Supported Topologies | 27 |
| 2.2.1 | Platform Topologies | 27 |
| 2.3 | I/O Sub-System | 30 |
| 3 | Interfaces | 33 |
| 3.1 | Introduction | 33 |
| 3.2 | Intel® QuickPath Interconnect | 33 |
| 3.2.1 | Physical Layer | 33 |
| 3.2.2 | Link Layer | 35 |
| 3.2.3 | Routing Layer | 35 |
| 3.2.4 | Protocol Layer | 35 |
| 3.3 | PCI Express* Interface | 38 |
| 3.3.1 | Gen1/Gen2 Support | 38 |
| 3.3.2 | PCI Express* Link Characteristics - Link Training, Bifurcation, and Lane Reversal Support | 38 |
| 3.3.3 | Degraded Mode | 39 |
| 3.3.4 | Lane Reversal | 40 |
| 3.3.5 | Form-Factor Support | 40 |
| 3.3.6 | IOH Performance Policies | 40 |
| 3.3.7 | Address Translation Caching (ATC) | 42 |
| 3.3.8 | PCI Express* RAS | 42 |
| 3.3.9 | Power Management | 43 |
| 3.4 | Enterprise South Bridge Interface (ESI) | 43 |
| 3.4.1 | Interface Speed and Bandwidth | 43 |
| 3.4.2 | Supported Widths | 43 |
| 3.4.3 | Performance Policies on ESI | 44 |
| 3.4.4 | Error Handling | 44 |
| 3.5 | Reduced Media Independent Interface (RMII) | 44 |
| 3.6 | Control Link (CLink) Interface | 44 |
| 3.7 | System Management Bus (SMBus) | 44 |
| 3.7.1 | SMBus Physical Layer | 45 |
| 3.7.2 | SMBus Supported Transactions | 45 |
| 3.7.3 | Addressing | 46 |



| | | |
|----------|---|-----------|
| 3.7.4 | SMBus Initiated Southbound Configuration Cycles | 47 |
| 3.7.5 | SMBus Error Handling..... | 48 |
| 3.7.6 | SMBus Interface Reset | 48 |
| 3.7.7 | Configuration and Memory Read Protocol | 49 |
| 3.8 | JTAG Test Access Port Interface..... | 54 |
| 3.8.1 | JTAG Configuration Register Access..... | 54 |
| 3.8.2 | JTAG Initiated Southbound Configuration Cycles | 56 |
| 3.8.3 | Error Conditions | 57 |
| 4 | Intel® QuickPath Interconnect..... | 59 |
| 4.1 | Introduction..... | 59 |
| 4.2 | Physical Layer | 59 |
| 4.2.1 | Supported Frequencies | 60 |
| 4.2.2 | Supported Widths | 60 |
| 4.2.3 | Initialization / Re-initialization | 60 |
| 4.3 | Link Layer..... | 60 |
| 4.3.1 | Link Layer Initialization..... | 61 |
| 4.3.2 | Initialization..... | 62 |
| 4.3.3 | Packet Framing | 62 |
| 4.3.4 | Sending Credit Counter | 62 |
| 4.3.5 | Retry Queue Depth | 63 |
| 4.3.6 | Receiving Queue..... | 63 |
| 4.3.7 | Link Error Protection | 63 |
| 4.3.8 | Message Class..... | 64 |
| 4.3.9 | Link Level Credit Return Policy..... | 64 |
| 4.3.10 | Ordering Requirements..... | 64 |
| 4.4 | Routing Layer..... | 65 |
| 4.4.1 | Outbound Routing..... | 65 |
| 4.5 | Protocol Layer | 65 |
| 4.5.1 | NodeID Assignment | 66 |
| 4.5.2 | Source Address Decoder (SAD)..... | 67 |
| 4.5.3 | Special Response Status | 70 |
| 4.5.4 | Abort Time-out | 70 |
| 4.5.5 | Illegal Completion/Response/Request..... | 70 |
| 4.5.6 | Inbound Coherent Transactions | 71 |
| 4.5.7 | Inbound Non-Coherent Transactions..... | 72 |
| 4.5.8 | Outbound Snoops | 73 |
| 4.5.9 | Outbound Non-Coherent | 74 |
| 4.6 | Profile Support | 76 |
| 4.7 | Lock Arbiter | 77 |
| 4.7.1 | Lock Arbiter Time-Out | 80 |
| 4.8 | Write Cache | 80 |
| 4.8.1 | Write Cache Depth..... | 80 |
| 4.8.2 | Coherent Write Flow..... | 80 |
| 4.8.3 | Cache State | 81 |
| 4.8.4 | System Directory Support..... | 81 |
| 4.9 | Outgoing Request Buffer (ORB) | 81 |
| 4.9.1 | ORB Depth..... | 81 |
| 4.9.2 | Requestor Transaction ID (RTID) | 82 |
| 4.9.3 | Time-Out Counter | 82 |
| 4.10 | Conflict Handling | 83 |
| 4.10.1 | Coherent Local-Local Conflicts | 83 |
| 4.10.2 | Coherent Remote-Local Conflicts..... | 85 |
| 4.10.3 | Resource Conflicts | 86 |
| 4.11 | Deadlock Avoidance | 86 |
| 4.11.1 | Protocol Channel Dependence..... | 86 |



| | | |
|----------|--|-----------|
| 5 | PCI Express* and ESI Interfaces | 87 |
| 5.1 | Introduction | 87 |
| 5.2 | PCI Express* Link Characteristics - Link Training, Bifurcation, Downgrading and Lane Reversal Support | 87 |
| 5.2.1 | Link Training | 87 |
| 5.2.2 | Port Bifurcation | 87 |
| 5.2.3 | Degraded Mode | 88 |
| 5.2.4 | Lane Reversal | 89 |
| 5.2.5 | PCI Express* Gen1/Gen2 Speed Selection | 89 |
| 5.2.6 | Form-Factor Support | 89 |
| 5.3 | IOH Performance Policies..... | 89 |
| 5.3.1 | Max_Payload_size | 89 |
| 5.3.2 | Isochronous Support and Virtual Channels | 90 |
| 5.3.3 | Non-Coherent Transaction Support | 90 |
| 5.3.4 | Completion Policy | 90 |
| 5.3.5 | Read Prefetching Policies | 90 |
| 5.3.6 | Error Reporting | 91 |
| 5.3.7 | Intel Chipset-Specific Vendor-Defined Messages | 91 |
| 5.4 | Inbound Transactions | 91 |
| 5.4.1 | Inbound Memory, I/O and Configuration Transactions Supported | 91 |
| 5.4.2 | Configuration Retry Completions | 92 |
| 5.4.3 | Inbound PCI Express Messages Supported | 93 |
| 5.5 | Outbound Transactions | 93 |
| 5.5.1 | Memory, I/O and Configuration Transactions Supported | 93 |
| 5.5.2 | Lock Support | 94 |
| 5.5.3 | Outbound Messages Supported | 94 |
| 5.6 | 32-/64-Bit Addressing | 95 |
| 5.7 | Transaction Descriptor | 95 |
| 5.7.1 | Transaction ID | 96 |
| 5.7.2 | Attributes..... | 97 |
| 5.7.3 | Traffic Class..... | 97 |
| 5.8 | Completer ID..... | 97 |
| 5.9 | Miscellaneous Information | 98 |
| 5.9.1 | Number of Outbound Non-Posted Requests | 98 |
| 5.9.2 | MSIs Generated from Root Ports and Locks..... | 98 |
| 5.9.3 | Completions for Locked Read Requests..... | 98 |
| 5.10 | PCI Express RAS | 98 |
| 5.10.1 | ECRC Support | 98 |
| 5.10.2 | Completion Time-Out | 98 |
| 5.10.3 | Data Poisoning..... | 99 |
| 5.10.4 | Role-Based Error Reporting..... | 99 |
| 5.11 | Link Layer Specifics | 99 |
| 5.11.1 | Ack/Nak | 99 |
| 5.11.2 | Link Level Retry | 100 |
| 5.11.3 | Ack Time-Out | 100 |
| 5.11.4 | Flow Control | 100 |
| 5.12 | Power Management | 102 |
| 5.13 | Enterprise South Bridge Interface (ESI)..... | 102 |
| 5.13.1 | ESI Port as a PCI Express Gen1 Port | 102 |
| 5.13.2 | Configuration Retry Completion | 102 |
| 5.13.3 | Inbound Transactions | 102 |
| 5.13.4 | Outbound Transactions | 103 |
| 5.13.5 | 64-Bit Addressing..... | 106 |
| 5.13.6 | Transaction Descriptor | 106 |
| 5.13.7 | Completer ID..... | 108 |
| 5.14 | Flow Control Credits Advertised on ESI | 108 |



| | | |
|----------|--|-----|
| 6 | Ordering | 109 |
| 6.1 | Introduction | 109 |
| 6.2 | Inbound Ordering Rules | 110 |
| 6.2.1 | Inbound Ordering Requirements | 110 |
| 6.2.2 | Special Ordering Relaxations | 111 |
| 6.3 | Outbound Ordering Rules | 111 |
| 6.3.1 | Outbound Ordering Requirements | 112 |
| 6.3.2 | Hinted Peer-to-Peer | 112 |
| 6.3.3 | Local Peer-to-Peer | 112 |
| 6.4 | Interrupt Ordering Rules | 113 |
| 6.4.1 | SpcEOI Ordering | 113 |
| 6.4.2 | SpcINTA Ordering | 113 |
| 6.5 | Configuration Register Ordering Rules | 113 |
| 6.6 | Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d) Ordering Exceptions | 114 |
| 7 | System Address Map | 115 |
| 7.1 | Memory Address Space | 115 |
| 7.1.1 | System DRAM Memory Regions | 117 |
| 7.1.2 | VGA/SMM and Legacy C/D/E/F Regions | 117 |
| 7.1.3 | Address Region Between 1 MB and TOLM | 119 |
| 7.1.4 | Address Region from TOLM to 4 GB | 119 |
| 7.1.5 | Address Regions above 4 GB | 122 |
| 7.1.6 | Protected System DRAM Regions | 123 |
| 7.2 | I/O Address Space | 124 |
| 7.2.1 | VGA I/O Addresses | 124 |
| 7.2.2 | ISA Addresses | 124 |
| 7.2.3 | CFC/CF8 Addresses | 124 |
| 7.2.4 | PCI Express Device I/O Addresses | 125 |
| 7.3 | Configuration/CSR Space | 125 |
| 7.3.1 | PCI Express Configuration Space | 125 |
| 7.3.2 | Processor CSR Space | 125 |
| 7.4 | IOH Address Map Notes | 125 |
| 7.4.1 | Memory Recovery | 125 |
| 7.4.2 | Non-Coherent Address Space | 126 |
| 7.5 | IOH Address Decoding | 126 |
| 7.5.1 | Outbound Address Decoding | 126 |
| 7.5.2 | Inbound Address Decoding | 129 |
| 7.6 | Intel VT-d Address Map Implications | 133 |
| 8 | Interrupts | 135 |
| 8.1 | Overview | 135 |
| 8.2 | Legacy PCI Interrupt Handling | 135 |
| 8.2.1 | Summary of PCI Express INTx Message Routing | 136 |
| 8.2.2 | Integrated I/OxAPIC | 137 |
| 8.2.3 | PCI Express INTx Message Ordering | 139 |
| 8.2.4 | INTR_Ack/INTR_Ack_Reply Messages | 140 |
| 8.3 | MSI | 140 |
| 8.3.1 | Interrupt Remapping | 142 |
| 8.3.2 | MSI Forwarding: IA-32 Processor-based Platform | 143 |
| 8.3.3 | MSI Forwarding: Intel Itanium Processor 9300 Series Based Platform | 149 |
| 8.3.4 | External I/OxAPIC Support | 150 |
| 8.4 | Virtual Legacy Wires | 150 |
| 8.5 | Platform Interrupts | 151 |
| 8.5.1 | GPE Events | 151 |
| 8.5.2 | PMI/SMI/NMI/MCA/INIT | 153 |
| 8.5.3 | CPEI | 154 |



| | | |
|-----------|--|-----|
| 9 | System Manageability | 155 |
| 9.1 | Introduction | 155 |
| 9.2 | Error Status and Logging..... | 155 |
| 9.3 | Component Stepping Information | 155 |
| 9.4 | Intel® Interconnect Built-In Self Test | 155 |
| 9.5 | Hot-Plug Status Access | 155 |
| 9.6 | Link Status Indication | 155 |
| 9.7 | Thermal Sensor..... | 156 |
| 10 | Power Management | 157 |
| 10.1 | Introduction | 157 |
| 10.2 | Supported Processor Power States..... | 158 |
| 10.3 | Supported System Power States..... | 158 |
| 10.3.1 | Supported Performance States | 159 |
| 10.3.2 | Supported Device Power States..... | 159 |
| 10.3.3 | Supported ESI Power States | 160 |
| 10.4 | Intel® QuickData Technology Power Management | 160 |
| 10.4.1 | Power Management with Assistance from OS Level Software | 160 |
| 10.5 | Device and Slot Power Limits | 161 |
| 10.5.1 | ESI Power Management | 161 |
| 10.6 | PCI Express Interface Power Management Support | 162 |
| 10.6.1 | Power Management Messages | 163 |
| 10.7 | Other Power Management Features | 163 |
| 10.7.1 | Fine-Grained Dynamic Clock Gating | 163 |
| 10.7.2 | Core Power Domains..... | 163 |
| 10.7.3 | LOs on Intel QuickPath Interconnect and PCIe | 163 |
| 10.7.4 | L1 on Intel QuickPath Interconnect and PCIe..... | 164 |
| 10.7.5 | Static Clock Gating | 164 |
| 11 | Partitioning | 165 |
| 11.1 | Partitioning Overview..... | 165 |
| 11.1.1 | Types of Partitioning | 165 |
| 11.1.2 | Configuration of Partition – Static & Dynamic | 167 |
| 11.2 | Hard Partitioning | 168 |
| 11.2.1 | Hard Partitioning Topologies | 169 |
| 11.3 | Virtual Partitioning..... | 173 |
| 11.4 | Hard Partition System Address Model..... | 173 |
| 11.5 | IOH Partitioning Support | 173 |
| 11.5.1 | Considerations for Multiple ICHs in a Partition | 174 |
| 12 | Scalable Systems | 175 |
| 12.1 | Introduction | 175 |
| 12.2 | Intel QuickPath Interconnect Standard and Extended Headers | 175 |
| 12.3 | Broadcast from IOH..... | 175 |
| 12.3.1 | Coherent..... | 175 |
| 12.3.2 | Non-Coherent (IA-32 Only)..... | 176 |
| 12.3.3 | Lock Arbiter (IA-32 Only) | 176 |
| 12.3.4 | Quiesce Master | 176 |
| 12.4 | Source Address Decoder..... | 176 |
| 12.5 | Performance | 177 |
| 12.6 | Time-Out | 177 |
| 12.7 | PCI Segments | 177 |
| 12.8 | Flat System Configurations | 177 |
| 12.9 | Hierarchical System Configurations | 179 |
| 13 | Intel® Management Engine | 181 |
| 13.1 | Intel Management Engine Overview | 181 |



| | | |
|-----------|---|------------|
| 13.2 | Management Engine External Interaction | 181 |
| 13.2.1 | Receive | 181 |
| 13.2.2 | Transmit..... | 181 |
| 13.3 | Controller Link (CLINK)..... | 183 |
| 13.4 | MESW Register..... | 183 |
| 13.4.1 | MESW_CBM_OVR_CTRL: Config Busmaster Override Control | 183 |
| 14 | Reset | 185 |
| 14.1 | Introduction..... | 185 |
| 14.1.1 | Reset Types..... | 185 |
| 14.1.2 | Reset Triggers..... | 187 |
| 14.1.3 | Trigger and Reset Type Association | 187 |
| 14.1.4 | Domain Behavior | 188 |
| 14.1.5 | Reset Sequences | 189 |
| 14.1.6 | Intel QuickPath Interconnect Reset | 191 |
| 14.2 | Platform Signal Routing Diagram | 194 |
| 14.3 | Platform Timing Diagrams..... | 195 |
| 15 | Component Clocking | 201 |
| 15.1 | Component Specification..... | 201 |
| 15.1.1 | Reference Clocks | 201 |
| 15.1.2 | JTAG..... | 201 |
| 15.1.3 | SMBus..... | 201 |
| 15.1.4 | Hot-Plug Serial Buses | 201 |
| 15.1.5 | RMII Bus | 202 |
| 15.1.6 | CLINK Bus | 202 |
| 15.1.7 | Intel Management Engine Clock | 202 |
| 15.1.8 | Clock Pin Descriptions | 202 |
| 15.1.9 | High Frequency Clocking Support..... | 203 |
| 15.2 | Miscellaneous Requirements and Limitations..... | 204 |
| 16 | Reliability, Availability, Serviceability (RAS) | 205 |
| 16.1 | RAS Overview | 205 |
| 16.2 | System Level RAS..... | 206 |
| 16.2.1 | Boot Processor | 206 |
| 16.2.2 | Inband System Management..... | 206 |
| 16.2.3 | Outband System Management | 206 |
| 16.2.4 | Dynamic Partitioning | 207 |
| 16.3 | IOH RAS Support | 207 |
| 16.4 | IOH Error Reporting | 207 |
| 16.4.1 | Error Severity Classification..... | 209 |
| 16.4.2 | Inband Error Reporting..... | 210 |
| 16.4.3 | IOH Error Registers Overview..... | 215 |
| 16.4.4 | Error Logging Summary..... | 223 |
| 16.5 | Intel QuickPath Interconnect Interface RAS | 227 |
| 16.5.1 | Link Level CRC and Retry..... | 227 |
| 16.5.2 | Intel QuickPath Interconnect Error Detection, Logging, and Reporting..... | 228 |
| 16.6 | PCI Express RAS..... | 228 |
| 16.6.1 | PCI Express* Link CRC and Retry..... | 228 |
| 16.6.2 | Link Retraining and Recovery | 229 |
| 16.6.3 | PCI Express Error Reporting Mechanism..... | 229 |
| 16.7 | IOH Error Handling Summary | 231 |
| 16.8 | IOH PCIe Hot Add/Remove Support..... | 244 |
| 16.8.1 | Hot Add/Remove Rules..... | 245 |
| 16.8.2 | PCI Express Hot-Plug | 245 |
| 16.9 | Virtual Pin Ports (VPP) | 249 |
| 16.10 | Operation | 249 |
| 16.10.1 | Intel QuickPath Interconnect Hot-Plug | 251 |



| | | |
|-----------|---|------------|
| 16.10.2 | IOH Hot-plug | 253 |
| 16.10.3 | SMBus and Memory Hot-plug | 253 |
| 17 | Intel® Trusted Execution Technology | 255 |
| 17.1 | Introduction | 255 |
| 18 | Intel® Virtualization Technology | 257 |
| 18.1 | Introduction | 257 |
| 18.2 | Intel VT-d | 257 |
| 18.3 | Intel VT-d2 Features | 257 |
| 18.4 | Other Virtualization Features Supported | 258 |
| 19 | Signal List | 259 |
| 19.1 | Conventions | 259 |
| 19.2 | Signal List | 260 |
| 19.3 | PCI Express Width Strapping | 267 |
| 19.4 | IOH Signal Strappings | 268 |
| 20 | DC Electrical Specifications | 271 |
| 20.1 | DC Characteristics | 271 |
| 20.2 | PCI Express* / ESI Interface DC Characteristics | 272 |
| 20.3 | Miscellaneous DC Characteristics | 273 |
| 21 | Configuration Register Space | 277 |
| 21.1 | Device Mapping: Functions Specially Routed by the IOH | 277 |
| 21.2 | Unimplemented Devices/Functions and Registers | 278 |
| 21.2.1 | Register Attribute Definition | 278 |
| 21.3 | RID Implementation in IOH | 280 |
| 21.3.1 | Background | 280 |
| 21.3.2 | Stepping Revision ID (SRID) | 280 |
| 21.3.3 | Conceptual Description | 280 |
| 21.4 | Standard PCI Configuration Space (0x0 to 0x3F) - Type 0/1 Common Configuration Space | 281 |
| 21.4.1 | Configuration Register Map | 281 |
| 21.4.2 | Register Definitions - Common | 282 |
| 21.4.3 | Register Definitions - Extended Config Space | 291 |
| 21.5 | IOxAPIC Controller | 312 |
| 21.5.1 | PCICMD: PCI Command Register (Dev #19) | 313 |
| 21.5.2 | PCISTS: PCI Status Register (Dev #19) | 315 |
| 21.5.3 | MBAR: IOxAPIC Base Address Register | 316 |
| 21.5.4 | ABAR: I/OxAPIC Alternate BAR | 317 |
| 21.5.5 | PMCAP: Power Management Capabilities Register | 317 |
| 21.5.6 | PMCSR: Power Management Control and Status Register | 318 |
| 21.5.7 | RDINDEX: Alternate Index to read Indirect I/OxAPIC Registers | 319 |
| 21.5.8 | RDWINDOW: Alternate Window to read Indirect I/OxAPIC Registers | 320 |
| 21.5.9 | IOAPICTETPC: IOxAPIC Table Entry Target Programmable Control | 320 |
| 21.5.10 | MBAR: IOxAPIC Base Address Register | 321 |
| 21.5.11 | ABAR: I/OxAPIC Alternate BAR | 322 |
| 21.5.12 | PMCAP: Power Management Capabilities Register | 322 |
| 21.5.13 | PMCSR: Power Management Control and Status Register | 323 |
| 21.5.14 | RDINDEX: Alternate Index to read Indirect I/OxAPIC Registers | 324 |
| 21.5.15 | RDWINDOW: Alternate Window to read Indirect I/OxAPIC Registers | 325 |
| 21.5.16 | IOAPICTETPC: IOxAPIC Table Entry Target Programmable Control | 325 |
| 21.5.17 | I/OxAPIC Memory Mapped Registers | 326 |
| 21.5.18 | Index Register | 327 |
| 21.5.19 | Window Register | 328 |
| 21.5.20 | PAR Register | 328 |
| 21.5.21 | EOI Register | 328 |
| 21.5.22 | APICID | 329 |
| 21.5.23 | Version | 329 |



| | | |
|-----------|--|------------|
| 21.5.24 | ARBID | 330 |
| 21.5.25 | BCFG | 330 |
| 21.5.26 | RTL[0:23]: Redirection Table Low DWORD | 330 |
| 21.5.27 | RTH[0:23]: Redirection Table High DWORD | 332 |
| 21.6 | Intel® VT, Address Mapping, System Management, Device Hide, Misc | 333 |
| 21.6.1 | GENPROTRANGE0.BASE: Generic Protected Memory Range 0 Base Address Register | 335 |
| 21.6.2 | GENPROTRANGE0.LIMIT: Generic Protected Memory Range 0 Limit Address Register | 335 |
| 21.6.3 | IOHMISCCTRL: IOH MISC Control Register | 336 |
| 21.6.4 | IOHMISCSS: IOH MISC Status | 337 |
| 21.6.5 | IOH System Management Registers | 338 |
| 21.6.6 | Semaphore and Scratch Pad Registers (Dev20, Function 1) | 362 |
| 21.6.7 | IOH System/Control Status Registers | 369 |
| 21.7 | Global Error Registers | 385 |
| 21.7.1 | Global Error Registers | 386 |
| 21.8 | IOH Local Error Registers | 395 |
| 21.8.1 | IOH Local Error Register | 398 |
| 21.8.2 | IOHERRST: IOH Core Error Status Register | 409 |
| 21.8.3 | THRERRST: Thermal Error Status | 414 |
| 21.8.4 | MIERRST: Miscellaneous Error Status | 417 |
| 21.8.5 | QPI[1:0]FERRFLIT0: Intel QuickPath Interconnect FERR FLIT log Register 0 | 421 |
| 21.9 | On-Die Throttling Register Map and Coarse-Grained Clock Gating | 423 |
| 21.9.1 | On-Die Throttling Registers | 424 |
| 21.10 | Intel QuickPath Interconnect Register Map | 428 |
| 21.11 | Intel QuickPath Interconnect Link Layer Registers | 429 |
| 21.11.1 | Intel QuickPath Interconnect Link Layer Register Tables | 430 |
| 21.11.2 | Intel QuickPath Interconnect Routing and Protocol Layer Registers | 440 |
| 21.11.3 | Intel QuickPath Interconnect Physical Layer Registers | 466 |
| 21.12 | PCI Express, ESI Configuration Space Registers | 474 |
| 21.12.1 | Other Register Notes | 474 |
| 21.12.2 | Standard PCI Configuration Space (0x0 to 0x3F) - Type 0/1 Common Configuration Space | 482 |
| 21.12.3 | Standard PCI Configuration Space (0x0 to 0x3F) - Type 1 - Only Common Configuration Space | 491 |
| 21.12.4 | Device-Specific PCI Configuration Space - 0x40 to 0xFF | 498 |
| 21.12.5 | PCI Express Enhanced Configuration Space | 525 |
| 21.12.6 | XP Common Block Link Control Registers | 545 |
| 21.13 | IOH Defined PCI Express Error Registers | 547 |
| 21.13.1 | XPCORERRSTS - XP Correctable Error Status Register | 547 |
| 21.13.2 | XPCORERRMSK - XP Correctable Error Mask Register | 547 |
| 21.13.3 | XPUNCERRSTS - XP Uncorrectable Error Status Register | 548 |
| 21.13.4 | XPUNCERRMSK - XP Uncorrectable Error Mask Register | 548 |
| 21.13.5 | XPUNCERRSEV - XP Uncorrectable Error Severity Register | 549 |
| 21.13.6 | XPGLBERRSTS - XP Global Error Status Register | 552 |
| 21.13.7 | XPGLBERRPTR - XP Global Error Pointer Register | 552 |
| 21.13.8 | CTOCTRL: Completion Time-Out Control Register | 553 |
| 21.13.9 | PCIE_LER_SS_CTRLSTS: PCI Express Live Error Recovery/Stop and Scream Control and Status Register | 553 |
| 21.13.10 | XP[10:0]ERRCNTSEL: Error Counter Selection Register | 554 |
| 21.13.11 | XP[10:0]ERRCNT: Error Counter Register | 555 |
| 21.14 | Intel VT-d Memory Mapped Register | 555 |
| 21.14.1 | Intel VT-d Memory Mapped Registers | 558 |
| 22 | Ballout and Package Information | 573 |
| 22.1 | I/O Hub (IOH) Ballout | 573 |



| | | |
|------|--------------------------------|-----|
| 22.2 | IOH Pin List and Ballout | 574 |
| 22.3 | Package Information..... | 614 |

Figures

| | | |
|------|--|-----|
| 1-1 | IOH High-Level Block Diagram | 20 |
| 2-1 | 4-Socket 2-IOH MP Topology | 28 |
| 2-2 | 2-Socket 2-IOH Topology | 28 |
| 2-3 | 8-Socket Glueless MP Topology | 29 |
| 2-4 | Hard Partitioned MP Topology | 30 |
| 2-5 | Example Server Topology (For Reference Only) | 31 |
| 3-1 | PCI Express* Interface Partitioning | 39 |
| 3-2 | SMBus Block-Size Configuration Register Read | 49 |
| 3-3 | SMBus Block-Size Memory Register Read..... | 50 |
| 3-4 | SMBus Word-Size Configuration Register Read | 50 |
| 3-5 | SMBus Word-Size Memory Register Read..... | 51 |
| 3-6 | SMBus Byte-Size Configuration Register Read | 51 |
| 3-7 | SMBus Byte-Size Memory Register Read..... | 52 |
| 3-8 | SMBus Block-Size Configuration Register Write..... | 53 |
| 3-9 | SMBus Block-Size Memory Register Write | 53 |
| 3-10 | SMBus Word-Size Configuration Register Write..... | 53 |
| 3-11 | SMBus Word-Size Memory Register Write | 53 |
| 3-12 | SMBus Configuration Byte Write, PEC Enabled | 54 |
| 3-13 | SMBus Memory Byte Write, PEC Enabled | 54 |
| 4-1 | Intel® QuickPath Interconnect Packet Visibility By The Physical Layer (Phit)..... | 59 |
| 4-2 | Intel® QuickPath Interconnect Packet Visibility By Link Layer (Flit) | 61 |
| 4-3 | Example NodeID Assignment 4-Socket, 2-IOH Platform (For Intel Itanium Processor 9300 Series based Platform)..... | 66 |
| 4-4 | Lock Arbiter Issue/Control State-Machine | 79 |
| 7-1 | System Address Map | 116 |
| 7-2 | VGA/SMM and Legacy C/D/E/F Regions | 117 |
| 7-3 | Peer-to-Peer Illustration..... | 130 |
| 8-1 | Legacy Interrupt Routing Illustration (INTA Example)..... | 136 |
| 8-2 | Interrupt Transformation Table Entry (IRTE) | 143 |
| 8-3 | Assert/Deassert_(HP, PME) GPE Messages | 152 |
| 8-4 | Intel QuickPath Interconnect GPE Messages from Processor and DO_SCI Messages from IOH | 153 |
| 10-1 | ACPI Power States in G0 and G1 States..... | 157 |
| 10-2 | Typical Platform Showing Power Saving Signals to BMC | 158 |
| 10-3 | ICH Timing Diagram for S4,S5 Transition..... | 162 |
| 11-1 | Partition Type and Isolation/Flexibility Chart | 166 |
| 11-2 | Example of Hard Partitioning..... | 169 |
| 11-3 | Example of Dynamic Hard Partitioning Violation..... | 170 |
| 11-4 | Examples of Hard Partitioning in 2-CPU Systems | 171 |
| 11-5 | Examples of Hard Partitioning in Topology 4-2-F (4 CPU, 2 IOH) | 172 |
| 11-6 | Legal Hard Partitioning..... | 172 |
| 12-1 | Example NodeID Assignment for 2-Cluster System with 4S Clusters..... | 175 |
| 12-2 | 8 Processor Topology Example | 178 |
| 12-3 | 8 Processor Topology Example (Intel Itanium Processor 9300 Series-based Platform Only) | 178 |
| 12-4 | Hierarchical System Example | 179 |
| 13-1 | Example of Intel ME Configuration with Intel SPS Implementation..... | 182 |
| 14-1 | Physical Layer Power-Up and Initialization Sequence | 192 |



| | | |
|-------|---|-----|
| 14-2 | Inband Reset Sequence Initiated by Port A to Port B..... | 192 |
| 14-3 | Basic Reset Distribution | 194 |
| 14-4 | Basic MP System Reset Distribution..... | 194 |
| 14-5 | Power-Up (example) | 196 |
| 14-6 | COREPWRGOOD Reset (example) | 197 |
| 14-7 | Hard Reset (Example) | 198 |
| 14-8 | IOH CORERST_N Re-Triggering Limitations..... | 199 |
| 16-1 | Error Signal Converted to Interrupt Example | 208 |
| 16-2 | Error Signal Converted to Error Pins Example | 208 |
| 16-3 | IOH Error Registers..... | 216 |
| 16-4 | Local Error Signaling on IOH Internal Errors | 218 |
| 16-5 | IOH Error Logging and Reporting Example..... | 220 |
| 16-6 | Global Error Logging and Reporting | 221 |
| 16-7 | Thermalert and Thermtrip Signaling | 222 |
| 16-8 | IOH Error Logging Flow | 224 |
| 16-9 | Clearing Global and Local FERR/NERR Registers | 226 |
| 16-10 | Error Signaling to IOH Global Error Logic on a PCI Express Interface Error | 230 |
| 16-11 | PCI Express Error Standard..... | 231 |
| 16-12 | PCI Express Hot-Plug Interrupt Flow..... | 248 |
| 16-13 | Intel QPI Hot Add/Remove Support | 252 |
| 20-1 | Differential Measurement Point for Rise and Fall Time | 274 |
| 20-2 | Differential Measurement Point for Ringback..... | 275 |
| 21-1 | PCI Express Root Port (Devices 1-10), ESI Port (Device 0) Type1 Configuration Space..... | 475 |
| 21-2 | Base Address of Intel VT-D Remap Engines | 558 |
| 22-1 | IOH Quadrant Map..... | 573 |
| 22-2 | IOH Ballout Left Side (Top View)..... | 574 |
| 22-3 | IOH Ballout Center Side (Top View)..... | 576 |
| 22-4 | IOH Ballout Right (Top View)..... | 578 |
| 22-5 | Package Diagram..... | 614 |

Tables

| | | |
|------|--|----|
| 1-1 | High-Level Feature Summary | 20 |
| 1-2 | Terminology..... | 23 |
| 1-3 | Related Documents | 26 |
| 3-1 | Intel QuickPath Interconnect Frequency Strapping Options | 34 |
| 3-2 | Protocol Transactions Supported | 36 |
| 3-3 | Supported Degraded Modes | 40 |
| 3-4 | SMBus Command Encoding | 45 |
| 3-5 | Internal SMBus Protocol Stack | 46 |
| 3-6 | SMBus Slave Address Format | 46 |
| 3-7 | Memory Region Address Field | 47 |
| 3-8 | Status Field Encoding for SMBus Reads..... | 48 |
| 3-9 | Memory Region Address Field | 55 |
| 3-10 | JTAG Configuration Register Access..... | 55 |
| 4-1 | Link Layer Parameter Values | 61 |
| 4-2 | Credit Values Programming Example | 63 |
| 4-3 | Supported Intel QuickPath Interconnect Message Classes..... | 64 |
| 4-4 | Memory Address Decoder Fields..... | 68 |
| 4-5 | I/O Decoder Entries | 69 |
| 4-6 | Inbound Coherent Transactions and Responses..... | 71 |



| | | |
|------|---|-----|
| 4-7 | Non-Coherent Inbound Transactions Supported | 72 |
| 4-8 | Snoops Supported and State Transitions | 74 |
| 4-9 | Protocol Transactions Supported | 74 |
| 4-10 | Profile Control – Intel Xeon Processor 7500 Series based Platform Only | 76 |
| 4-11 | Profile Control – Intel Itanium Processor 9300 Series-based Platform Only | 76 |
| 4-12 | Lock Master Participant List Usage | 79 |
| 4-13 | Time-Out Level Classification for IOH | 83 |
| 4-14 | Local-Local Conflict Actions | 84 |
| 4-15 | Remote-Local Conflict Actions | 85 |
| 4-16 | Conflict Completions Actions | 85 |
| 5-1 | Supported Degraded Modes | 89 |
| 5-2 | Incoming PCI Express Memory, I/O and Configuration Request/Completion Cycles | 91 |
| 5-3 | Incoming PCI Express* Message Cycles | 93 |
| 5-4 | Outgoing PCI Express* Memory, I/O and Configuration Request/Completion Cycles | 94 |
| 5-5 | Outgoing PCI Express Message Cycles | 94 |
| 5-6 | PCI Express Transaction ID Handling | 96 |
| 5-7 | PCI Express Attribute Handling | 97 |
| 5-8 | PCI Express CompleterID Handling | 97 |
| 5-9 | PCI Express Credit Mapping for Inbound Transactions | 100 |
| 5-10 | PCI Express Credit Mapping for Outbound Transactions | 101 |
| 5-11 | Incoming ESI Memory, I/O and Configuration Requests/Completions | 103 |
| 5-12 | Incoming ESI Messages | 103 |
| 5-13 | Outgoing ESI Memory, I/O and Configuration Requests/Completions | 104 |
| 5-14 | Outgoing ESI Messages | 104 |
| 5-15 | ESI Transaction ID Handling | 107 |
| 5-16 | ESI Attribute Handling | 107 |
| 5-17 | ESI CompleterID Handling | 108 |
| 5-18 | ESI Credit Mapping | 108 |
| 6-1 | Ordering Term Definitions | 109 |
| 7-1 | Outbound Target Decoder Entries | 128 |
| 7-2 | Decoding of Outbound Memory Requests from Intel QuickPath Interconnect (from CPU or Remote Peer-to-Peer) | 129 |
| 7-3 | Subtractive Decoding of Outbound I/O Requests from Common System Interface | 129 |
| 7-4 | Inbound Memory Address Decoding | 131 |
| 7-5 | Inbound I/O Address Decoding | 132 |
| 8-1 | Interrupt Sources in I/OxAPIC Table Mapping | 137 |
| 8-2 | I/OxAPIC Table Mapping to PCI Express Interrupts | 138 |
| 8-3 | Programmable IOxAPIC Entry Target for Certain Interrupt Sources | 138 |
| 8-4 | MSI Address Format when Remapping is Disabled | 141 |
| 8-5 | MSI Data Format when Remapping Disabled | 141 |
| 8-6 | MSI Address Format when Remapping is Enabled | 142 |
| 8-7 | MSI Data Format when Remapping is Enabled | 142 |
| 8-8 | Interrupt Delivery | 143 |
| 8-9 | IA-32 Physical APICID to NodeID Mapping | 144 |
| 8-10 | IA-32 Interrupt Delivery Summary | 147 |
| 8-11 | Itanium® Processor Family Platform Interrupt Delivery | 149 |
| 8-12 | Intel Itanium Processor 9300 Series APICID to NodeID Mapping Example (Setup through QPIPAPICSAD Register) | 150 |
| 9-1 | Status Register Location Table | 156 |
| 10-1 | IOH Platform Supported System States | 159 |



| | | |
|-------|--|-----|
| 10-2 | System and ESI Link Power States | 160 |
| 13-1 | Signal Type Definition | 181 |
| 13-2 | Controller Link Interface | 183 |
| 14-1 | Trigger and Reset Type Association | 187 |
| 14-2 | Intel QuickPath Interconnect Inband Reset Events..... | 193 |
| 14-3 | Core Power-Up, Core POWERGOOD, and Core Hard Reset Platform Timings | 199 |
| 15-1 | The Clock Options for a Intel ME and Non-Intel ME Configuration System | 202 |
| 16-1 | Clock Pins | 202 |
| 16-1 | Error Counter Register Locations..... | 227 |
| 16-2 | IOH Default Error Severity Map..... | 232 |
| 16-3 | IOH Error Summary | 232 |
| 16-4 | Hot-Plug Interface | 246 |
| 16-5 | I/O Port Registers in On-Board SMBus Devices Supported by IOH..... | 249 |
| 16-6 | Hot-Plug Signals on the Virtual Pin Port | 250 |
| 16-7 | Write Command | 250 |
| 16-8 | Read Command | 251 |
| 19-1 | Buffer Technology Types..... | 259 |
| 19-2 | Buffer Signal Directions | 259 |
| 19-3 | Signal Naming Conventions..... | 260 |
| 19-4 | JTAG Signals | 260 |
| 19-5 | Intel QuickPath Interconnect Signals | 261 |
| 19-6 | PCI Express Signals | 261 |
| 19-7 | Signals | 262 |
| 19-8 | MISC Signals..... | 262 |
| 19-9 | Controller Link Signals..... | 264 |
| 19-10 | RMII Signals | 265 |
| 19-11 | Power and Ground | 265 |
| 19-12 | PEWIDTH[5:0] Strapping Options | 267 |
| 20-1 | Clock DC Characteristics | 271 |
| 20-2 | PCI Express / ESI Differential Transmitter (Tx) Output DC Characteristics..... | 272 |
| 20-3 | PCI Express / ESI Differential Receiver (Rx) Input DC Characteristics..... | 272 |
| 20-4 | CMOS, JTAG, SMBUS, GPIO3.3V, and MISC DC Characteristics..... | 273 |
| 21-1 | Functions Specially Handled by the IOH | 277 |
| 21-2 | Register Attributes Definitions | 278 |
| 21-3 | PCIe Capability Registers for Devices with PCIe Extended Configuration Space | 281 |
| 21-4 | IOH Device 19 I/OxAPIC Configuration Map - Offset 0x00-0xFF..... | 312 |
| 21-5 | I/OxAPIC Direct Memory Mapped Registers | 327 |
| 21-6 | I/OxAPIC Indexed Registers (Redirection Table Entries) | 329 |
| 21-7 | Core Registers (Dev 20, Function 0) - Offset 0x00-0xFF | 333 |
| 21-8 | Core Registers (Dev 20, Function 0) | 334 |
| 21-9 | Semaphore and Scratch pad Registers (Dev 20, Function 1) | 362 |
| 21-10 | IOH Control/Status & Global Error Register Map (Dev 20, Function 2, Page 1 of 4) ... | 369 |
| 21-11 | IOH Control/Status & Global Error Register Map (Dev 20, Function 2, Page 2 of 4) ... | 370 |
| 21-12 | IOH Local Error Map #1 (Dev 20, Function 2, Page 3 of 4)..... | 371 |
| 21-13 | IOH Local Error Map #2 (Dev 20, Function 2, Page 4 of 4)..... | 372 |
| 21-14 | IOH Control/Status & Global Error Register Map (Dev 20, Function 2) | 385 |
| 21-15 | IOH Local Error Map #1 (Dev 20, Function 2) | 395 |
| 21-16 | IOH Local Error Map #2 (Dev 20, Function 2) | 396 |
| 21-17 | IOH Local Error Map #2 (Dev 20, Function 2, Page 4 of 4)..... | 397 |
| 21-18 | Device 20, Function 3: On-Die Throttling and Coarse-Grained Clock Gating..... | 423 |
| 21-19 | Intel QuickPath Interconnect Link Map Port 0 (Dev 16), Port 1 (Dev 17) | 429 |
| 21-20 | CSR Intel QPI Routing Layer, Protocol (Dev 16, Function 1) | 440 |



| | |
|--|-----|
| 21-21 QPIPH-Intel QuickPath Interconnect Tracking State Table | 466 |
| 21-22 QPIPH-Intel QuickPath Interconnect Tracking State Table | 470 |
| 21-23 IOH Device 0 (ESI mode) Configuration Map | 476 |
| 21-24 IOH Device 0 (ESI mode) Extended Configuration Map | 477 |
| 21-25 IOH Devices 0(ESI Mode) Configuration Map | 478 |
| 21-26 IOH Devices 0(PCIe Mode)-10 Legacy Configuration Map (PCI Express Registers) | 479 |
| 21-27 IOH Devices 0(PCIe Mode)-10 Extended Configuration Map (PCI Express Registers) Page#0 | 480 |
| 21-28 IOH Devices 0-10 Extended Configuration Map (PCI Express Registers) Page#1 | 481 |
| 21-29 MSI Vector Handling and Processing by IOH..... | 502 |
| 21-30 Intel VT-d Memory Mapped Registers - 0x00 - 0xFF, 1000-10FF | 555 |
| 21-31 Intel VT-d Memory Mapped Registers - 0x100 - 0x1FF, 0x1100-0x11FF | 556 |
| 22-1 Pin Listing by Pin Name | 580 |
| 22-2 Pin Listing by Signal Name | 597 |



Revision History

| Document Number | Revision Number | Description | Date |
|-----------------|-----------------|-----------------|------------|
| 322827 | -001 | Initial Release | March 2010 |

§



Product Features

- **Processor**
 - Intel® Xeon® Processor 7500 Series
 - Intel® Itanium® Processor 9300 Series
 - **Two Full-Width Intel® QuickPath Interconnect (Intel® QPI)**
 - Packetized protocol with 18 data/protocol bits and 2 CRC bits per link per direction
 - 4.8 GT/s, 5.86 GT/s, and 6.4 GT/s supporting different routing lengths
 - Fully-coherent write cache with inbound write combining
 - Read Current command support
 - Support for 64-byte cacheline size
 - **PCI Express* Features**
 - Two x16 PCI Express* Gen2 ports each supporting up to 8 GB/s/direction peak bandwidth
 - All ports are configurable as two independent x8 or four independent x4 interfaces
 - An additional x4 PCI Express Gen2 port configurable to 2 x 2 interfaces
 - Dual unidirectional links
 - Supports PCI Express Gen1 and Gen2 transfer rates
 - Peer-to-peer support between PCI Express interfaces
 - Support for multiple unordered inbound traffic streams
 - Support for Relaxed Ordering attribute
 - Full support for software-initiated PCI Express power management
 - x8 Server I/O Module (SIOM) support
 - Alternative Requester ID (ARI) capability
 - **Enterprise Southbridge Interface (ESI) Features**
 - One x4 ESI link interface supporting PCI Express Gen1 (2.5 Gbps) transfer rate
 - Intel® I/O Controller Hub (ICH) Support. Dedicated legacy bridge interface
 - **Supports Controller Link (CL)**
 - **Supports Intel® I/O Acceleration Technology (Intel® I/OAT) Gen3**
 - Includes previous generation Intel I/OAT features, focused on reduced latency and CPU utilization for I/O traffic
 - Increased bandwidth to support multiple 10 GbE links
 - Flow-through CRC
 - Virtualization friendly - Assignable channels, Inter/intra VM copy, page zeroing
 - **Supports Intel® Virtualization Technology for Directed I/O (Intel VT-d), Second Revision**
 - **Reliability, Availability, Serviceability (RAS)**
 - Supports SMBus Specification, Revision 2.0 slave interface for server management with Packet Error Checking
 - Improved RAS achieved by protecting internal data paths through ECC and parity protection mechanisms
 - Supports PCI Express Base Specification, Revision 2.0 CRC with link-level retry
 - Supports both standard and rolling Intel® QuickPath Interconnect CRC with link level retry
 - Advanced Error Reporting capability for PCI Express link interfaces
 - Native PCI Express Hot-Plug support
 - Error injection capabilities
 - Performance monitoring capabilities
 - Power Management
 - Intel QuickPath Interconnect hot-plug
 - **Security**
 - TPM 1.2 and Intel VT-d
 - **Package**
 - FC-BGA
 - 37.5 mm x 37.5 mm
 - 1295 balls
 - Full grid pattern
- Note:* Refer to Intel 7500 Chipset full specification and Intel 7500 Chipset specification update for further details.

§





1 Introduction

1.1 System Configuration Overview

The I/O Hub (IOH) component provides a connection point between various I/O components and Intel® QuickPath Interconnect based processors. platforms. Contact you local Intel representative for details on those processors.

For example topologies supported by the IOH, refer to [Chapter 2, “Platform Topology.”](#)

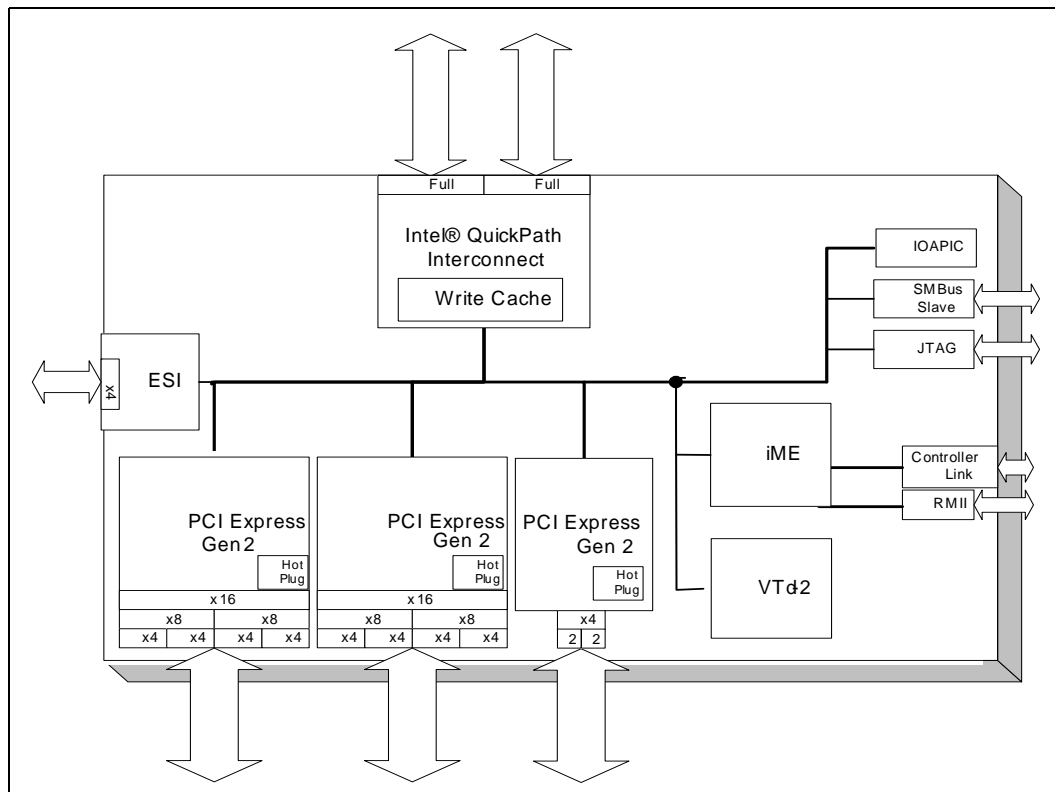
This document is for the common feature sets as well as feature differences between the platforms. Throughout the document, platform-specific features will be identified and spelled out. It is important for the reader to clearly comprehend the feature deltas and platform implications associated with designing the IOH for system compatibility.

1.2 Feature Summary

The IOH provides the interface between the processor Intel QuickPath Interconnect and industry-standard PCI Express* components. The two Intel QuickPath Interconnect interfaces are full-width links (20 lanes in each direction). The two x16 PCI Express Gen2 ports are also configurable as x8 and x4 links compliant to the *PCI Express Base Specification*, Revision 2.0. The single x4 PCI Express Gen2 port can bifurcate into two independent x2 interfaces. In addition, the legacy IOH supports a x4 ESI link interface for the legacy bridge. For multiprocessor platforms, non-legacy IOHs also support an additional x4 PCI Express Gen1 interface. Refer to [Figure 1-1](#) for a high-level view of the IOH and its interfaces. The IOH supports the following features and technologies:

- Intel® QuickPath Interconnect multiprocessor-small profile
- Intel® QuickPath Interconnect multiprocessor-enterprise profile (Intel® Itanium® processor 9300 series-based platforms)
 - Interface to CPU or other IOH (limited configurations)
- PCI Express Gen2
- Intel® I/O Accelerated Technology (Intel® I/OAT) Gen3 (updated DMA engine with virtualization enhancements)
- Intel® Management Engine (Intel® ME)

Figure 1-1. IOH High-Level Block Diagram



Note: The internal IOH interfaces are designed to communicate with each other. This communication is illustrated in the diagram above as a shared bus; however, this is a conceptual diagram and does not represent actual implementation and connectivity.

1.2.1 Features By Segment based on PCI Express Ports

The following table shows the features that will be supported for this chipset.

Table 1-1. High-Level Feature Summary

| SKU | Intel® QPI Ports | PCIe* Lanes | Manageability |
|--|------------------|--|---------------|
| Multiprocessor Intel Xeon processor 7500 series server | 2 | 36 lanes, 2x16's (bifurcatable to 4x8's or 8x4's), and 1x4 | Node Manager |
| Intel Itanium processor 9300 series server | 2 | 36 lanes, 2x16's (bifurcatable to 4x8's or 8x4's), and 1x4 | |

1.2.1.1 Addressability By Profile

The IOH supports MP addressability up to 46 bits for Intel Xeon processor 7500 series based platforms and 51 bits for Intel Itanium processor 9300 series-based platforms. Note that the CPU may further limit the addressability.



1.2.2 Non-Legacy IOH

For multiprocessor configurations where there is more than one IOH, one of the IOHs will be considered the legacy IOH. The legacy IOH will be connected to the ICH10 legacy component. For the remaining non-legacy IOHs, the ESI port to the ICH10 will be either configured as a Gen1 x4 PCIe* link, or treated as an ESI port that is disabled.

For additional details on configurations with more than one IOH, please refer to [Chapter 2, "Platform Topology"](#).

1.2.3 Intel® QuickPath Interconnect Features

- Two full-width Intel QuickPath Interconnect link interfaces:
- Packetized protocol with 18 data/protocol bits and 2 CRC bits per link per direction
 - Supporting 4.8 GT/s, 5.86 GT/s and 6.4 GT/s
- Fully-coherent write cache with inbound write combining
- Read Current command support
- Support for 64-byte cacheline size

1.2.4 PCI Express* Features

- Two x16 PCI Express* Gen2 ports each supporting up to 8 GB/s/direction peak bandwidth
 - All ports are configurable as two independent x8 or four independent x4 interfaces
- An additional x4 PCI Express Gen2 port configurable to 2 x2 interfaces
- An additional x4 PCI Express Gen1 port on non-legacy IOHs. This port is the ESI port on legacy IOHs
- Dual unidirectional links
- Supports PCI Express Gen1 and Gen2 transfer rates
- Peer-to-peer support between PCI Express interfaces
- Support for multiple unordered inbound traffic streams
- Support for Relaxed Ordering attribute
- Full support for software-initiated PCI Express power management
- x8 Server I/O Module (SIOM) support
- Alternative Requester ID (ARI) capability
- Auto negotiation is not a supported feature

1.2.5 Enterprise South Bridge Interface (ESI) Features

- One x4 ESI link interface supporting PCI Express Gen1 (2.5 Gbps) transfer rate
 - Dedicated legacy bridge (Intel® I/O Controller Hub (ICH)) interface
- ICH Support
- For non-legacy IOHs, this interface is configurable as a x4 Gen1 PCI Express port



1.2.6 Controller Link (CL)

The Controller Link is a private, low pin count, low power, communication interface between the IOH and ICH portions of the Manageability Engine subsystem.

1.2.7 Intel® I/OAT Gen3

- Includes previous generation Intel I/OAT features, focused on reduced latency and CPU utilization for I/O traffic
- Increased bandwidth to support multiple 10 GbE links
- Flow-through CRC
- Virtualization friendly
 - Assignable channels, Inter/intra VM copy, page zeroing

1.2.8 Intel® Virtualization Technology for Directed I/O (Intel® VT-d), Second Revision

- Builds upon first generation of Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d) features
- Improved performance through better invalidation architecture
- Support for end-point Address Translation Caching (ATC) compliant with the PCI-SIG IOV *Address Translation Services (ATS), Revision 1.0* specification.
- Interrupt remapping
- Optimized translation of sequential accesses
- IOV support (ARI)

1.2.9 Reliability, Availability, Serviceability (RAS) Features

- Supports an *SMBus Specification, Revision 2.0* slave interface for server management with Packet Error Checking:
 - SMBus and JTAG access to IOH configuration registers for out-of-band server management
- Improved RAS achieved by protecting internal datapaths through ECC and parity protection mechanisms
- Supports *PCI Express Base Specification, Revision 2.0* CRC with link-level retry
- Supports both standard and rolling Intel QuickPath Interconnect CRC with link level retry
- Advanced Error Reporting capability for PCI Express link interfaces
- Native PCI Express Hot-Plug support
- Error injection capabilities
- Performance monitoring capabilities
- PCIe Live Error Recovery
- Intel QuickPath Interconnect hot-plug
- Support for quiescing and re-instantiation of Intel® QuickPath Interconnect links
- Machine Check Architecture (MCA) support
- OS-Assisted CPU migration



- Self Healing Intel QuickPath Interconnect links

1.2.10 Power Management Support

- PCI Express Link states (L0, L0s-tx only, L1, and L3)
- Intel QuickPath Interconnect Link states (L0)
- ESI states (L0, L0s-tx only, L1)
- Intel ME states (M0, M1, Mdisable, Moff)
- System states (S0, S1, S4, S5)

1.2.11 Security

- TPM1.2 and Intel VT-d for server security

1.2.12 Other

- Integrated IOxAPIC
- Extended IA-32 mode - 32 bit APIC ID (Legacy IA-32 mode is 8 bits)

1.3 Terminology

Table 1-2 defines the acronyms, conventions, and terminology used throughout the specification.

Table 1-2. Terminology (Sheet 1 of 3)

| Term | Description |
|---------------------|--|
| APIC | Advanced Programmable Interrupt Controller |
| ASIC Repeater | A chip which intercepts the Intel QuickPath Interconnect traffic. It repeats the traffic, but also sends appropriate data to the logic analyzer. |
| BIST | Built-In Self Test |
| BMC | Baseboard Management Controller. A microcontroller used for remote platform management. |
| IOH | IOH with two Intel® QuickPath Interconnect ports and 36 PCIe* lanes |
| CA | Completer Abort |
| Caching Agent | Intel QuickPath Interconnect coherency agent that participates in the MESIF protocol. Caches copies of the coherent memory space, potentially from multiple home agents. May also support the read-only forwardable cache state F. |
| CEM | Refers to the PCI Express Card Electromechanical specification |
| CPEI | Correctable Platform Event Interrupt |
| CRC | Cyclic Redundancy Code |
| DMA | Direct Memory Access |
| EHCI | Enhanced Host Controller Interface |
| ESI | Enterprise South Bridge Interface is the interface to the I/O legacy bridge component of the ICH |
| FW | Firmware; software stored in ROM |
| Hinted Peer-to-Peer | A transaction initiated by an I/O agent destined for an I/O target within the same root port (PCIe port) |
| HOA | High Order Address |



Table 1-2. Terminology (Sheet 2 of 3)

| Term | Description |
|---|--|
| Home Agent | Intel® QuickPath Interconnect coherency agent that interfaces to the main memory and is responsible for tracking cache-state transitions |
| ICH10 | Intel® I/O Controller Hub, Tenth Generation |
| Inbound Transaction | Transactions initiated on a PCI Express port destined for an Intel® QuickPath Interconnect port |
| Intel® QPI | Intel® QuickPath Interconnect |
| Intel® QuickPath Interconnect Link Full Width | Intel® QuickPath Interconnect link with 20 physical lanes in each direction |
| IOH | I/O Hub |
| IRB | Inbound Request Buffer |
| Lane | A set of differential signal pairs: one pair for transmission and one pair for reception. A by-N Link is composed of N Lanes |
| LCI | LAN Connect Interface |
| Legacy ICH | The ICH that has legacy features enabled, and is typically where the firmware boot code resides |
| Legacy IOH | The IOH that has the Legacy ICH directly attached |
| Link | A dual-simplex communications path between two components. The collection of two ports and their interconnecting lanes. |
| Local Peer-to-Peer | A transaction initiated by an I/O agent destined for an I/O target within the same root complex |
| LOM | LAN on Motherboard |
| LPC | Low Pin Count |
| MP | Multi-processor |
| MSI | Message Signaled Interrupt |
| Oplin | Dual 10 Gb Ethernet Controller |
| ORB | Outgoing Request Buffer |
| Outbound Transactions | Transactions initiated on an Intel® QuickPath Interconnect port destined for a PCI Express or ESI port |
| PA | Physical Address |
| PCI Express Gen1 | Common reference for 1st generation PCI Express (Base Spec revision 1.x) and speed |
| PCI Express Gen2 | Common reference for 2nd generation PCI Express (Base Spec revision 2.x) and speed |
| PCM | Pulse Code Modulation |
| PLC | Platform LAN Connect |
| PME | Power Management Event |
| Port | In physical terms, a group of transmitters and receivers physically located on the IOH that define one side of a link |
| Processor | Common reference term for “processor socket”, used throughout this document |
| RAS | Reliability, Availability, Serviceability |
| Remote Peer-to-Peer | A transaction initiated by an I/O agent destined for an I/O target on a different root complex |
| RID | Revision ID of the IOH |
| RTA | Router Table Array |
| RTC | Real Time Clock |
| SATA | Serial ATA |
| SCMD | Sub Command |



Table 1-2. Terminology (Sheet 3 of 3)

| Term | Description |
|----------------|--|
| SEC | Single Error Correction |
| SMBus | System Management Bus. A two-wire interface through which various system components can communicate. |
| Socket | Processor (cores + uncore) |
| SPD | Serial Presence Detect |
| S/PDIF | Sony/Phillips Digital Interface |
| SPI | Serial Peripheral Interface. The interface for serial flash components. |
| SPS | Server Platform Services |
| SSP | System Service Processor |
| STD | Suspend To Disk |
| TCO | Total Cost of Ownership |
| UHCI | Universal Host Controller Interface |
| UR | Unsupported Request |
| USB | Universal Serial Bus |
| Intel VT-d | Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d) |
| Intel® 82575EB | Formerly code named "Zoar" for Dual Gigabit Ethernet Controller |



1.4 Related Documents

The reader of this document should also be familiar with the components, material and concepts presented in the documents listed in [Table 1-3](#).

Table 1-3. Related Documents

| Document | Comment ¹ |
|---|--|
| <i>Intel® 7500 Chipset Thermal Mechanical Design Guide</i> | www.intel.com |
| <i>Intel® Xeon® Processor 7500 Series Datasheet, Volume 1</i> | www.intel.com |
| <i>Intel® Xeon® Processor 7500 Series Datasheet, Volume 2</i> | www.intel.com |
| PCI Express Base Specification, Revision 1.1 | www.pcisig.com |
| PCI Express Base Specification, Revision 2.0 | www.pcisig.com |
| <i>Intel® I/O Controller Hub 9 (ICH9) Family Datasheet</i> | www.intel.com |
| <i>Intel® I/O Controller Hub 10 (ICH10) Family Datasheet</i> | www.intel.com |
| <i>SMBus Specification, Revision 2.0</i> | www.smbus.org |

§



2 Platform Topology

2.1 Introduction

The I/O Hub component (IOH) provides a connection point between various I/O components and Intel QuickPath Interconnect based processors. The IOH supports the Intel Xeon processor 7500 series and Intel Itanium processor 9300 series.

The Intel QuickPath Interconnect ports are used for processor-to-processor and processor-to-IOH connections.

2.2 IOH Supported Topologies

The IOH-based platform supports a subset of the possible system topologies. The supported configurations are specifically listed in the following figures. Note that the figures do not represent the different variations of processor and IOH population in the system, rather the figures represent the physical layout and connections of the topologies. The following terminology is used to describe the following topologies.

Terminology

Legacy Bridge: In the following figures, legacy bridge refers to the ICH component. The legacy bridge contains the legacy functions required for a industry standard operating system. The ICH is connected solely by the ESI port. The IOH has one ESI port capable of connecting to a legacy bridge. Legacy bridge connections are not explicitly illustrated in all figures. Readers should assume that only one IOH is connected to the active legacy bridge.

Legacy IOH: One IOH functions as a "Legacy IOH". The legacy IOH contains the central resources for the system and interfaces to the legacy bridge. The legacy IOH is the only IOH where the Intel Manageability Engine (Intel ME) will be enabled.

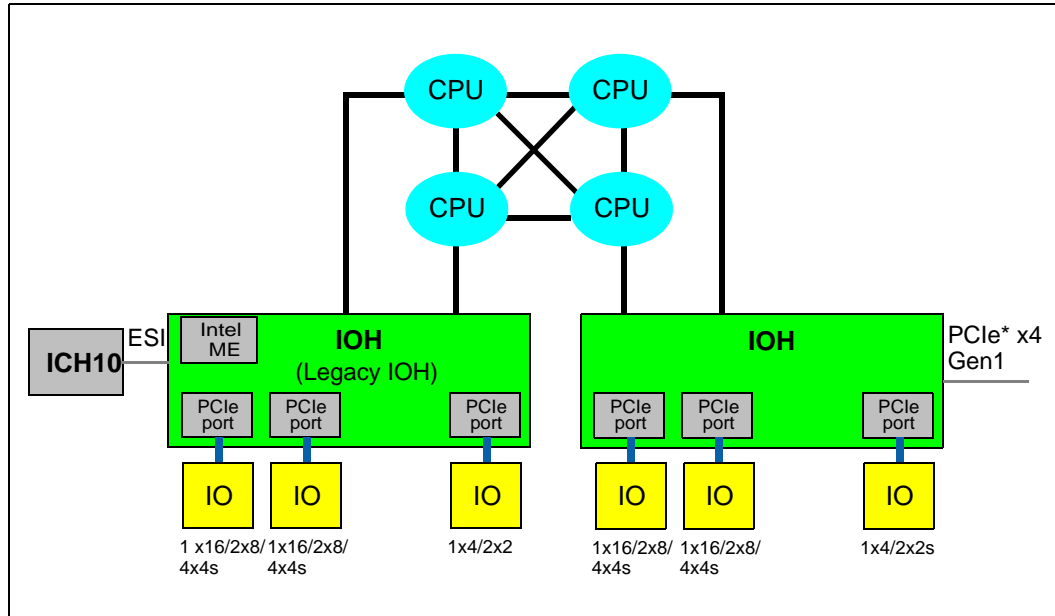
Non-Legacy IOH: In a non-partitioned system, the IOHs that are not the "Legacy IOH" are referred to as non-legacy IOHs. The ESI and Intel ME are not enabled and they are not connected to a legacy bridge (ICH).

2.2.1 Platform Topologies

This section illustrates platform topologies supported for the Intel Xeon processor 7500 series and Intel Itanium processor 9300 series.

Figure 2-1 illustrates the standard 4-socket 2-IOH topology.

Figure 2-1. 4-Socket 2-IOH MP Topology



Note:

1. For multiprocessor topologies, the unused ESI port on the non-legacy IOH can be configured as a Gen1 PCIe x4 link.
2. For partially populated systems, the ICH10 must be connected to an IOH that is directly connected to a CPU.

Figure 2-2 demonstrates a 2-socket 2-IOH topology.

Figure 2-2. 2-Socket 2-IOH Topology

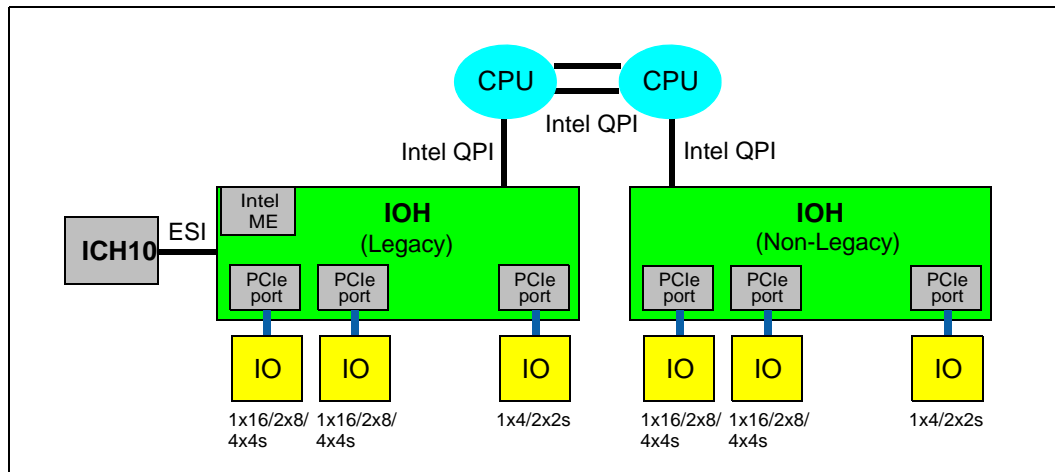
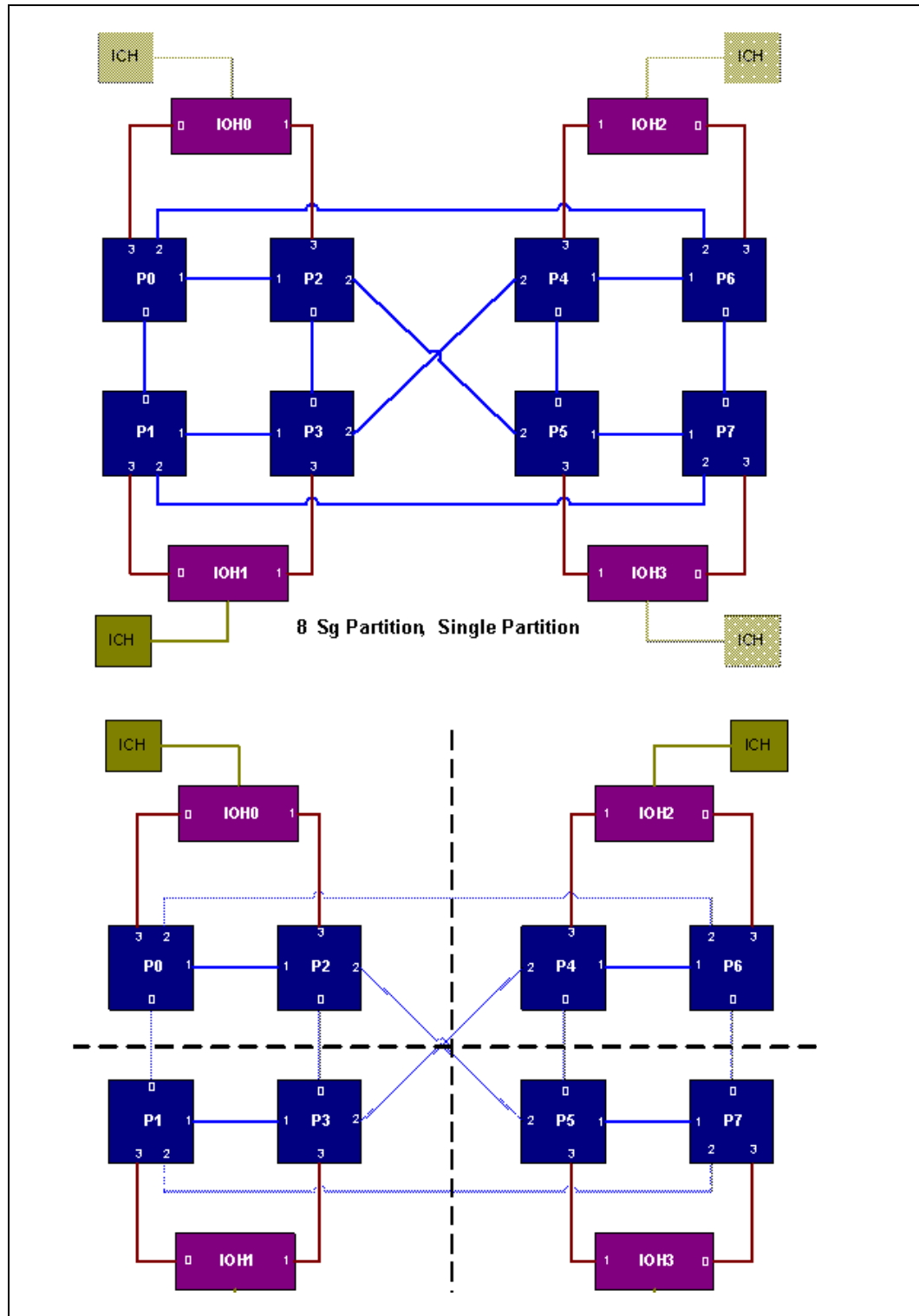


Figure 2-3 illustrates an 8-socket glueless topology.

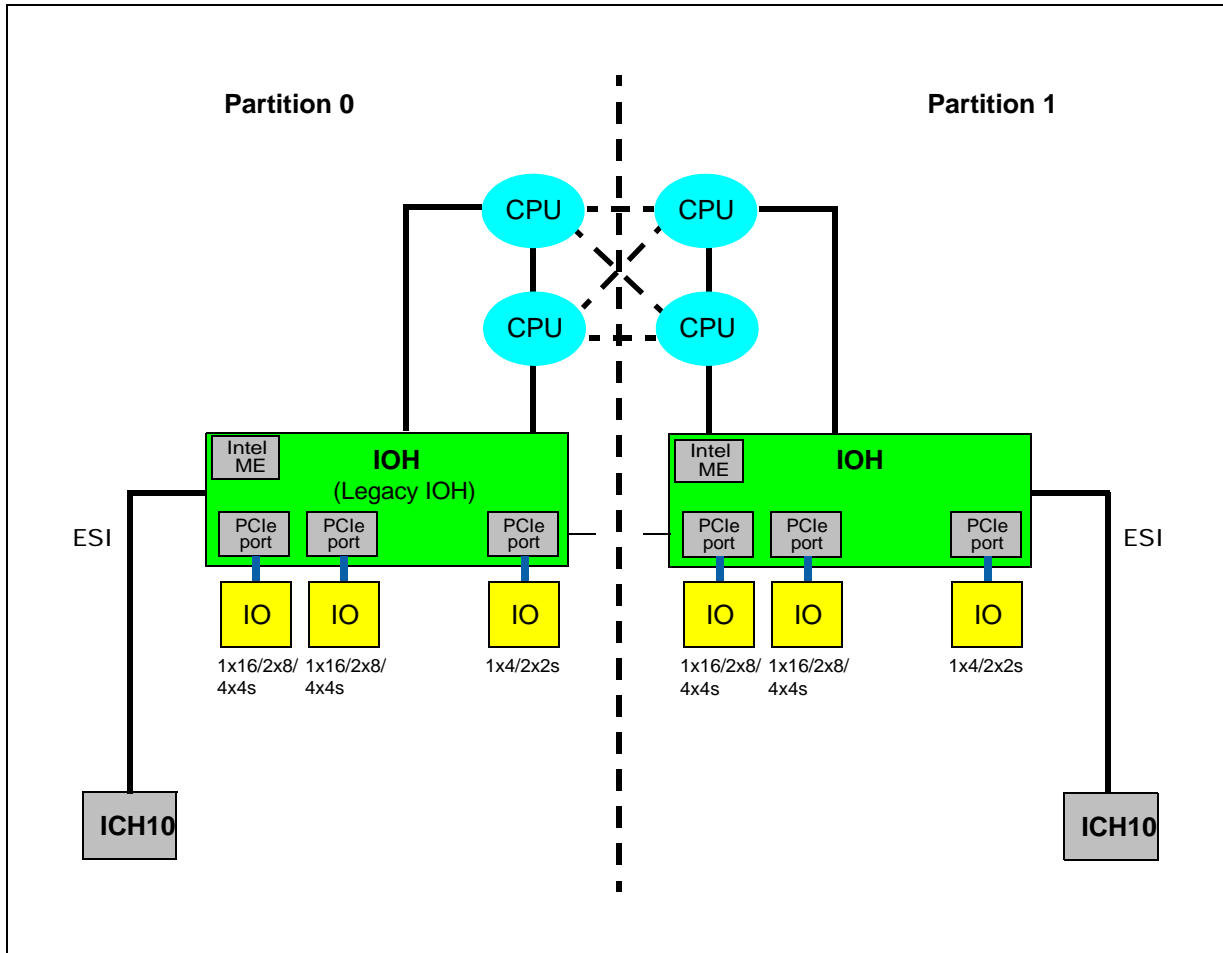
Figure 2-3. 8-Socket Glueless MP Topology



2.2.1.1 Partitioning

Figure 2-4 is an example topology of a partitioned system. For details about partitioning, please refer to the [Chapter 11, "Partitioning"](#).

Figure 2-4. Hard Partitioned MP Topology



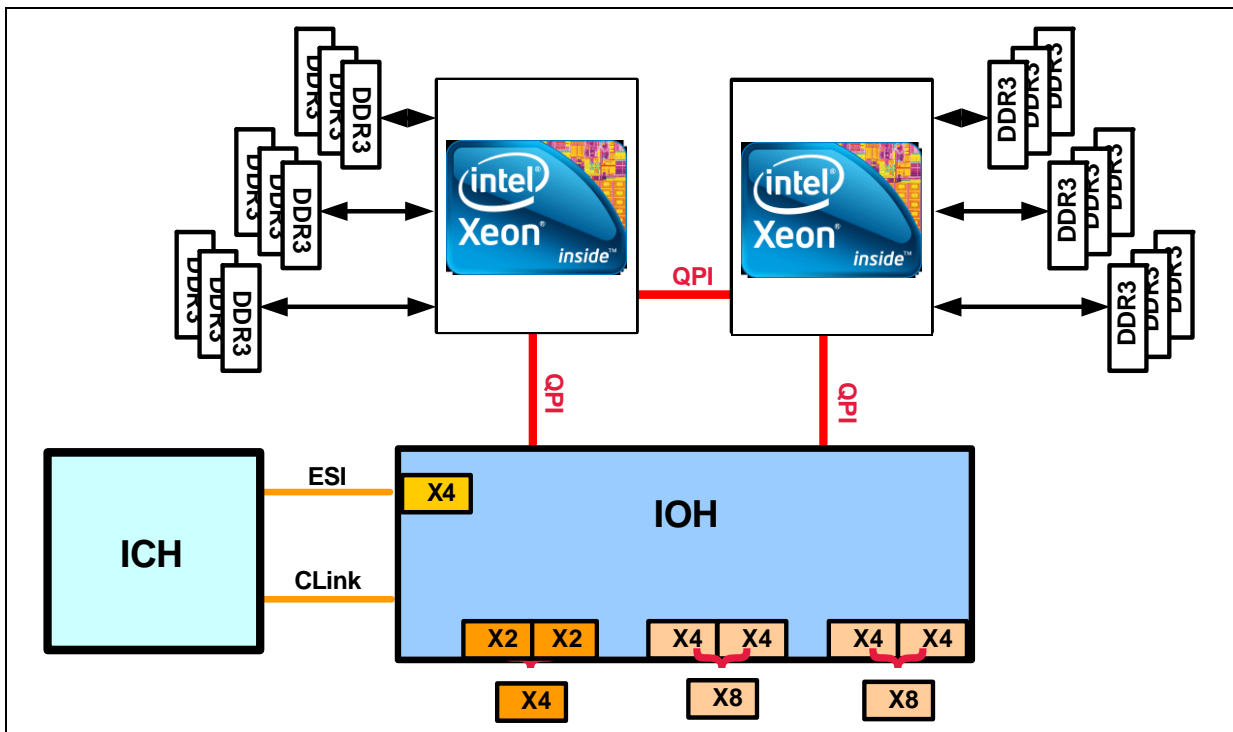
Note: The dashed lines represent Intel® QuickPath Interconnect links which are quiesced as well as delineation between the two logical partitions.

2.3 I/O Sub-System

The IOH has multiple PCI Express ports. [Figure 2-3](#) illustrates potential I/O sub-systems using Intel and other common I/O ingredients. They are meant for reference only since it is platform dependent. This example shows a typical server topology. There are other topologies with different size switches (or none at all) depending on the performance/cost/connectivity requirements. Dual IOH topologies would enable roughly twice as many PCIe lanes, but add cost and have I/O bandwidth implications.



Figure 2-5. Example Server Topology (For Reference Only)



§





3 Interfaces

3.1 Introduction

This chapter describes the physical properties and protocols for each of the IOH's major interfaces.

3.2 Intel® QuickPath Interconnect

The Intel QuickPath Interconnect interface is a proprietary cache-coherent links-based interconnect.

The IOH supports the following Intel QuickPath Interconnect features:

- 64-byte cache lines
- CRC protection: 8-bit and 16-bit rolling CRC
- L0 power states
- Virtual Networks VNO and VNA
- SMP (small-MP) profile support
- EMP (extended-MP) profile support (Intel Itanium processor 9300 series based platforms only)
- 5-bit NodeID (max) for Intel Xeon processor 7500 series; 6-bit NodeID (max) for Intel Itanium processor 9300 series
- 46-bit Physical Addressing (max) for Intel Xeon processor 7500 series based platforms; 51-bit Physical Addressing (max) for Intel Itanium processor 9300 series based platforms

Note: L0p power state is not supported

Note: Intel QuickPath Interconnect Port Bifurcation is not supported by the IOH.

3.2.1 Physical Layer

The Intel QuickPath Interconnect Physical layer implements a high-speed differential serial signaling technology. The IOH implements the following features associated with this technology:

- Differential signaling
- Forwarded clocking
- 4.8 GT/s, 5.86 GT/s or 6.4 GT/s data rate (up to 12.8 GB/s/direction peak bandwidth per port)
 - Intel Itanium processor 9300 series supports only 4.8 GT/s
- Slow boot speed of 66.66 MT/s
- L0, L0s, and L1 power states (Intel Itanium processor 9300 series does not support L0s or L1)
- Common reference clocking (same clock generator for both sender and receiver)
- Unidirectional data path in each direction supporting full duplex operation



- Intel® Interconnect Built-In Self Test (Intel® IBIST) for high-speed testability
- Polarity and Lane Reversal
- Clock Fail-safe Mode
- All required features for SMP and EMP profiles (for Intel Itanium processor 9300 series-based platforms only)

No support is provided for any runtime determinism in the IOH.

3.2.1.1 Supported Frequencies

The frequencies used on the Intel QuickPath Interconnect will be common for all ports. Support for normal operating mode of 4.8 GT/s, 5.86 GT/s or 6.4 GT/s is provided by the Physical layer. The Intel Itanium processor 9300 series based platform supports only 4.8 GT/s. Settings for the operational frequency is done through strapping pins.

The Intel QuickPath Interconnect links will come out of reset in slow mode (66.66 MT/s) independent of the operational frequency. This is based purely on the reference clock (133 MHz) divided by four. Firmware will then program the Physical layer to the operational frequency, followed by a soft reset of the Physical layer, at which point the new frequency takes over.

Table 3-1. Intel QuickPath Interconnect Frequency Strapping Options

| OPIFreqSel[1:0] | Intel QuickPath Interconnect Operational Frequency Mode |
|-----------------|--|
| 00 | 4.8 GT/s |
| 01 | 5.86 GT/s (Intel Xeon processor 7500 series-based platform)s |
| 10 | 6.4 GT/s |
| 11 | Reserved |

3.2.1.2 Supported Widths

The IOH supports two full-width Intel QuickPath Interconnect ports. Bifurcation of one full-width port into two half-width ports is not supported.

Note: The Intel Itanium processor 9300 series-based platform also supports connecting Intel® QuickPath Interconnect ports as quarter-width ports with unused lanes unconnected.

3.2.1.3 Physical Layer Initialization

The Intel QuickPath Interconnect physical layer initialization establishes:

- Link wellness through test pattern exchange
- Negotiated width of the port for degraded mode
- Presence of an Intel QuickPath Interconnect component
- Frequency is determined via strapping pins. See [Section 3.2.1.1](#) for details.

See [Section 3.2.1.5](#) for references to configuration registers in the physical layer used in the initialization process.



3.2.1.4 Clocking

The Intel QuickPath Interconnect uses a common (plesiochronous) clock between components to remove the need for “elastic buffers”, such as those used in PCI Express for dealing with the clock frequency differential between sender and receiver. This decreases latency through the Physical layer.

A single clock signal, referred to as the “forwarded” clock, is sent in parallel with the data from every Intel QuickPath Interconnect sender. Forwarded clocking is a sideband differential clock sent from each agent. The forwarded clock is not used to directly capture the data as in classical parallel buses, but is used to cancel jitter, noise, and drift that can cause reduced margin at the receiver.

3.2.1.5 Physical Layer Registers

The IOH Intel QuickPath Interconnect register details can be found in [Chapter 21](#), “Configuration Register Space.”

3.2.2 Link Layer

The IOH Link layer supports the following features:

- Virtual Networks VNO and VNA (Adaptive)
- SNP, HOM, DRS, NDR, NCB, NCS
- 8-bit and 16-bit rolling CRC

3.2.3 Routing Layer

The Routing layer provides bypassing for each target Intel QuickPath Interconnect physical port to allow requests that target other Intel QuickPath Interconnect physical ports to bypass under normal traffic patterns.

3.2.3.1 Routing Table

The IOH uses a routing table for selecting the Intel QuickPath Interconnect port to send a request to based on the target NodeID. After reset, the routing table is defaulted to disabled. In this mode, all responses are sent on the same port on which they were received. No requests can be sent from the IOH until the routing table is initialized.

3.2.4 Protocol Layer

The protocol layer is responsible for translating requests from the core into the Intel QuickPath Interconnect domain, and for maintaining Intel QuickPath Interconnect protocol semantics. The IOH is a fully-compatible Intel QuickPath Interconnect caching agent. It is also a fully-compliant I/O proxy agent for non-coherent I/O traffic. The IOH Protocol layer supports the following features:

- Intel QuickPath Interconnect caching agent
- Intel QuickPath Interconnect firmware agent, configuration agent, and I/O proxy agent
- Source Broadcast Snooping (Intel Xeon processor 7500 series based platforms only)
- Home Node Broadcast/Directory Snooping



- Router Broadcast Snooping (Intel Xeon processor 7500 series based platforms only)
- Source address decoder compatibility with Intel Xeon processor 7500 series and Intel Itanium processor 9300 series based platforms architectures
- Lock Arbiter

3.2.4.1 Supported Transactions

This section gives an overview of all the Intel QuickPath Interconnect transactions that the IOH supports.

Transactions are broken up into four broad categories for the IOH. The direction indication, inbound and outbound, is based on system transaction flow toward main memory, not the Intel QuickPath Interconnect port direction. Inbound is defined as “transactions that IOH sends to the Intel QuickPath Interconnect”, while outbound are “transactions that IOH receives from the Intel QuickPath Interconnect.”

Inbound Coherent are transactions that require snooping of other caching agents.

Inbound Non-Coherent transactions do not snoop other agents.

Outbound Snoops are snoops from peer agents that need to check the IOH write cache.

Outbound Non-coherent transactions target the IOH as the home agent for I/O space. This also includes transactions to the lock arbiter within the IOH.

Table 3-2. Protocol Transactions Supported (Sheet 1 of 2)

| Category | Intel QuickPath Interconnect Type | Intel QuickPath Interconnect Transaction |
|----------------------------------|-----------------------------------|--|
| Inbound Coherent + Responses | Home Requests | RdCur ⁶ , RdCode, InvItoE, WbMtoI, WbIData, WbIDataPtl |
| | Snoop Requests | Snpcur ⁶ , SnpCode, SnpInvItoE |
| | Normal Response | Cmp, DataC_[I,S/F ¹], DataC_[I,S/F ¹] _{Cmp} , Gnt_Cmp |
| | Conflict Response | FrcAckCnflt, DataC_[I,S/F ¹] _{FrcAckCnflt} , Gnt_FrcAckCnflt |
| | Forward Response ² | Cmp_FwdCode, Cmp_FwdInvOwn, Cmp_FwdInvItoE |
| Inbound Non-Coherent + Responses | Request DRAM | NonSnpWr ⁵ , NonSnpWrData ⁵ , NonSnpWrDataPtl ⁵ , NonSnpRd ⁵ |
| | Request I/O | NcP2PS, NcP2PB |
| | Request Special | PrefetchHint ⁵ , IntLogical ⁵ , IntPhysical ⁵ , NcMsgB-PMReq ⁵ , NcMsgB-VLW ⁵ IntPhysical ⁶ |
| | Lock & Quiescence Flows | NcMsgS-StopReq1, NcMsgS-StopReq2, NcMsgS-StartReq1, NcMsgB-StartReq2 |
| | Response | Cmp, DataNC, CmpD, DataC_I_Cmp, DataC_I |
| Outbound Snoop | Snoop Request | SnpCode, SnpData, SnpCur ⁶ , SnpInvOwn, SnpInvItoE, SnpInvXtoI ⁶ |
| | Response to Home | RspI, RspCnflt, RspIWb, WbIData, WbIDataPtl |
| | Response to Requestor | N/A |



Table 3-2. Protocol Transactions Supported (Sheet 2 of 2)

| Category | Intel QuickPath Interconnect Type | Intel QuickPath Interconnect Transaction |
|-----------------------|-----------------------------------|--|
| Outbound Non-Coherent | Request I/O or internal IOH space | NcWr, NcWrPtl, WcWr, WcWrPtl, NcRd, NcRdPtl, NcIOWr, NcIORd, NcCfgWr, NcCfgRd, NcP2PS, NcP2PB |
| | Special Messages | IntPhysical ³ , IntLogical ^{3,5} , IntAck, NcMsgB-EOI, NcMsgS-Shutdown, NcMsgB-GPE, NcMsgB-CPEI, NcMsgB/S-<other> ⁴ , IntPrioUpd, DebugData ³ , FERR |
| | Quiescence | <Done through CSR reads and writes to control StopReq*/StartReq* flow> |
| | Lock | NcMsgS-ProcLock ⁵ , NcMsgS-ProcSplitLock ⁵ , NcMsgS-Quiesce ⁵ , NcMsgS-Unlock ⁵ , NcMsgS-StopReq1, NcMsgS-StopReq2, NcMsgS-StartReq1, NcMsgB-StartReq2 |
| | Response | Cmp, DataNC, CmpD |

Notes:

1. S-state transfers are only supported in Intel® QuickPath Interconnect for Large MP systems with no F-state support (Intel Itanium processor 9300 series based platforms only)
2. Forward Response only occurs after an AckConflit was sent
3. IOH takes no action and responds with Cmp
4. IOH takes no action and responds with CmpD
5. Intel Xeon processor 7500 series based platforms only
6. Intel Itanium processor 9300 series based platforms only

3.2.4.2 Snooping Modes

The IOH supports peer agents that are involved in coherency. The IOH contains a 32-bit or 64-bit (EMP profiles) vector to indicate peer caching agents that will specify up to 31 or 63 (EMP profiles) peer caching agents that are involved in coherency. When the IOH sends an inbound coherent request, snoops will be sent to all agents in this vector, masking the home agent. Masking of the agent is required normal behavior in the Intel QuickPath Interconnect, but a mode to disable masking is also provided in the IOH.

3.2.4.3 Broadcast Support

The IOH supports broadcast to any 5-bit NodeID (SMP profile) or 6-bit NodeID (EMP profile).

3.2.4.4 Lock Arbiter

The Lock Arbiter is a central Intel QuickPath Interconnect system resource used for coordinating lock and quiescence flows on the Intel QuickPath Interconnect. There is a single lock arbiter in the IOH which can accommodate a maximum of eight simultaneous issuers with 31 peer NodeID targets for Intel Xeon processor 7500 series and 63 peer NodeID targets for Intel Itanium processor 9300 series-based platforms. For PHold support, the lock arbiter must be assigned to the IOH that has the legacy ICH connected. IOH will not support sending PHold on the Intel QuickPath Interconnect.

The Lock Arbiter uses two different participant lists for issuing the StopReq*/StartReq* broadcasts: one for Lock, and another for quiescence.



3.3 PCI Express* Interface

PCI Express offers a high bandwidth-to-pin interface for general-purpose adapters that interface with a wide variety of I/O devices. The *PCI Express Base Specification*, Revision 2.0 provides the details of the PCI Express protocol.

3.3.1 Gen1/Gen2 Support

The IOH supports both the PCI Express First Generation (Gen1) and the PCI Express Second Generation (Gen2) specifications. The Gen2 ports can be configured to run at Gen1 speeds; however, Gen1 ports cannot be configured to run at Gen2 speeds.

All PCI Express ports are capable of operating at both Gen1 and Gen2 speeds.

3.3.2 PCI Express* Link Characteristics - Link Training, Bifurcation, and Lane Reversal Support

3.3.2.1 Port Bifurcation

The IOH supports port bifurcation using PEWIDTH[5:0] hardware straps. [Table 19-12, "PEWIDTH\[5:0\] Strapping Options"](#) illustrates the strapping options for IO Unit (IOU) 0 to 2. The IOH supports the following configuration modes:

- The width of all links are exactly specified by the straps
- The width of all links are programmed by the BIOS using the PCIE_PRTx_BIF_CTRL register (wait on BIOS mode)

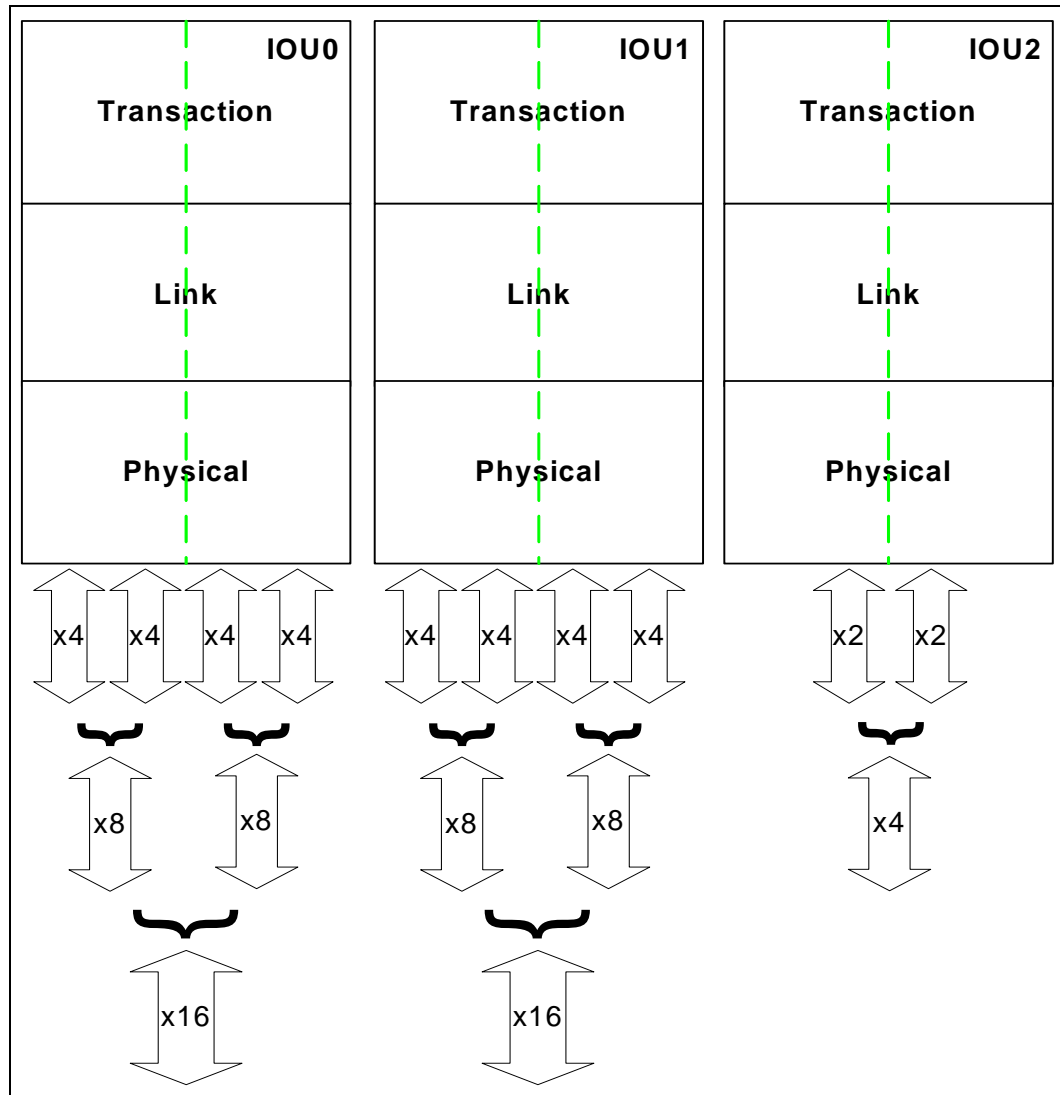
3.3.2.2 Link Training

The IOH PCI Express devices support the following link widths: x16, x8, x4, x2, x1, up to the maximum allowed for the device based on the bifurcation settings. Each device will first attempt to train at the highest possible width configured. If there is a failure to train at the maximum width, the IOH will attempt to link at progressively smaller widths until training is successful.

For full-width link configurations, lane reversal is supported. Most degraded link widths also support lane reversal, see [Table 3-3, "Supported Degraded Modes"](#).



Figure 3-1. PCI Express* Interface Partitioning



3.3.3 Degraded Mode

Degraded mode is supported for x16, x8, x4 and x2 link widths. The IOH supports degraded mode operation at half the original width and quarter of the original width or at x1. This mode allows one half or one quarter of the link to be mapped out if one or more lanes should fail during normal operation. This allows for continued system operation in the event of a lane failure. Without support for degraded mode, a failure on a critical lane such as lane 0 could bring the entire link down in a fatal manner. This can be avoided with support for degraded mode operation. For example, if lane 0 fails on a x8 link, then the lower half of the link will be disabled and the traffic will continue at half the performance on lanes 4-7. Similarly, a x4 link would degrade to a x2 link. This remapping will occur in the physical layer, and the link and transaction layers are unaware of the link width change. The degraded mode widths are automatically attempted every time the PCI Express link is trained. The events that trigger PCI Express link training are documented in the *PCI Express Base Specification, Revision 2.0*.

IOH-supported degraded modes are shown below. Table 3-3 should be read such that the various modes indicated in the different rows would be tried by IOH, but not necessarily in the order shown in the table. IOH would try a higher width degraded mode before trying any lower width degraded modes. IOH reports entry into or exit from degraded mode to software (see Chapter 21 and also records which lane failed. Software can then report the unexpected or erroneous hardware behavior to the system operator for attention, by generating a system interrupt per Chapter 16, "IOH Error Handling Summary."

Table 3-3. Supported Degraded Modes

| Original Link Width ¹ | Degraded Mode Link width and Lanes Numbers |
|----------------------------------|--|
| x16 | x8 on either lanes 7-0,0-7,15-8,8-15 |
| | x4 on either lanes 3-0,0-3,4-7,7-4,8-11,11-8,12-15,15-12 |
| | x2 on either lanes 1-0,0-1,4-5,5-4,8-9,9-8,12-13,13-12 |
| | x1 on either lanes 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15 |
| x8 | x4 on lanes 7-4,4-7,3-0,0-3 |
| | x2 on lanes 5-4,4-5, 1-0, 0-1 |
| | x1 on lanes 0,1,2,3,4,5,6,7 |
| x4 | x2 on lanes 1-0,0-1 |
| | x1 on lanes 0,1,2,3 |
| x2 | x1 on lanes 0,1 |

Note:

1. This is the native width the link is running at when degraded mode operation kicks-in.

3.3.4 Lane Reversal

The IOH supports lane reversal on all PCI Express ports, regardless of the link width (x16, x8, x4, and x2). The IOH allows a x4 or x8 card to be plugged into a x8 slot that is lane-reversed on the motherboard, and operate at the maximum link width of the card; similarly for a x4 card plugged into a lane-reversed x4 slot, and a x2 card plugged into a lane-reversed x2 slot. Note that for the purpose of this discussion, a "xN slot" refers to a CEM/SIOM slot that is capable of any width higher than or equal to xN but is electrically wired on the board for only a xN width. A x2 card can be plugged into a x8, or x4 slot and work as x2 only if lane-reversal is *not* done on the motherboard; otherwise, it would operate in x1 mode.

3.3.5 Form-Factor Support

The IOH supports Cardedge and Server I/O Module (SIOM) form-factors. Form-factor specific differences that exist for hot-plug and power management are captured in their individual sections. SIOM does not support double-wide x16 modules and supports only single-wide x8 modules.

3.3.6 IOH Performance Policies

Unless otherwise noted, the performance policies noted in this section apply to a standard PCI Express port on the IOH.

3.3.6.1 Max_Payload_Size

The IOH supports a Max_Payload_Size of 256 Bytes on PCI Express ports and 128 Bytes on ESI.



3.3.6.2 Isochronous Support and Virtual Channels

The IOH does not support isochrony.

3.3.6.3 Write Combining

The IOH does not support outbound write combining or write combining on peer-to-peer transactions. Inbound memory writes to system memory could be combined in the IOH write cache.

3.3.6.4 Relaxed Ordering

The IOH does not support relaxed ordering optimizations in the outbound direction.

3.3.6.5 Non-Coherent Transaction Support

3.3.6.5.1 Inbound

Non-coherent transactions are identified by the NoSnoop attribute in the PCI Express request header being set. For writes the NoSnoop attribute is used in conjunction with the Relaxed Ordering attribute to reduce snoops on the Intel QuickPath Interconnect interface. For inbound reads with the NoSnoop attribute set, the IOH does not perform snoops on the Intel QuickPath Interconnect. This optimization for reads and writes can be individually disabled.

3.3.6.5.2 Outbound

IOH always clears the NoSnoop attribute bit in the PCI Express header for transactions that it forwards from the processor. For peer-to-peer transactions from other PCI Express ports and ESI, the NoSnoop attribute is passed as-is from the originating port.

3.3.6.6 Completion Policy

The *PCI Express Base Specification*, Revision 2.0 requires that completions for a specific request must occur in linearly-increasing address order. However, completions for different requests are allowed to complete in any order.

Adhering to this rule, the IOH sends completions on the PCI Express interface in the order received from the Intel QuickPath Interconnect interface and never artificially delays completions received from the Intel QuickPath Interconnect to PCI Express. The IOH always attempts to send completions within a stream in address-order on PCI Express, however, it will *not* artificially hold back completions that can be sent on PCI Express to achieve this in-orderness.

3.3.6.6.1 Read Completion Combining

The *PCI Express Base Specification*, Revision 2.0 allows that a single request can be satisfied with multiple "sub-completions" as long as they return in linearly-increasing address order. The IOH must split requests into cache line quantities before issue on the Intel QuickPath Interconnect, and, therefore will often complete a large request in cache line-sized sub-completions.

As a performance optimization, the IOH implements an opportunistic read completion combining algorithm for all reads towards main memory. When the downstream PCI Express interface is busy with another transaction, and multiple cache lines have returned before completion on PCI Express is possible, the PCI Express interface will combine the cache line sub-completions into larger quantities up to MAX_PAYLOAD.

3.3.6.7 PCI Express* Port Arbitration

The IOH provides a weighted round robin scheme for arbitration between the PCI Express ports for both main memory and peer-to-peer accesses, combined. Each PCI Express/ESI port is assigned a weight based on its width and speed.

3.3.6.8 Read Prefetching Policies

The IOH does not perform read prefetching for downstream PCI Express components. The PCI Express component is solely responsible for its own prefetch algorithms as it is best suited to make appropriate trade-offs.

The IOH also does not perform outbound read prefetching.

3.3.6.9 Direct Cache Access

IOH supports Direct Cache Access (DCA) from PCI Express via the standard memory write transaction. The RequesterID in an incoming memory write packet towards coherent DRAM space (note that NoSnoop bit being set in a PCI Express write packet does not necessarily mean that the write is towards non-coherent DRAM space) is compared against the list of Requesters that are DCA-capable. If there is a match and if DCA is enabled for the PCI Express ports, the tag field provides the destination NodeID information for DCA. DCA writes that are received towards non-coherent address regions (for example, peer-to-peer region, Intel QuickPath Interconnect non-coherent address space) are treated as errors. Also, the RequesterID authentication for DCA can be overridden using bit 40 in the PERFCTRL register.

There are eight requesterID that are valid DCA targets. When VT2 is enabled, it is up to the software to guarantee a valid NodeID for DCA. IOH will not do anything special to handle DCA in VT2 mode.

Note: DCA is only supported on IA-32 architecture processors.

3.3.7 Address Translation Caching (ATC)

IOH allows caching of DMA translations in PCI Express Endpoints. The purpose of having an Address Translation Cache (ATC) in an Endpoint is to minimize time-critical latencies and to provide a way of mitigating the impact on the RC of a Device that does high-bandwidth, widely-scattered DMA. The IOH will support the following ATC requirements:

- Send a translation in response to a translation request from an endpoint
- Issue translation invalidations to endpoint caches
- Identify whether the endpoint has completed an invalidation

3.3.8 PCI Express* RAS

The IOH supports the PCI Express Advanced Error Reporting (AER) capability. Refer to *PCI Express Base Specification*, Revision 2.0 for details.

Additionally, the IOH supports:

- PCI Express data poisoning mechanism. This feature can be optionally turned off, in which case the IOH will drop the packet and all subsequent packets.
- The PCI Express completion time-out mechanism for non-posted requests to PCI Express.



- The new role-based error reporting mechanism. Refer to [Chapter 5, “PCI Express* and ESI Interfaces”](#) for details.

Note: The IOH does not support the ECRC mechanism, that is, the IOH will not generate ECRC on transmitted packets and will ignore/drop ECRC on received packets.

Refer to [Chapter 16, “Reliability, Availability, Serviceability \(RAS\)”](#) for details on PCI Express hot-plug.

3.3.9 Power Management

The IOH does not support the beacon wake method on PCI Express. IOH supports Active State Power Management (ASPM) transitions into L0s and L1 state. Additionally, the IOH supports the D0 and D3hot power management states, per PCI Express port, and also supports a wake event from these states on a PCI Express hot-plug event. In D3hot, the IOH master aborts all configuration transactions targeting the PCI Express link. Refer to [Chapter 10, “Power Management,”](#) for details of PCI Express power management support.

3.4 Enterprise South Bridge Interface (ESI)

The Enterprise South Bridge Interface (ESI) is the chip-to-chip connection between the IOH and ICH10. This high-speed interface integrates advanced priority-based servicing allowing for concurrent traffic capabilities. Base functionality is completely software transparent permitting current and legacy software to operate normally.

The IOH ESI interface supports features that are listed below in addition to the PCI Express specific messages:

- A chip-to-chip connection interface to ICH10
- 2 GB/s point-to-point bandwidth (1 GB/s each direction)
- 100 MHz reference clock
- 62-bit downstream addressing
- APIC and MSI interrupt messaging support. Will send Intel-defined “End of Interrupt” broadcast message when initiated by the processor
- Message Signaled Interrupt (MSI) messages
- SMI, SCI, and SERR error indication
- Legacy support for ISA regime protocol (PHOLD/PHOLDA) required for parallel port

3.4.1 Interface Speed and Bandwidth

Raw bit-rate on the data pins of 2.5 Gb/s, resulting in a real bandwidth per pair of 250 MB/s. In addition, the maximum theoretical realized bandwidth on the interface is 1 GB/s each direction simultaneously, for an aggregate of 2 GB/s when operating as x4 link.

3.4.2 Supported Widths

The ESI port supports x4 link width; the link width is auto-negotiated at power-on. Port bifurcation is NOT supported on ESI; the ESI port is always negotiated as a single port.

3.4.3 Performance Policies on ESI

3.4.3.1 Completion Policy

Ordering rules for the ESI port is identical to that of the PCI Express interfaces described in [Chapter 6, "Ordering."](#) However, for the case when the ICH10 sends multiple read requests with the same transaction ID, then the read completions must be returned in order. As a consequence, Read completions can be returned out of order only if they have different transaction ID's. But, as a simplification, IOH will always return all completions in original request order on ESI. This includes both peer-to-peer and memory read requests.

3.4.3.2 Prefetching Policy

ESI does not perform any speculative read prefetching for inbound or outbound reads.

3.4.4 Error Handling

The same RAS features that exist on a PCI Express port also exist on the ESI port. Refer to [Section 3.3.8](#) for details.

3.4.4.1 PHOLD Support

The IOH supports the PHOLD protocol. This protocol is used for legacy ISA devices which do not allow the possibility for being both a master and a slave device simultaneously. Example devices that use the PHOLD protocol are legacy floppy drives, and LPC bus masters.

3.5 Reduced Media Independent Interface (RMII)

The Reduced Media Independent Interface (RMII) is a standard, low pin count, low power interface.

RMII supports connection to another management entity or LAN entity. The interface is designed as a MAC type interface, not a PHY type interface.

3.6 Control Link (CLink) Interface

The control link interface is a low pin count, low power interface. This interface is used to connect the Management Engine in IOH to the ICH10. The usage model for this interface requires lower power as it remains powered during even the lower power states. Since Platform Environmental Control Interface (PECI) signals are routed through the ICH10, these signals can also pass to the management engine over the control link interface. Firmware and data stored in the SPI Flash memory connected to the ICH10 are also read over the Control Link interface.

3.7 System Management Bus (SMBus)

The IOH includes an *SMBus Specification*, Revision 2.0 compliant slave port. This SMBus slave port provides server management (SM) visibility into all configuration registers in the IOH. Like JTAG accesses, the IOH's SMBus interface is capable of both accessing IOH registers and generating inband downstream configuration cycles to other components.



SMBus operations may be split into two upper level protocols: writing information to configuration registers and reading configuration registers. This section describes the required protocol for an SMBus master to access the IOH's internal configuration registers. Refer to the *SMBus Specification*, Revision 2.0 for the specific bus protocol, timings, and waveforms.

3.7.1 SMBus Physical Layer

The IOH SMBus operates at 3.3 V and complies with the SMBus SCL frequency of 100 kHz.

3.7.2 SMBus Supported Transactions

The IOH supports six SMBus commands:

- Block Write
- Block Read
- Word Write
- Word Read
- Byte Write
- Byte Read

To support longer PCIe timeouts the SMBus master is required to poll the busy bit to know when the data in the stack contains the desired data. This applies to both reads and writes. The protocol diagrams (Table 3-7 through Figure 3-12) only shows the polling in read transactions. This is due to the length of PCIe timeouts which may be upto several seconds. The SMBus slave can then hold the bus until completion to be returned from the PCIe.

Each SMBus transaction has an 8-bit command the master sends as part of the packet to instruct the IOH on handling data transfers. The format for this command is illustrated in Table 3-4.

Table 3-4. SMBus Command Encoding

| 7 | 6 | 5 | 4 | 3:2 | 1:0 |
|-------|-----|----------|--------|---|--|
| Begin | End | MemTrans | PEC_en | Internal Command: 00 - Read DWord 01 - Write Byte 10 - Write Word 11 - Write DWord | SMBus Command: 00 - Byte 01 - Word 10 - Block 11 - Reserved. Block command is selected. |

- The *Begin* bit indicates the first transaction of the read or write sequence. The examples below illustrate when this bit should be set.
- The *End* bit indicates the last transaction of the read or write sequence. The examples below best describe when this bit should be set.
- The *MemTrans* bit indicates the configuration request is a memory mapped addressed register or a PCI addressed register (bus, device, function, offset). A logic 0 will address a PCI configuration register. A logic 1 will address a memory mapped register. When this bit is set it will enable the designation memory address type.

- The *PEC_en* bit enables the 8-bit packet error checking (PEC) generation and checking logic. For the examples below, if PEC was disabled, no PEC would be generated or checked by the slave.
- The *Internal Command* field specifies the internal command to be issued by the SMBus slave. The IOH supports dword reads and byte, word, and dword writes to configuration space.
- The *SMBus Command* field specifies the SMBus command to be issued on the bus. This field is used as an indication of the transfer length so the slave knows when to expect the PEC packet (if enabled).

The SMBus interface uses an internal register stack that is filled by the SMBus master before a request to the config master block is made. Shown in [Table 3-5](#) is a list of the bytes in the stack and their descriptions.

Table 3-5. Internal SMBus Protocol Stack

| SMBus stack | Description |
|-----------------|---|
| Command | Command byte |
| Byte Count | The number of bytes for this transaction |
| Bus Number | Bus number |
| Device/Function | Device[4:0] and Function[2:0] |
| Address High | The following fields are further defined. Reserved[3:0] Address[11:8]: This is the high order PCIe address field. |
| Register Number | Register number is the lower order 8 bit register offset |
| Data3 | Data byte 3 |
| Data2 | Data byte 2 |
| Data1 | Data byte 1 |
| Data0 | Data byte 0 |

3.7.3 Addressing

The slave address each component claims is dependent on the NODEID and SMBUSID pin straps (sampled on the assertion of PWRGOOD). The IOH claims SMBus accesses to address 11X0_XXX. The Xs represent strap pins on the IOH. Refer to [Table 3-6](#) for the mapping of strap pins to the bit positions of the slave address.

Note: The slave address is dependent on strap pins only and cannot be reprogrammed. It is possible for software to change the default NodeID by programming the QPIPC register but this will **not** affect the SMBus slave address.

Table 3-6. SMBus Slave Address Format (Sheet 1 of 2)

| Slave Address Field Bit Position | Slave Address Source |
|----------------------------------|----------------------|
| [7] | 1 |
| [6] | 1 |
| [5] | SMBUSID strap pin |
| [4] | 0 |
| [3] | 0 |


Table 3-6. SMBus Slave Address Format (Sheet 2 of 2)

| Slave Address Field Bit Position | Slave Address Source |
|----------------------------------|---|
| [2] | NODEID[3] strap pin |
| [1] | NODEID[2] strap pin |
| [0] | Read/Write# bit. This bit is in the slave address field to indicate a read or write operation. It is not part of the SMBus slave address. |

If the Mem/Cfg bit is cleared, the address field represents the standard PCI register addressing nomenclature, namely: bus, device, function and offset.

If the Mem/Cfg bit is set, the address field has a new meaning. Bits [23:0] hold a linear memory address and bits[31:24] is a byte to indicate which memory region it is. [Table 3-7](#) describes the selections available. A logic one in a bit position enables that memory region to be accessed. If the destination memory byte is zero, no action is taken (no request is sent to the configuration master).

If a memory region address field is set to a reserved space the IOH slave will perform the following:

- The transaction is not executed.
- The slave releases the SCL signal.
- The master abort error status is set.

Table 3-7. Memory Region Address Field

| Bit Field | Memory Region Address Field |
|-----------|---|
| 0Fh | LT_CSI BAR |
| 0Eh | LT_PR BAR |
| 0Dh | LT_PB BAR |
| 0Bh - 0Ch | Reserved |
| 0Ah | DMI RC BAR |
| 09h | IOAPIC memory BAR |
| 08h | VT memory BAR (Virtualization Technology) |
| 07h | CB memory BAR 7 |
| 06h | CB memory BAR 6 |
| 05h | CB memory BAR 5 |
| 04h | CB memory BAR 4 |
| 03h | CB memory BAR 3 |
| 02h | CB memory BAR 2 |
| 01h | CB memory BAR 1 |
| 00h | Crystal Beach (CB) memory BAR 0 |

3.7.4 SMBus Initiated Southbound Configuration Cycles

The platform SMBus master agent that is connected to an IOH slave SMBus agent can request a configuration transaction to a downstream PCI Express device. If the address decoder determines that the request is not intended for this IOH (that is, not the IOH's bus number), it sends the request to port with the bus address. All requests outside of this range are sent to the legacy ESI port for a master abort condition.

3.7.5 SMBus Error Handling

SMBus Error Handling features:

- Errors are reported in the status byte field.
- Errors in [Table 3-8](#) are also collected in the FERR and NERR registers.

The SMBus slave interface handles two types of errors: internal and PEC. For example, internal errors can occur when the IOH issues a configuration read on the PCI Express port and that read terminates in error. These errors manifest as a Not-Acknowledge (NACK) for the read command (*End* bit is set). If an internal error occurs during a configuration write, the final write command receives a NACK just before the stop bit. If the master receives a NACK, the entire configuration transaction should be reattempted.

If the master supports packet error checking (PEC) and the PEC_en bit in the command is set, then the PEC byte is checked in the slave interface. If the check indicates a failure, then the slave will NACK the PEC packet.

Each error bit is routed to the FERR and NERR registers for error reporting. The status field encoding is defined in the [Table 3-8](#). This field reports if an error occurred. If bits[2:0] are 000b then the transaction was successful only to the extent that the IOH is aware. In other words, a successful indication here does not necessarily mean that the transaction was completed correctly for all components in the system.

The busy bit is set whenever a transaction is accepted by the slave. This is true for reads and writes but the affects may not be observable for writes. This means that since the writes are posted and the communication link is so slow the master should never see a busy condition. A time-out is associated with the transaction in progress. When the time-out expires a time-out error status is asserted.

Table 3-8. Status Field Encoding for SMBus Reads

| Bit | Description |
|-----|---|
| 7 | Busy |
| 6:3 | Reserved |
| 2:0 | 101-111: Reserved 100: SMBus time out error. 011: Master Abort. An error that is reported by the IOH with respect to this transaction. 010: Completer Abort. An error is reported by downstream PCI Express device with respect to this transaction. 001: Memory Region encoding error. This bit is set if the memory region encoding is not orthogonal (one-hot encoding violation) 000: Successful |

3.7.6 SMBus Interface Reset

The slave interface state machine can be reset by the master in two ways:

- The master holds SCL low for 25 ms cumulative. Cumulative in this case means that all the “low time” for SCL is counted between the Start and Stop bit. If this totals 25 ms before reaching the Stop bit, the interface is reset.
- The master holds SCL continuously high for 50 μ s.

Note: Since the configuration registers are affected by the reset pin, SMBus masters will not be able to access the internal registers while the system is reset.



3.7.7 Configuration and Memory Read Protocol

Configuration and memory reads are accomplished through an SMBus write(s) and later followed by an SMBus read. The write sequence is used to initialize the Bus Number, Device, Function, and Register Number for the configuration access. The writing of this information can be accomplished through any combination of the supported SMBus write commands (Block, Word or Byte). The *Internal Command* field for each write should specify Read DWord.

After all the information is set up, the last write (*End* bit is set) initiates an internal configuration read. The slave will assert a busy bit in the status register and release the link with an acknowledge (ACK). The master SMBus will perform the transaction sequence for reading the data, however, the master must observe the status bit [7] (busy) to determine if the data is valid. Because the PCIe time-outs may be long the master may have to poll the busy bit to determine when the previous read transaction has completed.

If an error occurs then the status byte will report the results. This status field indicates abnormal termination and contains status information such as target abort, master abort, and time-outs.

Examples of configuration reads are illustrated below. All of these examples have Packet Error Code (PEC) enabled. If the master does not support PEC, then bit 4 of the command would be cleared and no PEC byte exists in the communication streams. For the definition of the diagram conventions below, refer to the *SMBus Specification*, Revision 2.0. For SMBus read transactions, the last byte of data (or the PEC byte if enabled) is NACKed by the master to indicate the end of the transaction.

3.7.7.1 SMBus Configuration and Memory Block-Size Reads

Figure 3-2. SMBus Block-Size Configuration Register Read

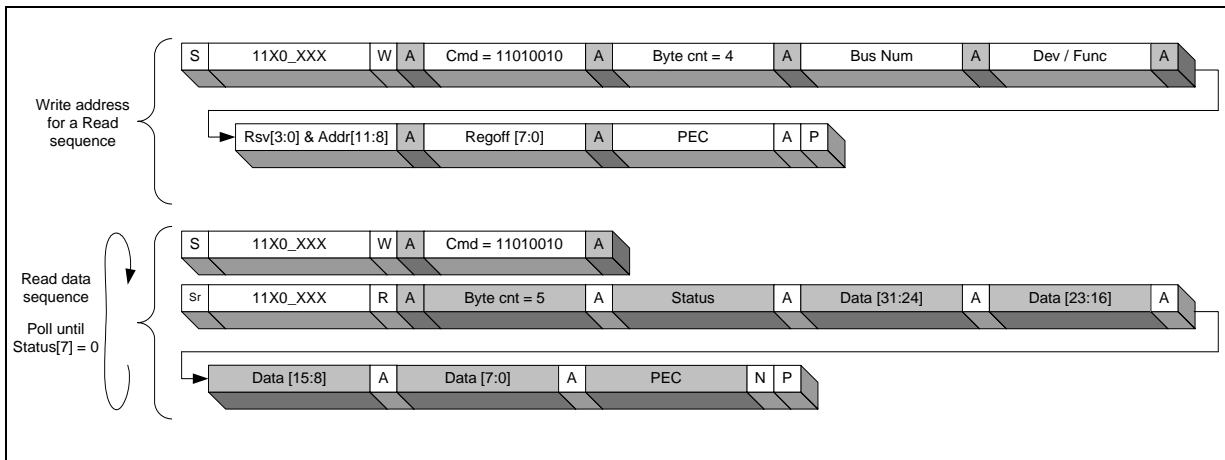
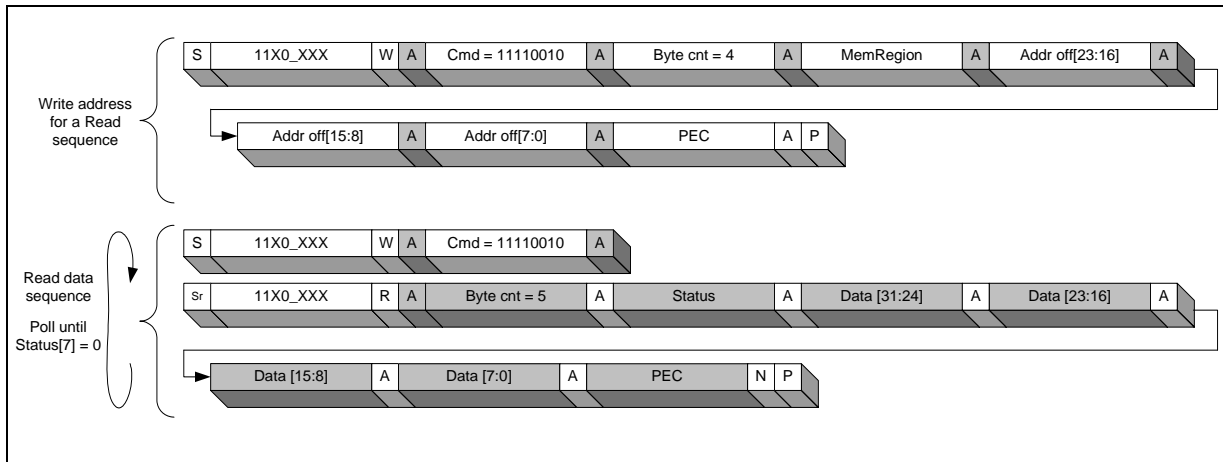
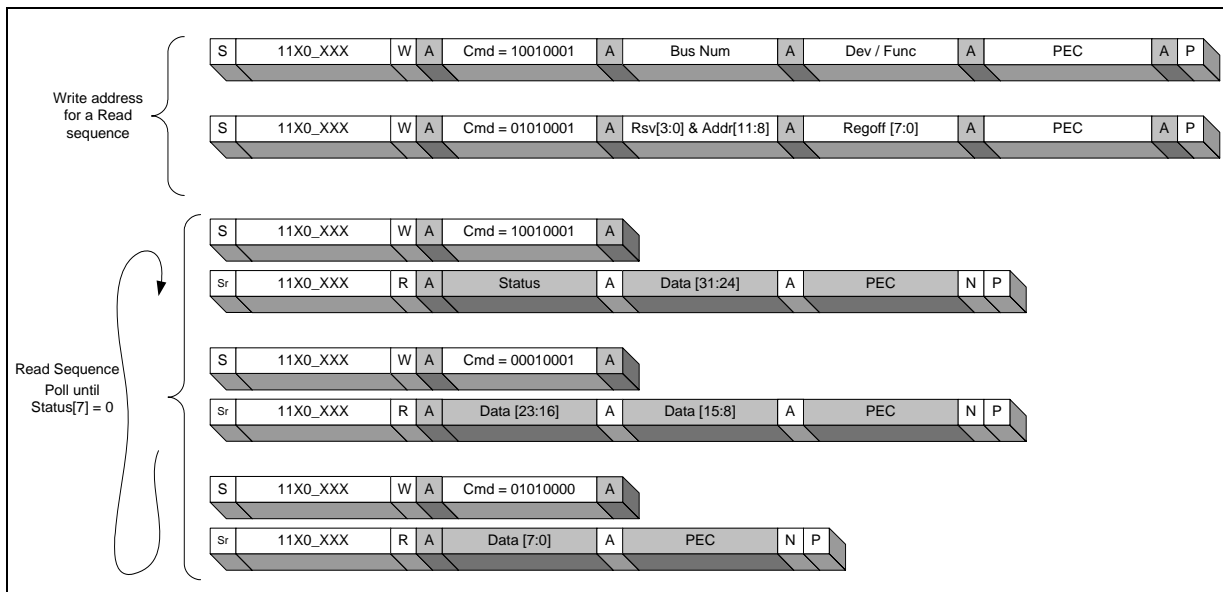


Figure 3-3. SMBus Block-Size Memory Register Read



3.7.7.2 SMBus Configuration and Memory Word-Sized Reads

Figure 3-4. SMBus Word-Size Configuration Register Read



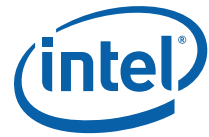
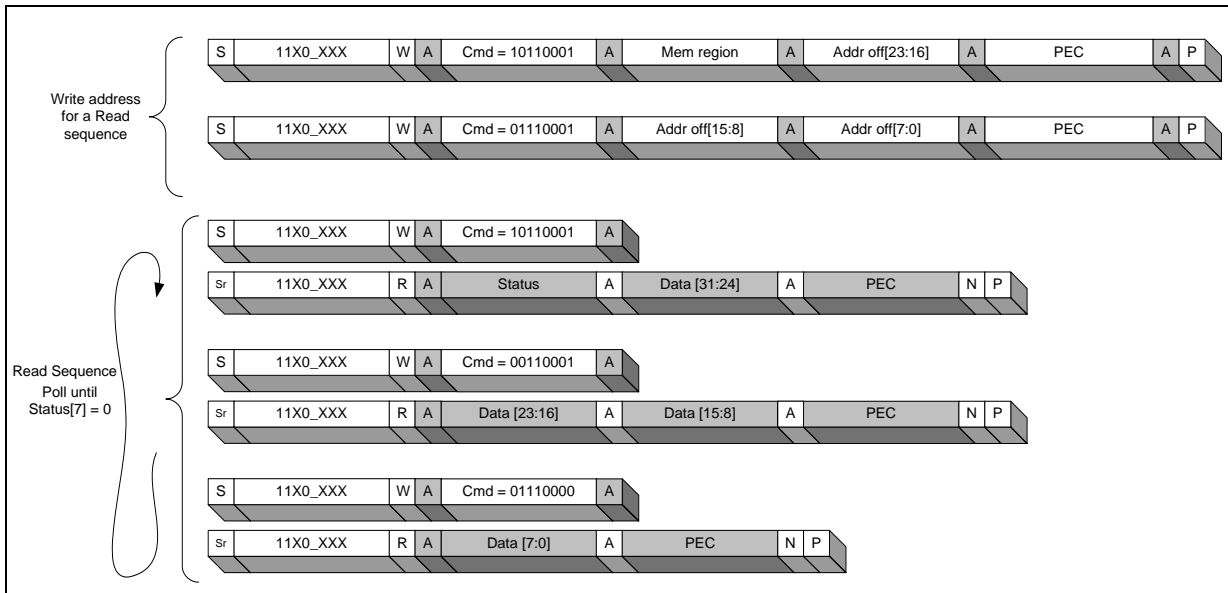


Figure 3-5. SMBus Word-Size Memory Register Read



3.7.7.3 SMBus Configuration and Memory Byte Reads

Figure 3-6. SMBus Byte-Size Configuration Register Read

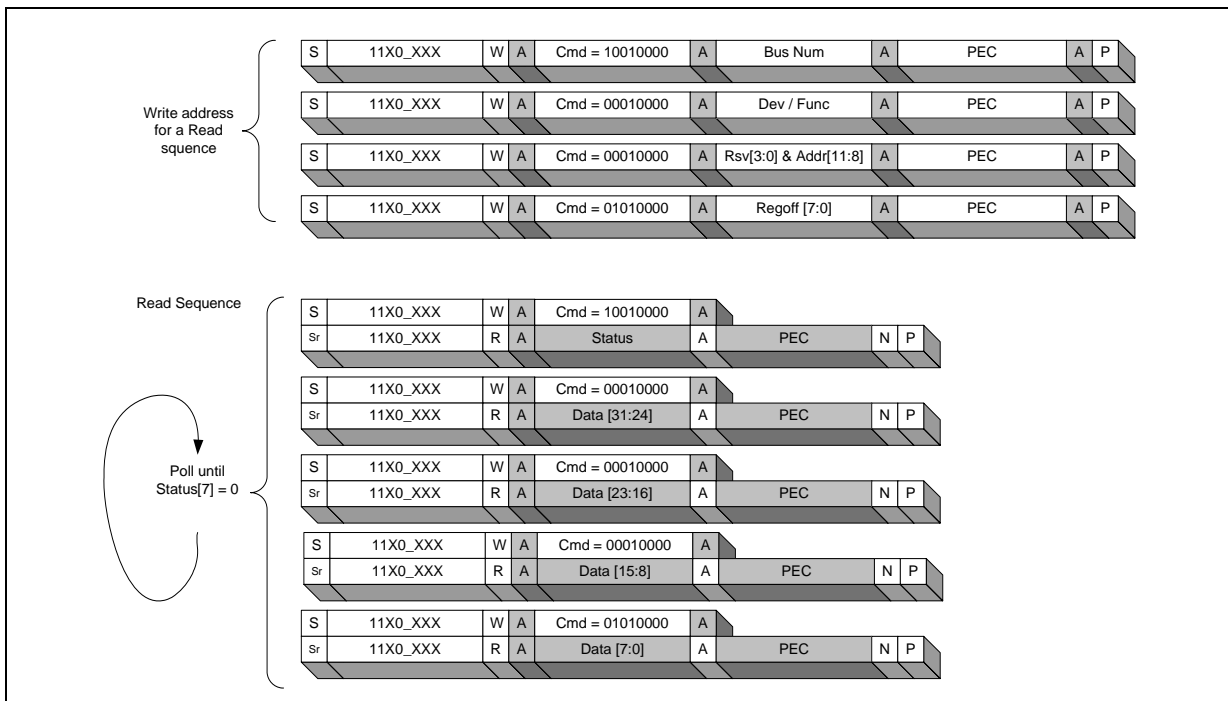
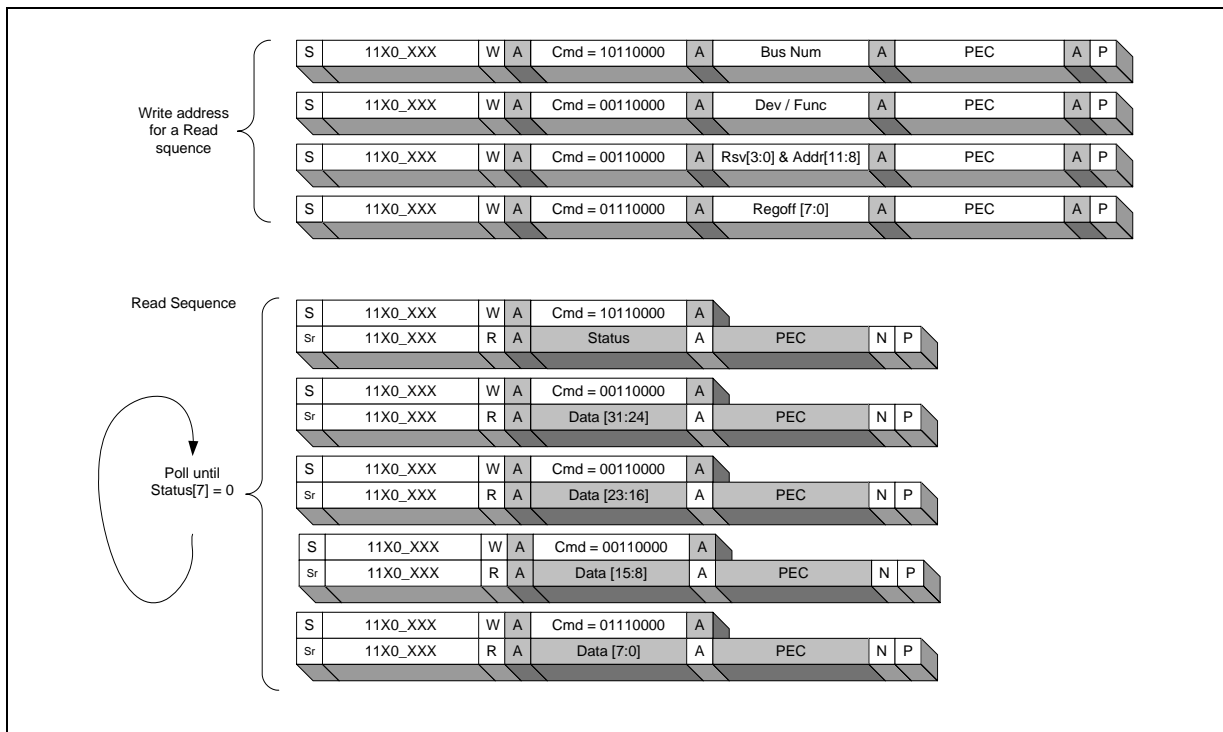


Figure 3-7. SMBus Byte-Size Memory Register Read



3.7.7.4 Configuration and Memory Write Protocol

Configuration and memory writes are accomplished through a series of SMBus writes. As with configuration reads, a write sequence is first used to initialize the Bus Number, Device, Function, and Register Number for the configuration access. The writing of this information can be accomplished through any combination of the supported SMBus write commands (Block, Word or Byte).

Note: On SMBus, there is no concept of byte enables. Therefore, the Register Number written to the slave is assumed to be aligned to the length of the Internal Command. In other words, for a Write Byte internal command, the Register Number specifies the byte address. For a Write DWORD internal command, the two least-significant bits of the Register Number or Address Offset are ignored. This is different from PCI where the byte enables are used to indicate the byte of interest.

After all the information is set up, the SMBus master initiates one or more writes which sets up the data to be written. The final write (*End* bit is set) initiates an internal configuration write. The slave interface could potentially clock stretch the last data write until the write completes without error. If an error occurs, the SMBus interface NACKs the last write operation just before the stop bit.

Examples of configuration writes are illustrated below. For the definition of the diagram conventions below, refer to the *SMBus Specification, Revision 2.0*.



3.7.7.5 SMBus Configuration and Memory Block Writes

Figure 3-8. SMBus Block-Size Configuration Register Write

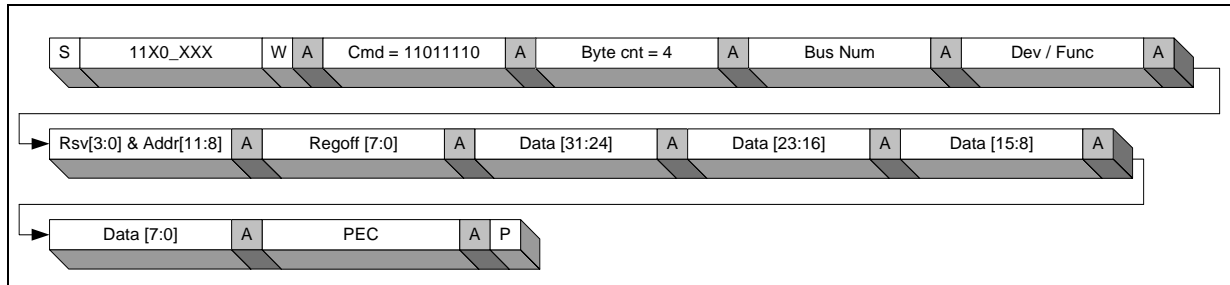
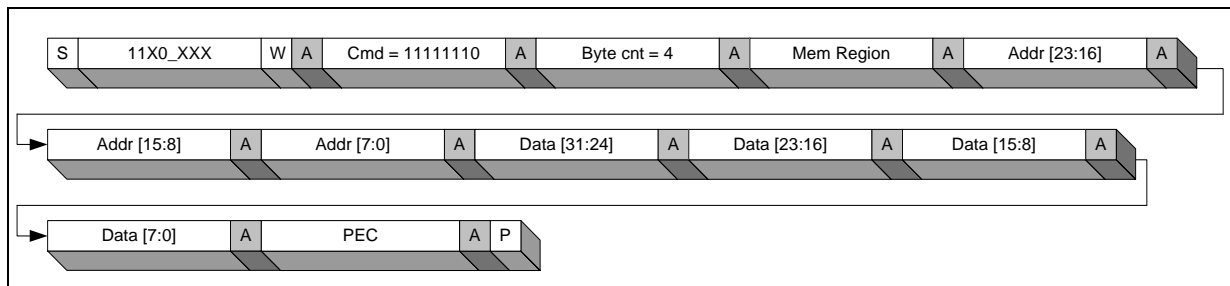


Figure 3-9. SMBus Block-Size Memory Register Write



3.7.7.6 SMBus Configuration and Memory Word Writes

Figure 3-10. SMBus Word-Size Configuration Register Write

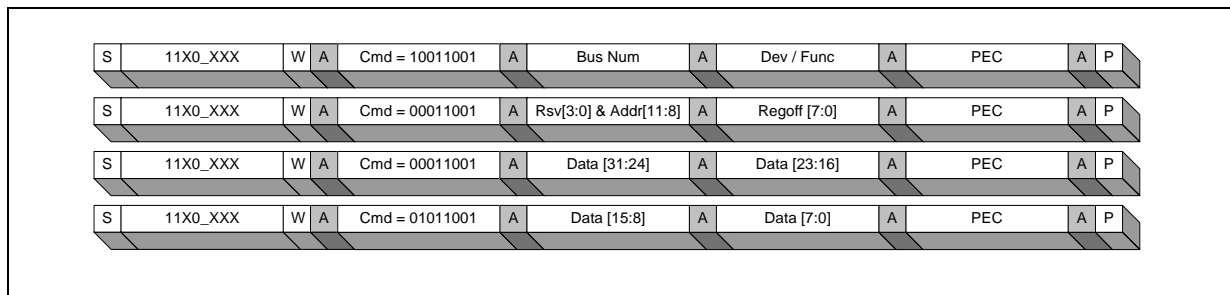
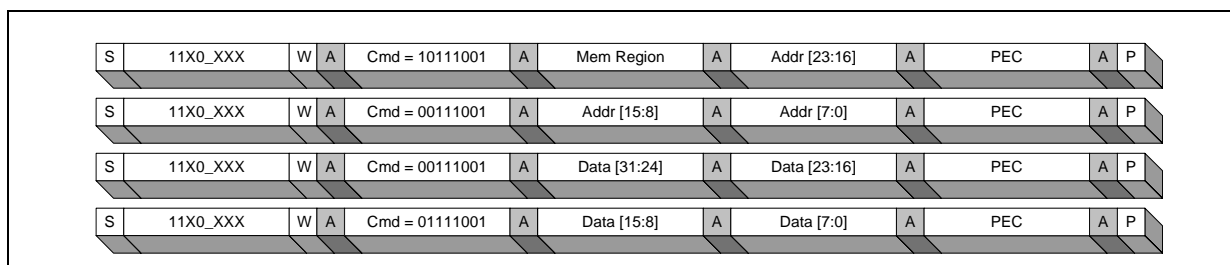


Figure 3-11. SMBus Word-Size Memory Register Write



3.7.7.7 SMBus Configuration and Memory Byte Writes

Figure 3-12. SMBus Configuration Byte Write, PEC Enabled

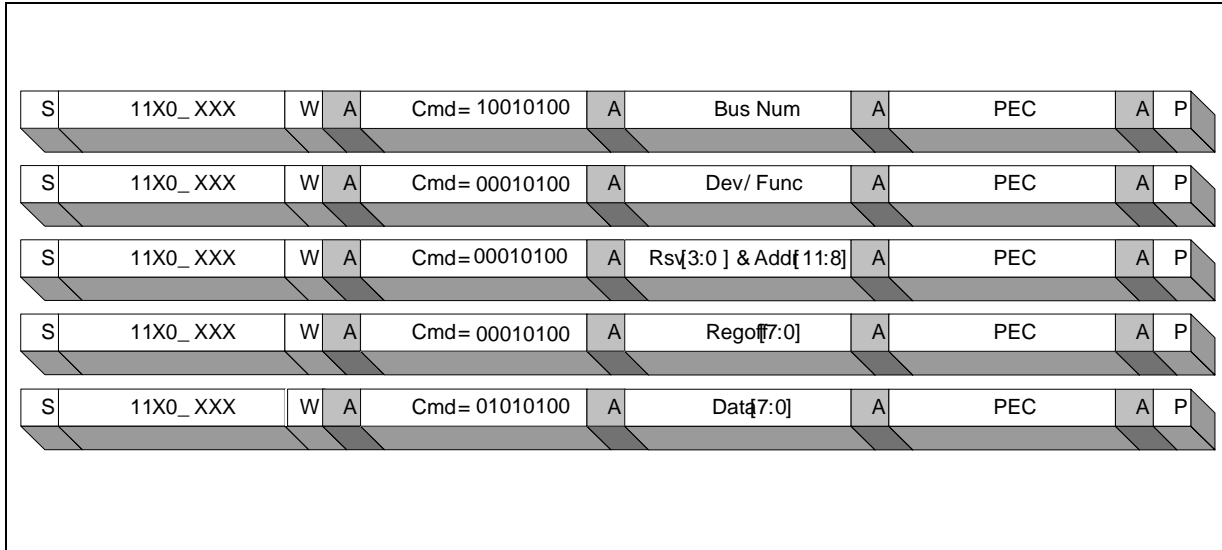
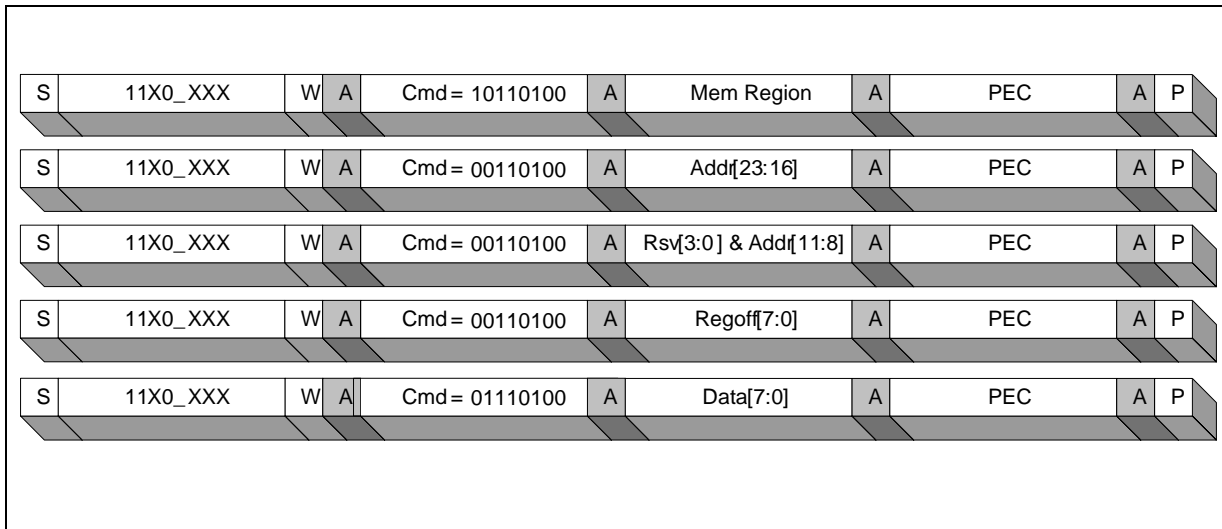


Figure 3-13. SMBus Memory Byte Write, PEC Enabled



3.8 JTAG Test Access Port Interface

The IOH supports the 1149.1 (JTAG) Test Access Port (TAP) for test and debug. The TAP interface is a serial interface comprising five signals: TDI, TDO, TMS, TCK, and TRST_N. The JTAG interface frequency limit is 1/16 of the core frequency.

3.8.1 JTAG Configuration Register Access

The IOH provides a JTAG configuration access mechanism that allows a user to access any register in the IOH and south bridge components connected to the IOH. When the specified instruction is shifted into the IOH TAP, a configuration data chain is connected between TDI and TDO. A JTAG master controller (run control tool) will shift in the



appropriate request (read request or write with data). A serial to parallel converter makes a request to the configuration master in the IOH. When the request has been serviced (a read returns data, writes are posted) the request is completed. Data resides in the JTAG buffer waiting for another JTAG shift operation to extract the data.

A general configuration chain is connected to the configuration master in the IOH that is arbitrated for access in the outbound data path address decoder. Based on the results of the decode the transaction is sent to the appropriate PCI Express port. Upon receiving a completion to the transaction the original request is retired and the busy bit is cleared. Polling can be conducted on the JTAG chain to observe the busy bit, once it is cleared the data is available for reading.

The busy bit is set whenever a transaction is accepted by the slave. This is true for reads and writes but the affects may not be observable for writes. This means that since the writes are posted and the communication link is so slow the master should never see a busy condition. A time-out is associated with the transaction in progress. When the time-out expires a time-out error status is asserted.

The region field for a memory addressed CSR access is the same as the destination memory for the SMBus described earlier. Table 3-9 shows which regions are available to JTAG for reading or writing.

If a memory region address field is set to a reserved space the JTAG port will perform the following:

- The transaction is not executed
- The master abort error status is set

Table 3-9. Memory Region Address Field

| Bit Field | Memory Region Address Selection |
|------------|--|
| All others | Reserved |
| 03h | IOAPIC memory region |
| 02h | Reserved |
| 01h | Reserved |
| 00h | Intel QuickPath Interconnect CSR memory region |

Table 3-10. JTAG Configuration Register Access (Sheet 1 of 2)

| Bit | Description | Field Definition When Transaction Type = 0 | Field Definition When Transaction Type = 1 |
|-------|--|--|--|
| 71:64 | Data Byte 3 MSB of the read or write data byte. Data[31:24] | Data3 | Data3 |
| 63:56 | Data Byte 2 MSB-1 of the read or write data byte. Data[23:16] | Data2 | Data2 |
| 55:48 | Data Byte 1 LSB+1of the read or write data byte. Data[15:8] | Data1 | Data1 |
| 47:40 | Data Byte 0 LSB of the read or write data byte. Data[7:0] | Data0 | Data0 |
| 39:32 | Register Address[7:0] Offset to a device on a bus or a memory addressed CSR. | Register[7:0] | Memory Address [7:0] |



Table 3-10. JTAG Configuration Register Access (Sheet 2 of 2)

| Bit | Description | Field Definition When Transaction Type = 0 | Field Definition When Transaction Type = 1 |
|-------|--|--|--|
| 31:29 | Function[2:0] PCI equivalent of a function number to obtain access to register banks within a device. Or this field is part of an overall memory addressed CSR. | Function | Memory Address [10:8] |
| 28:24 | Device ID[4:0] PCI equivalent to uniquely identify a device on a bus. Or this field represents a memory addressed CSR with the Region selection. | Device | Memory Address [15:11] |
| 23:16 | Bus number[7:0] PCI equivalent of a bus number to recognize devices connected to a bus. Or this field contains the high order bits for the Region selection. | Bus | Memory Address [23:16] |
| 15:12 | Extended Register Address[11:8] Extended register offset to support PCI Express configuration space. | Extended register offset [11:8] | Memory Region high [7:4] |
| 11:8 | Memory Region Low | 0 | Memory Region low [7:0] |
| 7:5 | Error Status Assertion of this bits due to an error condition resulting from an incorrect access. If the bit is logic 0 then the transaction completed successfully. 000: No error 001: Memory Region encoding error. This bit is set if the memory region encoding is not orthogonal (one-hot encoding violation). 010: Completer abort 011: Master abort 100: JTAG time-out error. Remote config transaction time out expired. 101: Reserved 110: Reserved 111: Reserved | | |
| 4 | Transaction Type Defines the type of transaction JTAG will access either config space registers or memory mapped registers. 0: Config type; use Bus/Dev.Func/Offset 1: Memory mapped type | 0 | 1 |
| 3 | Busy Bit: Set when read or write operation is in progress. | | |
| 2:0 | Command: 000: NOP. Used in polling the chain to determine if the unit is busy. 001: write byte 010: write word 011: write dword 100: read dword 101: NOP, reserved 110: NOP, reserved 111: NOP, reserved | | |

3.8.2 JTAG Initiated Southbound Configuration Cycles

The IOH allows register access to I/O components connected to the IOH PCI Express ports.



3.8.3 Error Conditions

If the configuration was targeted towards a southbound PCI Express component and the transaction returned an error the error bit is set.

§





4 Intel® QuickPath Interconnect

4.1 Introduction

Intel QuickPath Interconnect is the cache-coherent interconnect between processors and the IOH. Intel QuickPath Interconnect is a proprietary interconnect specification for links-based processor and chipset components. The IOH uses a single Intel QuickPath Interconnect NodeID.

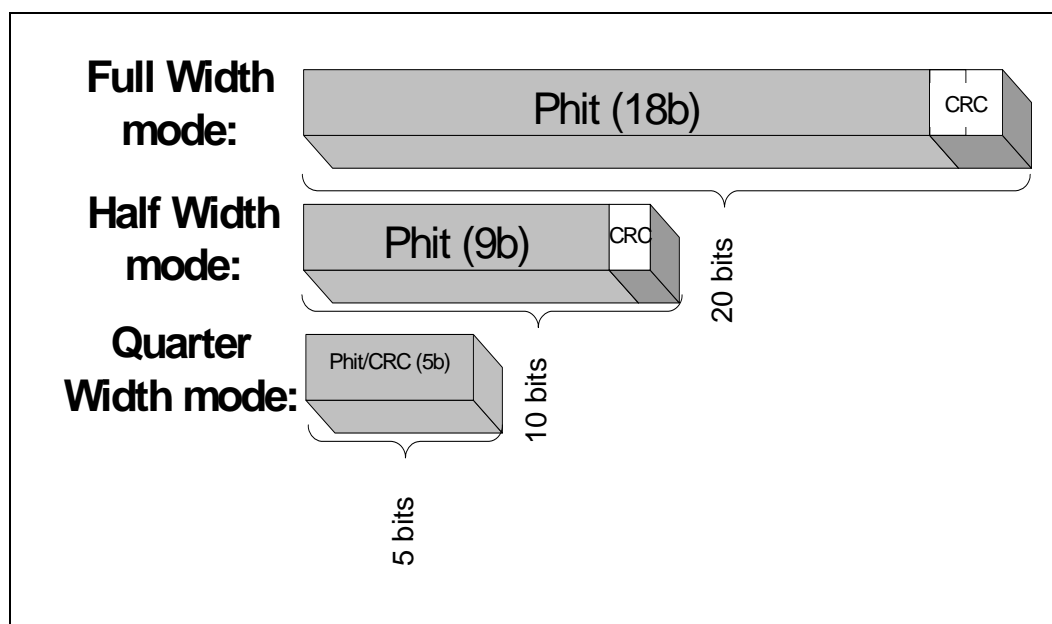
The IOH implements the Physical, Link, Routing, and Protocol layers of the Intel QuickPath Interconnect interface; however, the IOH implements only a subset of the Routing layer functionality.

Note: IOH performance requirements given in this chapter are based on a link speed of 6.4 GT/s.

4.2 Physical Layer

Figure 4-1 illustrates the scope of the physical layer on an Intel QuickPath Interconnect packet. The grayed out segment (phits) and CRC is not decoded by the Physical layer. One phit is transmitted per data clock and consists of 10 bits in half-width mode and 20 bits in full-width mode. The physical layer combines the phits into flits and passes flits to the link layer. Each flit consists of 80 bits. There are 80 bits for each flit, regardless of the port width.

Figure 4-1. Intel® QuickPath Interconnect Packet Visibility By The Physical Layer (Phit)



Note: Quarter width mode is only supported for the Intel Itanium processor 9300 series based platforms.



4.2.1 Supported Frequencies

The frequencies used on the Intel QuickPath Interconnect will be common for all ports. Support for the normal operating mode of 6.4, 5.86, or 4.8 GT/s data rate is provided by the physical layer.

4.2.2 Supported Widths

The IOH supports two full-width Intel QuickPath Interconnect ports. Each of these ports may also be connected as half-width ports with unused lanes unconnected. This allows the natural Self-Healing process to detect the active lanes. Bifurcation of one full-width ports into two half-width ports is not supported.

The IOH supports Intel® QuickPath Interconnect Dynamic Link Width Reduction which enables a degraded mode of link operation. This mode reduces a full-width port to half-width operation when transmission errors are detected. This occurs automatically in the Physical layer when reset occurs. The Physical layer runs a self test to look for bad bits. If any errors are discovered, Dynamic Link Width Reduction is invoked. The IOH allows manual selection of this mode of operation through Intel® QuickPath Interconnect configuration register settings.

The IOH supports combining quadrants 0 and 1, or, 2 and 3, to create a half-width port. A processor may support any quadrant combining when reducing to half-width. The agents on the Intel® QuickPath Interconnect bus negotiate in order to determine which quadrants are combined.

In the event of a failed clock, data bits 9 or 10, in priority order, will be used to send the clock. In this mode, the data bits in the corresponding quadrant are disabled. Bit 9 corresponds to quadrant 2, and bit 10 corresponds to quadrant 3. These bit positions apply to a full-width Intel® QuickPath Interconnect port. For a half-width Intel® QuickPath Interconnect port, the same physical pins on the IOH are used for clock failover. This implies that the half-width port will use data bit 9 for failover, and the other half-width port will use its bit 0. These pins require connection to the clock failover pins on the processor.

4.2.3 Initialization / Re-initialization

Initialization of the Physical layer can be invoked by any of the following:

- Component Reset (any type). Initialization type is always Default.
- Inband signaling (clock is no longer received). Initialization is always soft.
- Intel QuickPath Interconnect register QPIPHCTR.physical layer reset. Initialization type set by QPIPHCTR.Reset Modifier to soft or default.

Initialization will be stalled on a “Default” initialization if QPIPHCTR.PhyInitBegin is not set.

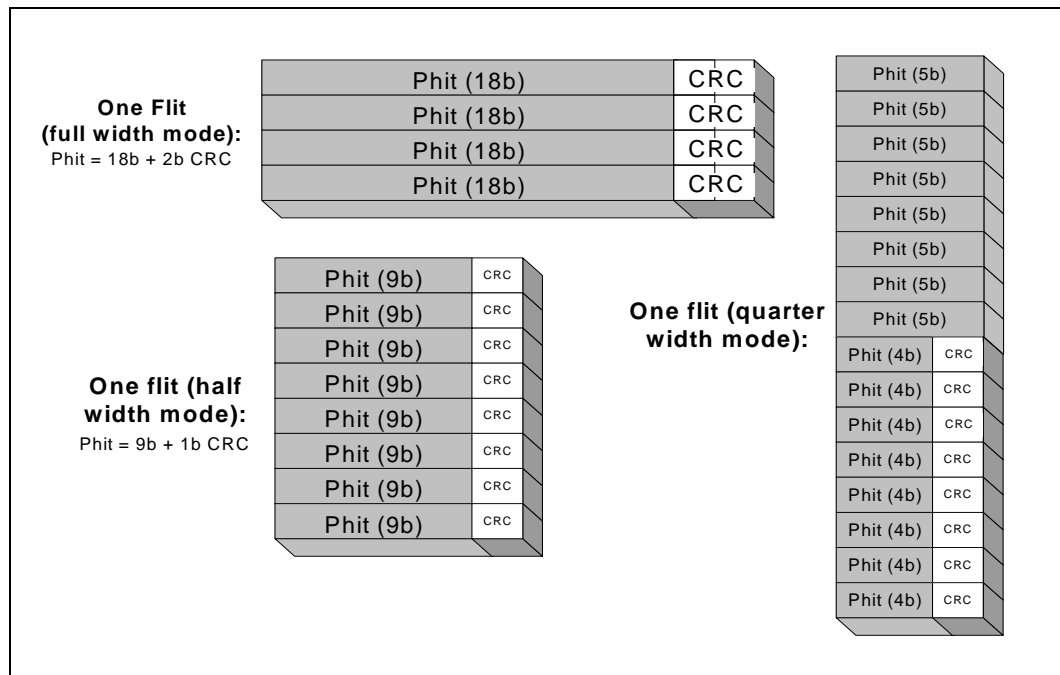
4.3 Link Layer

The Link layer provides independent flow control for each message class going to and from the Routing layer. VNA/VNO differentiation is at the Link layer of the IOH. Support is provided for the SMP and EMP (including extended header) packet formats. EMP formats are only supported by Intel Itanium processor 9300 series. See [Chapter 3](#) for supported Link Layer features.

[Figure 4-2](#) shows the flit formats supported.



Figure 4-2. Intel® QuickPath Interconnect Packet Visibility By Link Layer (Flit)



Note: Quarter width mode is only supported for the Intel Itanium processor 9300 series based platform.

4.3.1 Link Layer Initialization

During initialization, parameters are exchanged by hardware. These parameters are stored by the IOH and the information is used to setup link operation. Parameters can be accessed through the configuration registers outlined in [Chapter 21](#). Refer to [Table 4-1](#) for details on these parameter values.

Table 4-1. Link Layer Parameter Values (Sheet 1 of 2)

| Parameter | Field | Value | Notes |
|-----------|-------------|----------------------|---|
| 0 | LLR Wrap | 0x7F | Equal max to LL Retry queue size - 1. Value does not change if LL Retry queue is reduced via configuration bits. |
| | NodeID[9:3] | b'0000 & NodeID[5:3] | NodeID[5:3] value comes from "QPICTRL: Intel QuickPath Interconnect Protocol Control" register. NodeID[9:6] will always be 0x0 for IOH. |
| | #NodeID | 0 | "0" corresponds to 1 NodeID in the IOH |
| | Port# | 0 (port 0) | Corresponds to port# on the IOH. |



Table 4-1. Link Layer Parameter Values (Sheet 2 of 2)

| Parameter | Field | Value | Notes |
|-------------|---|----------|---|
| 1 | L1 Power State | 1 | Supported ¹ |
| | L0s Power State | 1 | Supported ¹ |
| | Command Insert Interleave | 0 | Not supported as a receiver or a sender |
| | Scheduled Data Interleave: Send Requested | 0 | Not supported |
| | Scheduled Data Interleave: Receiver Requested | 0 | Not supported |
| | CRC Mode: Preferred Send Modes | 01 or 10 | Depends on Mode in QPILCL register |
| | CRC Mode: Receive Modes Supported | 10 | IOH supports receiving 8b or 16b rolling CRC |
| 2 & 3 | Caching Agent | 1 | Only one section will be filled in corresponding to the NodeID definition in QPIPCTRL register. |
| | Home Agent | 0 | |
| | I/O Proxy Agent | 1 | Firmware Agent status depends on Firmware Strap. |
| | Router | 0 | |
| | Firmware Agent | 0 or 1 | |
| | Configuration Agent | 1 | |
| POC 0,1,2,3 | RSVD | 0 | Reserved |

Notes:

1. Not supported on Intel Itanium processor 9300 series

After Parameter exchange is completed, credits are exchanged via normal flow.

4.3.2 Initialization

Three conditions can start the initialization of the Link layer:

- Component reset (any type)
- A CSR configuration write to the re-init bit assigned to each Link layer defined in the Link Control register (QPILCL)
- Receipt of the parameter “ready-for-init” from the Intel QuickPath Interconnect interface

4.3.3 Packet Framing

Framing is accomplished through the “Opcode” and “Message Class” encoding in the first flit of every packet. The IOH supports both the Standard Intel QuickPath Interconnect header formats, configurable via the Intel QuickPath Interconnect Protocol Control register, “QPIPCTRL”.

4.3.4 Sending Credit Counter

The Link Layer supports a number of counters for sending requests and responses. The counters are separated based on VNA and VNO. VNO has additional separation of counters for each Message Class. The counter for VNA is based on flits, whereas the VNO counters are based on packets.

VNA will support an 8-bit counter with 0-255 flit credits, across all Message Classes.



VNO credits, in many systems, are simply available for deadlock prevention, so a minimum of 1 credit will be given. In other receivers, VNO may be the primary, or only, means of communication, so the IOH will support larger than the minimum size.

VNO Home, NDR, DRS, NCS, and NCB all support a 4-bit counter, for 0-15 packet credits in each message class.

VNO Snp message class supports a 6-bit counter for 0-63 packets. The larger counter is due to the need to support higher packet rate on this message class in source broadcast snooping protocol where each request can result in multiple snoops.

4.3.5 Retry Queue Depth

The retry queue depth is 128 flits deep to support round trip latency for a full width port from allocating the retry buffer to de-allocation. This allows for a round trip worst case delay of 512 UI (80 ns at 6.4 GT/s) round trip from retry buffer allocation to the Ack causing the retry buffer to be de-allocated.

4.3.6 Receiving Queue

The IOH has a receive queue that is 128 flits deep. This queue dynamically receives both VNA and VNO packets. The number of credits that are sent on initialization to the other side of the link is determined by configuration registers. These registers must be programmed such that the total number of flits represented cannot exceed 128, otherwise overflow of the receive queue can occur. See [Table 4-3](#) for details on Flits per Credit. For VNO, the flits per credit is always the size of the biggest packet.

Table 4-2. Credit Values Programming Example

| Virtual Network - Message Class | Flits Per Credit - Standard Header | Max Flits Per Credit - Extended Header | Notes |
|---------------------------------|------------------------------------|--|--|
| VNA | 1 | 1 | VNA is shared by all message classes |
| VNO - SNP | 1 | 2 | |
| VNO - Hom | 1 | 2 | |
| VNO - DRS | 11 | 11 | IOH does not expect to receive 11 flit NDR messages, but for simplification 11 flit should be used anyway. |
| VNO - NDR | 1 | 2 | |
| VNO - NCB | 11 | 11 | |
| VNO - NCS | 3 | 3 | |

At least one VNO credit must be given for each message class that the IOH receives. VNA should normally be set to the maximum value of the remaining flits. With Standard Headers and VNO set to one credit per Message Class (MC) this leaves 100 flit credits for VNA.

4.3.7 Link Error Protection

Error detection is done in the link layer using CRC. Two modes of CRC are supported: 16-bit rolling and 8-bit. The mode is determined as part of Link Level configuration. The 8-bit mode provides CRC protection per flit. The 16-bit rolling CRC protection transmits 8-bits of CRC in each flit which protects the current flit and previous flit. This requires the receiver to wait for one additional flit to determine if the current flit is good. This gives protection similar to that of a true 16-bit CRC.



4.3.7.1 Link Level Retry

Link level retry is supported using a circular FIFO retry queue, where every info or idle flit being sent is put into the queue. It is only removed from the queue when an acknowledgment is returned from the receiver. The acknowledgment indicates that the target Link layer received an info or idle flit error-free. If the target receives a flit with a CRC error, it returns a link level retry indication.

4.3.8 Message Class

The link layer defines eight Message Classes. The IOH supports five of those channels for receiving and six for sending. There is a restriction regarding home channel receive support as noted in the table. Table 4-3 shows the message class details.

Arbitration for sending requests between message classes uses a simple round robin between classes with available credits.

Table 4-3. Supported Intel QuickPath Interconnect Message Classes

| Message Class | VC Description | Send Support | Receive Support |
|---------------|--|--------------|------------------|
| SNP | Snoop Channel. Used for snoop commands to caching agents. | Yes | Yes |
| HOM | Home Channel. Used by coherent home nodes for requests and snoop responses to home. Channel is preallocated and guaranteed to sink all requests and responses allowed on this channel. | Yes | Yes ¹ |
| DRS | Response Channel Data. Used for responses with data and for EWB data packets to home nodes. This channel must also be guaranteed to sink at a receiver without dependence on other VC. | Yes | Yes |
| NDR | Response Channel Non-Data. | Yes | Yes |
| NCB | Non-Coherent Bypass. | Yes | Yes |
| NCS | Non-Coherent Standard. | Yes | Yes |

Notes:

1. Only supported in Route-Through modes

4.3.9 Link Level Credit Return Policy

The credit return policy requires that when a packet is removed from the Link layer receive queue, the credit for that packet/flit be returned to the sender. Credits for VNA are tracked on a flit granularity, while VNO credits are tracked on a packet granularity.

4.3.10 Ordering Requirements

The Link layer keeps each Message Class ordering independent. Credit management is kept independent on VNO. This ensures that each Message Class may bypass the other in blocking conditions.

Ordering is not assumed within a single Message Class, but is not explicitly disallowed. The Home Message Class is an exception to this rule, because it requires ordering between transactions corresponding to the same cache line address. This requirement is driven from a Protocol layer assumption on this Message Class for resolving cache line conflicts.

VNA and VNO follow similar ordering to the Message Class. With Home message class requiring ordering across VNA/VNO for the same cache line. All other message classes have no ordering requirement.



It is up to the Protocol Layer to ensure against starvation between different Message Classes.

4.4 Routing Layer

A direct routing table is supported in IOH for requests from IOH. This routing table is 32 or 64 entries deep, corresponding to the 5-bit or 6-bit NodeIDs for Intel Xeon processor 7500 series (Boxboro-EX) and Intel Itanium processor 9300 series, respectively. Alternate and Adaptive routing is not supported.

See [Chapter 16](#) for details on error logging of unexpected transactions.

4.4.1 Outbound Routing

This section discusses how packets received by the IOH are routed. There are multiple modes of routing requests received by the IOH over Intel QuickPath Interconnect. These modes exist in order to meet the needs of our different platform configurations. Selection between the routing modes is done through configuration bits. Defaults vary depending on the strapping of the IOH.

4.4.1.1 End-Point Only

End-Point Only means the IOH is always the final destination of packets. In this mode, all traffic received on Intel QuickPath Interconnect must have the Destination NodeID (DNID) equal to the IOH's NodeID, with the exception of routing snoop packets. In snoop router broadcast mode, the DNID will be set to the home NodeID and DNID checking will be disabled.

4.5 Protocol Layer

The Protocol Layer is responsible for translating requests from the core into the Intel QuickPath Interconnect domain and for maintaining Intel QuickPath Interconnect protocol semantics. The IOH is a fully-compatible Intel QuickPath Interconnect caching agent. It is also a fully-compliant firmware agent, configuration agent, and I/O proxy agent for non-coherent I/O traffic. By appropriately programming the PeerAgents list, the IOH will operate in the Intel QuickPath Interconnect in-order coherent protocol with source issued snooping of up to 31 or 63 peer caching agents for either Intel Xeon processor 7500 series or Intel Itanium processor 9300 series (maximum size of local cluster using 5 or 6-bit NodeID). By configuring the PeerAgents list to be null, the IOH will operate in system configurations which require home issued snooping, with no inherent limitation to the number of peer caching agents in this mode. The limitation is the source address decoder's ability to target the home agents.

Lock arbiter support in IA-32 systems is provided for up to eight CPU lock requesters. Systems that require support for more lock requesters need to implement a separate lock arbiter agent. Only the legacy IOH will be source of a lock as a result of PHold on ESI.

The Protocol layer supports 64 byte cache lines. There is an assumption in this section that all transactions from PCI Express will be broken up into 64-byte aligned requests to match Intel QuickPath Interconnect packet size and alignment requirements. Transactions of less than a cacheline are also supported.

4.5.1 NodeID Assignment

The IOH must have a NodeID assigned before it can begin normal operation. This NodeID will be assigned by strap pins. A maximum of 5-bits of NodeID is provided for general use in the SMP profile. In the EMP profile, the IOH provides capability for a 6-bit NodeID.

After Reset, the IOH will use three strapping pins (NodeID[2:0]) to assign the NodeID[4:2]. NodeID[5] and NodeID[1:0] will default to '0'. The NodeID may be overwritten by config writes from SMBus.

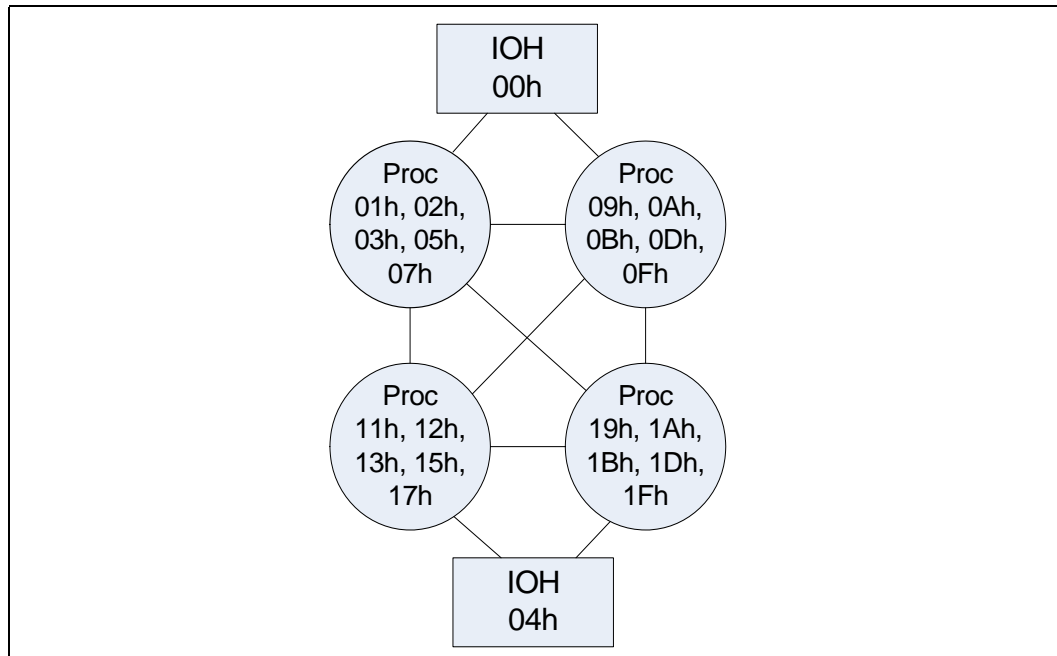
See [Chapter 21](#) for details on configuration for NodeIDs.

4.5.1.1 Component NodeID Assignment

The Intel Xeon processor 7500 series requires that IOH NodeID bits [1:0] = "00." The processor will use the other three combinations of bits [1:0] for its internal 3 NodeID assignments.

The Intel Itanium processor 9300 series requires that IOH NodeID bits [2:0] = "000" or "100" or "110". The processor will use the other three combinations of bits [1:0] for its internal 3 NodeID assignments. [Figure 4-3](#) shows an example of this type of assignment. The Intel Itanium processor 9300 series is using 5 NodeIDs with NodeID[2:0] reserving the encodings 001, 010, 011, 101, and 111.

Figure 4-3. Example NodeID Assignment 4-Socket, 2-IOH Platform (For Intel Itanium Processor 9300 Series based Platform)





4.5.2 Source Address Decoder (SAD)

Every inbound request and outbound snoop response going to Intel QuickPath Interconnect must go through the source address decoder to identify the home NodeID. For inbound requests, the home NodeID is the target of the request. For snoop requests received by the IOH, the home NodeID is not included in the snoop packet, but the home NodeID is required in the snoop response.

The source address decoder is only used for decode of the DRAM address ranges and APIC targets to find the correct home NodeID. Other ranges including any protected memory holes are decoded elsewhere. See [Chapter 7](#) for details on the address map.

The source address decoder in the IOH is designed to scale up to four sockets, each having one or two Home Agent NodeIDs. Scaling beyond four sockets is possible with a limit corresponding to the 5-bit or 6-bit NodeID, but memory interleaving across more than eight Home Agent NodeIDs is not supported. Support is provided for 3 peer IOH agents, or 1 peer IOH and 2 node controllers.

The description of the source address decoder requires that some new terms be defined:

- Memory Address – Memory address range used for coherent DRAM, MMIO, CSR.
- Bus Number Routing – Each IOH will be assigned a contiguous range of Bus numbers. This is used for routing P2P Completions and P2P Messages.
- Physical Address (PA) – This is the address field seen on the Intel QuickPath Interconnect and in the IOH, a distinction from the processor core that operates on the virtual address.

There are two basic spaces that use a source address decoder: Memory Address and PCI Express Bus Number. Each space will be decoded separately based on the transaction type.

4.5.2.1 NodeID Generation

This section provides an overview of how the source address decoder generates the NodeID. There are assumed fields for each decoder entry. In the case of some special decoder ranges, the fields in the decoder may be fixed or shifted to match different address ranges, but the basic flow is similar across all ranges.

[Table 4-4](#) defines the fields used per memory source address decoder. The process for using these fields to generate a NodeID is:

1. Match Range
2. Select TargetID from TargetID List using the Interleave Select address bit(s)NodeID[4:0] is directly assigned from the TargetID for the SMP profile, and NodeID[5:0] is directly assigned from the TargetID for the EMP profile

4.5.2.2 Memory Decoder

The MMIO ranges will never use low-order interleave, which means that all targetID entries must be programmed equivalently.

Note: The memory source address decoder in IOH contains no attribute, unlike the processor SAD. All attribute decode (MMIO, memory, NC-memory) is done with coarse range decoding prior to the request reaching the Source Address Decoder. See [Chapter 7](#) for details on the coarse decode ranges.

Table 4-4. Memory Address Decoder Fields

| Field Name | Number of Bits | Description |
|-------------------|----------------|--|
| Valid | 1 | Enables the source address decoder entry |
| Interleave Select | 3 | Determines how targets are interleaved across the range. Sys_Interleave value is set globally using the "QPIPIINT: Intel QuickPath Interconnect Protocol Interleave Mask" register. Modes: 0x0 - Addr[8:6] 0x1 - Addr[8:7], Sys_Interleave 0x2 - Addr[9:8], Sys_Interleave 0x3 - Addr[8:6] XOR Addr[18:16] 0x4 - (Addr[8:7] XOR Addr[18:17]), Sys_Interleave 0x5 - Addr[8:6]: Similar to 0x0 encoding but the nodeid[1] will be overwritten by the Sys_Interleave bit (as calculated per the CSR setting), i.e. final dnodeid = dnodeid[5:2], Sys_Interleave, dnodeid[0]. 0x6 - Addr[8:6] XOR Addr[18:16]: Similar to 0x3 encoding but the nodeid[1] will be overwritten by the Sys_Interleave bit (as calculated per the CSR setting), i.e. final dnodeid = dnodeid[5:2], Sys_Interleave, dnodeid[0] >0x6 - Reserved |
| Address Base | 25 | Starting address of the Range specified as PA[50:26] (64 MB alignment) |
| Address Limit | 25 | Ending address of the Range specified as PA[50:26] (64 MB alignment) |
| TargetID List | 48 | A list of eight 6-bit TargetID values |

4.5.2.3 Interleaving Modes

There are five interleave modes, with the first 4 referencing DRAM-only configurations.

1. Low order: low 3 bits are looked up in a table

This is commonly configured for 1, 2, 4, or 8 way interleaving.

2. Low order + Low hash: Similar to low order, but the low order nodeID bit is replaced with parity(PA<19,13,10,6>)

This is conceptually a socket-level interleave, where accesses to a socket are spread across the 2 memory controllers on a socket in a hashed manner. This should be done as a maximum 4-way socket interleave + hash, resulting in an 8-way maximum interleave.

3. Mid order: low order(PA<8:6>) XOR PA<18:16> is looked up in a table

This mode is designed to spread accesses across DRAM banks.

4. Mid order + Low hash: as above -but the low order nodeID bit is replaced with parity(PA<19,13,10,6>)
5. High order: 3 MSBs of region are looked up in a table, and which bits are determined by the region size.

Only used for IO regions in IO decoder (MMIOL, PCI CFG, legacy IO, and so on).

4.5.2.4 I/O Decoder

The I/O decoder contains a number of specialized regions. [Table 4-5](#) defines the requirements of each special decoder.

MMIOL and MMIOH use standard memory decoders.



Table 4-5. I/O Decoder Entries

| Field | Type | Address Base/Range | Size | Attr | Interleave | CSR Register | Comments |
|--------------------------------|------------|--------------------|------|--------|----------------|---|--|
| VGA/CSEg | Memory | 000A_0000 | 128K | MMIO | None | “QPIPVGASAD: QPI Protocol VGA Source Address Decode” | Space can be disabled |
| LocalxAPIC | Memory | FEE0_0000 | 1M | IPI | 8 deep table | “QPIPAPICSAD: Intel QuickPath Interconnect Protocol APIC Source Address Decode” | Which bits of address select the table entry is variable |
| Special Requests Targeting ESI | Memory | N/A | N/A | Varies | None | “QPIPSUBSAD: Intel QuickPath Interconnect Protocol Subtractive Source Address Decode” | This is used for P2P requests that target a remote ESI port. Can be disabled if no remote ESI port exists. |
| Bus Number | PCIe Bus # | 0 | 256 | Bus# | 8 deep table | “QPIPBUSAD: Intel QPI Protocol Bus# Source Address Decode” | 16 or 32 bus# per TargetID |
| DCA Tag ¹ | NodeID | 0 | 64 | NodeID | Direct Mapping | “QPIPDCASAD: Intel QuickPath Interconnect Protocol DCA Source Address Decode” | Direct mapping modes for tag to NodeID. |

Note:

1. The DCA tag is only valid for Intel Xeon processor 7500 series.

4.5.2.4.1 APIC ID Decode

The APIC ID discussed in this section is based on the Intel QuickPath Interconnect packet definition for those bits.

APIC ID decode is used to determine the Target NodeID for non-broadcast interrupts. 3 bits of the APIC ID is used to select from 8 targets. Selection of the APIC ID bits is dependent on the processor. Modes exist within the APIC ID decode register to select the appropriate bits. The bits that are used are also dependent on the type of interrupt (physical or extend logical). See [Chapter 8](#) for a detailed description of the bit definitions.

In a hierarchical MP system, certain bits of the APIC ID are matched to determine whether the interrupt is targeting the local or remote cluster. If not matched, the interrupts target the specified “Remote NodeID”. Otherwise, the standard target list is used to determine the target NodeID. See [Chapter 8](#) for a detailed description of the bit definitions.

4.5.2.4.2 Bus Number

Bus Number decode is used for peer-to-peer completions and peer-to-peer Messages. It is decoded using a single decoder that is interleaved across all possible 256 PCI Express bus numbers using the same methodology as the Memory Decoder. This is done such that each entry in the target list is allocated 16 or 32 contiguous bus numbers (in 32 mode: entry 0 - Bus0-31, entry 1 - Bus32-63...). When in “16 Buses per entry” mode, there is a base Bus# of either 0 or 128. In this mode, if the Bus# does not fall within the 128 Bus# range, then the request is sent to the Remote NodeID which is assigned to the hierarchical controller. The Remote NodeID is only used in this mode.



4.5.2.4.3 Subtractive Decode

Requests that are subtractively decoded in the datapath or PCI Express cluster are sent to the legacy ESI port. When the legacy ESI port is located on a remote IOH over the Intel QuickPath Interconnect this decoder simply specifies the NodeID of the peer IOH that is targeted. If this decoder is disabled, then legacy DMI is not available over Intel QPI, and any subtractive decoded request that is received by the Intel QPI cluster will result in an error.

4.5.2.4.4 DCA Tag

DCA enabled writes will result in a PrefetchHint message on Intel QPI that will be sent to a caching agent on Intel QPI. The NodeID of the caching agent is determined by the PCI Express Tag. The IOH will support a number of modes for tag bits which correspond to NodeID bits. See the “[QPIPDCASAD: Intel QuickPath Interconnect Protocol DCA Source Address Decode](#)” register for details.

4.5.3 Special Response Status

The Intel QuickPath Interconnect includes two response types: normal and failed. Normal is the default; failed response is described below.

On receiving a failed response status the IOH will continue to process the request in the standard manner, but the failed status is forwarded with the completion. This response status should also be logged as an error as described in [Chapter 16](#).

The IOH will send the failed response to the Intel QuickPath Interconnect for some failed response types from PCI Express. See [Chapter 16](#) for error translation requirements.

4.5.4 Abort Time-out

An AbortTO response received at the IOH will cause a reset of the time-out counter in the Outgoing Request Buffer (ORB). This response could be received to any request or to an empty ORB entry. This is because the Completion of a request may bypass the AbortTO, which will result in the entry be de-allocated and may be empty or re-used by a different request by the time the AbortTO is seen at the ORB.

The AbortTO will be sent on QPI for an outbound NcCfgRd/Wr when the time pending in IOH exceeds a time threshold. The AbortTO response does not complete the transaction at the IOH or in the original sender. Because support for this response status is an optional feature on QuickPath Interconnect, there is a configuration bit to disable it (QPIPCTRL register). This register also controls the frequency of the AbortTO when it is enabled. The frequency may vary by up to +100% of this programmed value. This response will be sent repeatedly at the designated threshold until the request is completed or aborted.

The programmable values supported for AbortTO threshold are: 5uS, 327 μS, 41 mS. These values assume a 400 MHz core clock. Scaling will occur when the core clock speed deviates from this assumption.

4.5.5 Illegal Completion/Response/Request

IOH explicitly checks all transaction for compliance to the request-response [Table 4-6](#), [Table 4-7](#), [Table 4-8](#), and [Table 4-9](#) in this chapter. If an illegal response is detected it is logged the illegal packet is dropped. See [Chapter 16](#), “[Reliability, Availability, Serviceability \(RAS\)](#)” for details on error logging.



4.5.6 Inbound Coherent Transactions

The IOH will only send a subset of the coherent transactions supported by the Intel QuickPath Interconnect. This section will describe only the transactions that are considered coherent. The determination of Coherent versus Non-Coherent is made by the address decode. If a transaction is determined coherent by address decode, it may still be changed to non-coherent as a result of its PCI Express attributes (see [Table 4-6](#) for details).

Table 4-6. Inbound Coherent Transactions and Responses

| Core Request | Intel QuickPath Interconnect Transaction | Spawned Snoop ¹ | Non-Conflict Responses | Final Cache State | Conflict Response |
|------------------|--|----------------------------|---|-------------------|--|
| Coh-Rd | RdCur ^{2, 7} | Snpcur | DataC_I, Cmp, DataC_I_Cmp | I | FrcAckCnflt, DataC_I_FrcAckCnflt |
| Coh-Rd | RdCode ² | Snpcode | DataC_S, Cmp ⁷ , DataC_S_Cmp ⁷ , DataC_F, Cmp ⁶ , DataC_F_Cmp ⁶ | I ³ | FrcAckCnflt, DataC_[F ⁶ ,S ⁷]-FrcAckCnflt |
| RFO ⁴ | InvltoE ⁵ | SnpcInvltoE | Cmp, Gnt_Cmp | E | FrcAckCnflt, Gnt_FrcAckCnflt |
| EWB - Full | WbMtoI and WbIData | N/A | Cmp | I | FrcAckCnflt |
| EWB - Partial | WbMtoI and WbIDataPtl | N/A | Cmp | I | FrcAckCnflt |

Notes:

1. See [Section 4.5.8](#) for details on how snoops are sent out
2. Based on RdCur versus RdCode Mode, see [Section 4.5.8](#)
3. State immediately degrades from
4. RFO = Request For Ownership
5. Flow selection based on Write flow mode described in [Section 4.5.8](#)
6. platform only

4.5.6.1 Snooping Modes

Intel QuickPath Interconnect protocol has assumptions that require the home agent to send snoops for any agent under directory control. Second, the home agent in a processor sending snoops for the IOH can only spawn a single snoop.

When IOH is in the router broadcast mode, it sends a single snoop targeting the home agent's NodeID. The processor's router will ensure that snoops are correctly sent to all caching agents. In this mode the DNID on incoming snoop packets will be equal to the home NodeID. In this mode, checking DNID equal to the IOH's NodeID will be disabled for incoming snoops.

4.5.6.2 RdCur versus RdCode

When the IOH issues a read request to coherent space, it will not cache the data received in the completion, but it does need to have the latest coherent copy available. There are Intel QuickPath Interconnect commands that supports this behavior; RdCur directly supports this and RdCode indirectly supports it. RdCur is defined such that the requestor's final state is always I. For RdCode the requestor's state is always given as F or S, but the IOH can immediately degrade F or S to I. based platforms.

The RdCode/RdCur mode is set only at boot, and not modified during normal operation. RdCode requires differences in how the IOH responds to conflicting snoops versus the RdCur conflict requirements. See [Section 4.10](#) for details on conflict handling.



4.5.6.3 Directory Update Requirements

Every Request For Ownership (RFO) that is sent to get exclusive ownership of a line (E state) will have a corresponding EWB which will transition the directory and the modified line in the IOH to I-state. Exceptions to this rule can occur when an error is detected in the PCI Express packet that initiated the RFO request. In this case, IOH will send an EWB-partial with none of the Byte Enables set. This allows directories or snoop filters in the processors to be kept in sync with the IOH's cache.

For coherent memory reads the IOH will use RdCur which results in the cache line in the IOH in I-state and the directory in I-state. If RdCode is used then an inconsistency could occur between the directory and the IOH. The IOH will not support any special Directory Update command for this mode of operation. If RdCode is used with a directory, then additional snooping of the IOH will occur because the directory will show IOH with F/S state. The protocol allows S/F state to be dropped silently, so that coherency is not violated.

4.5.7 Inbound Non-Coherent Transactions

The IOH sends transactions as non-coherent transactions on the Intel QuickPath Interconnect as specified in [Table 4-7](#).

Table 4-7. Non-Coherent Inbound Transactions Supported

| Core Source | Core Type | Intel® QPI Transaction | Intel® QPI Completion | Targets | Notes |
|--------------------------|--|--|---------------------------|--|---|
| PCI Express, DMA | NC Read DRAM | NonSnpRd | DataC_I_Cmp, DataC_I, Cmp | DRAM Home | |
| PCI Express, DMA | NC Write DRAM (Full Line) | NonSnpWr & NonSnpWrData | Cmp | DRAM Home | |
| PCI Express, DMA | NC Write DRAM (Partial) | NonSnpWr & NonSnpWrDataPtI | Cmp | DRAM Home | |
| PCI Express | Peer-to-peer Deferred | NcP2PS | Cmp | IOH | |
| PCI Express, DMA | Peer-to-peer Posted | NcP2PB | Cmp | IOH | |
| PCI Express | Interrupt | IntPhysical, IntLogical ¹ | Cmp | Processor Interrupt Agent (sometimes Broadcast) | IA-32 requires broadcast for some interrupt modes |
| <pins> or ESI | Virtual Wire | NcMsgB-VLW ¹ | CmpD | Broadcast to all Processor agents | Broadcast using NC Broadcast list |
| ESI | Power Management | NcMsgB-PmReq ¹ | CmpD | Broadcast to all Processor agents | Sleep state power management comes from ESI port. See Chapter 10 for more details |
| Lock Arbiter (Intel QPI) | Freeze1, Freeze2, UnFreeze1, UnFreeze2 | NcMsgS-StopReq1, NcMsgS-StopReq2, NcMsgS-StartReq1, NcMsgB-StartReq2 | CmpD | Broadcast to all Processor or all IOH agents (depending on phase) ¹ Broadcast to all IOH agents ² | Lock Flow only active for IA-32 systems. See Section 4.7 for details |
| PCI Express | DCA Hint | PrefetchHint ¹ | Cmp | Processor Caching Agents | Destination NodeID based on PCI Express tag encoding. |

Notes:

1. Intel Xeon processor 7500 series-based platform (Boxboro-EX platform) only
2. Intel Itanium processor 9300 series-based platform only



4.5.7.1 Non-Coherent Broadcast

Support is provided for a non-coherent broadcast list to deal with non-coherent requests that are broadcast to multiple agents. Transaction types that use this flow:

- Broadcast Interrupts
- Power management requests
- Lock flow
- Global SMI
- Quiescence flow
- VLW (Virtual Legacy Wires) for Intel Xeon processor 7500 series based platform only

There are three non-coherent broadcast lists:

- The primary list is the “non-coherent broadcast list” which is used for power management, Broadcast Interrupts, and VLW. This list will be programmed to include all processors in the partition.
- The Lock Arbiter list of IOHs
- The Lock Arbiter list of CPUs

The broadcast lists are implemented with a 32-bit vector (or 64-bit vector for the EMP profile) corresponding to NodeIDs 0-31 (or 0-63 for EMP). Each bit in this vector corresponds to a destination NodeID receiving the broadcast.

The Transaction ID (TID) allocation scheme used by the IOH results in a unique TID for each non-coherent request that is broadcast. See [Section 4.9.2](#) for additional details on the TID allocation.

Broadcasts to the IOH’s local NodeID will only be spawned internally and do not appear on the Intel QuickPath Interconnect bus.

4.5.7.2 Lock Arbiter

StopReq1&2 and StartReq1&2 are broadcast to all agents specified in the quiescence list. Details on the lock arbiter are found in [Section 4.7](#).

4.5.7.3 Legacy Messages

Legacy messages are sent to the target NodeID based on address decoder output. See [Section 4.5.2, “Source Address Decoder \(SAD\)” on page 67](#) for more details.

VLW messages from ESI are broadcast to all processor targets specified in the NC Broadcast list.

4.5.8 Outbound Snoops

Outbound clean snoops are critical to system performance when the IOH is included as a broadcast peer (no IOH directory/snoop filter). In this case, the expectation is that a clean response (RspI) will result from the majority of snoops because of the small size of the IOH write cache. Because of this, the IOH must keep the clean snoop latency to a minimum. Snoop conditions that hit in the Write Cache or conflict with pending requests do not have the strict latency requirements.

This section does not address snoop conflict cases, see [Section 4.10.2](#) for details on conflicts.



Table 4-8. Snoops Supported and State Transitions

| Snoop | IOH Current State | IOH Next State | Response Requestor | Response to Home Node | Notes |
|-------|------------------------|----------------|--------------------|-----------------------|--|
| Snp* | M-full line | I | --- | RsplWb + WbIData | |
| Snp* | M-partial ¹ | I | --- | RsplWb + WbIDataPtl | |
| Snp* | E/I | I | -- | Rspl | IOH will always degrade to I from E state because no data exists |

Note:

1. Partial is defined as a line that does not have all bytes modified by inbound writes.

4.5.9 Outbound Non-Coherent

IOH will support a large number of outbound non-coherent transactions.

Table 4-9. Protocol Transactions Supported (Sheet 1 of 2)

| Intel QuickPath Interconnect Type | Core Target | Intel QuickPath Interconnect Transaction | Intel QuickPath Interconnect Completion | Notes |
|-----------------------------------|--|--|---|---|
| Special Interrupt Debug | Local Intel QuickPath Interconnect Cluster | NcMsgB/S- <other> | CmpD | Any NcMsg* that is not explicitly declared in this table will result in a CmpD with no action from the IOH. |
| | | IntPrioUpd | Cmp | Monitor these requests to find interrupt deliver mode. See Chapter 8, "Interrupts" for details. |
| | | IntPhysical, IntLogical | Cmp | No Action, just send a Cmp |
| | | DebugData | | |
| IntAck | ESI | IntAck | DataNc | |
| Messages | ESI | FERR | Cmp | Target is FERR pin and the completion sent after it is asserted |
| Messages | ESI | NcMsgS-Shutdown NcMsgB-GPE NcMsgB-CPEI | CmpD | Posted to ESI |
| Config | Local Config, PCI Express, ESI | NcCfgWr | Cmp | |
| | | NcCfgRd | DataNc | |
| Broadcast | Broadcast to all ESI, PCI Express | NcMsgB-EOI | CmpD | Broadcast to all active PCI Express and ESI ports. Cmp delivered after posting to all PCI Express ports. |



Table 4-9. Protocol Transactions Supported (Sheet 2 of 2)

| Intel QuickPath Interconnect Type | Core Target | Intel QuickPath Interconnect Transaction | Intel QuickPath Interconnect Completion | Notes |
|-----------------------------------|-------------------|--|---|---|
| MMIO | PCI Express, ESI | NcWrPtl, WcWrPtl | Cmp | Includes 64 Byte Enables. Needs to be broken up into PCI Express compliant sizes. Completion sent on Intel QuickPath Interconnect after all writes are PCI ordered. WcWrPtl will use identical flow to NcWrPtl. |
| | | NcWr, WcWr | Cmp | Cmp Sent after PCI ordered to PCI Express. WcWr will use identical flow to NcWr. |
| | | NcRd | DataNc | |
| | | NcRdPtl | DataNc | Sent for requests less than 64 bytes. 8 Byte Enables apply only when length = 0-8 bytes. |
| Legacy I/O | | NcIOWr | Cmp | |
| | | NcIORd | DataNc | Length is 4 bytes, but could cross 8-byte boundary. Needs to be broken up internally to meet PCI Express 4-byte boundary requirements for IORd. |
| Peer-to-Peer | | NcP2PS, NcP2PB | Cmp | See Section 4.5.9.2 for more information. |
| Lock | Lock Arbiter | NcMsgS-ProcLock, NcMsgS-ProcSplitLock, NcMsgS-Quiesce, NcMsgS-Unlock | CmpD | See Section 4.7 for details |
| | Core logic in IOH | NcMsgS-StopReq1, NcMsgS-StopReq2, NcMsgS-StopReq1, NcMsgB-StopReq2 | CmpD | |

4.5.9.1 Outbound Non-Coherent Request Table

Outbound non-coherent requests use a table to hold Intel QuickPath Interconnect state information while the request is pending to the Datapath and I/O interface clusters. The IOH table stores all NodeID and Transaction ID information need to generate the Completion on the Intel QuickPath Interconnect. The table has the following attributes:

- Reserved Entry for a request received from the NCB Virtual Channel. But no reservation is necessary for NCS. This is necessary to avoid deadlock with peer-to-peer requests.
- The depth needs to support the loaded round trip latency for posting a packet to PCI Express at the max outbound write bandwidth from the Intel QuickPath Interconnect.
- 8 entries for pending outbound non-posted.

4.5.9.2 Peer-to-Peer Across Intel QuickPath Interconnect

Intel QuickPath Interconnect translates some peer-to-peer transactions between IOHs into a special NcP2PS for non-posted or NcP2PB for posted packets on Intel QuickPath Interconnect. An exception is for NcIORd/Wr and NcCfgRd/Wr, which will use the standard transaction type.



4.6 Profile Support

The IOH can support a variety of small and large system configurations through the use of configuration registers.

Table 4-10 and Table 4-11 defines the features that are used in setting the profile and the corresponding register requirements.

Enabling extended headers requires the IOH to be connected to a BMC. After reset the BMC must set the extended header enable register before Intel QuickPath Interconnect is allowed to initialize. This capability pertains to the Intel Itanium processor 9300 series-based platform only.

The default values for these configuration registers are set to the MP system configuration, as noted in Table 4-10 and Table 4-11.

Table 4-10. Profile Control – Intel Xeon Processor 7500 Series based Platform Only

| Feature | Register | SMP Profile | Notes |
|-------------------------------|---|------------------|--|
| Source Address decoder enable | "QPIPCTRL: Intel QuickPath Interconnect Protocol Control" | enable | |
| Address bits | "QIPMADDDATA: Intel QuickPath Interconnect Protocol Memory Address Decode Data" | <=46 bits [45:0] | Can be reduced from the max to match a processor's support. |
| Address bits 45:41 | "QPIPCTRL: Intel QuickPath Interconnect Protocol Control" | enable | Must be enabled if Extended header is enabled. When disabled Addr[42:41] become RSVD_CHK, and Addr[45:43] become critical_chunk. The critical_chunk field is ignore by IOH and treated as 000. |
| NodeID width | "QPIPCTRL: Intel QuickPath Interconnect Protocol Control" | 5-bit | Other NodeID bits will be set to zero, and will be interpreted as zero when received. |
| Remote P2P | <I/O SAD> ¹ | enable | |
| Poison | "QPIPCTRL: Intel QuickPath Interconnect Protocol Control" | enable | When disabled any uncorrectable data error will be treated identically to a header parity. |
| Routing Table | "QPIRTBL : Intel QuickPath Interconnect Routing Table" | <prog> | |
| Viral support | "QPIPCTRL: Intel QuickPath Interconnect Protocol Control" | enable | Masks both sending and receiving viral. |

Notes:

1. See Table 4-5 for details on which registers are affected.

Table 4-11. Profile Control – Intel Itanium Processor 9300 Series-based Platform Only (Sheet 1 of 2)

| Feature | Register | SMP Profile | EMP Profile | Notes |
|-------------------------------|---|------------------|------------------|--|
| Source Address decoder enable | "QPIPCTRL: Intel QuickPath Interconnect Protocol Control" | enable | enable | |
| Address bits | "QIPMADDDATA: Intel QuickPath Interconnect Protocol Memory Address Decode Data" | <=46 bits [45:0] | <=51 bits [50:0] | Can be reduced from the max to match a processor's support. To get more than 46-bits in EMP you need to enable extended header mode. |
| Address bits 45:41 | "QPIPCTRL: Intel QuickPath Interconnect Protocol Control" | enable | enable | Must be enabled if Extended header is enabled. When disabled Addr[42:41] become RSVD_CHK, and Addr[45:43] become critical_chunk. The critical_chunk field is ignore by IOH and treated as 000. |



Table 4-11. Profile Control – Intel Itanium Processor 9300 Series-based Platform Only (Sheet 2 of 2)

| Feature | Register | SMP Profile | EMP Profile | Notes |
|-----------------|---|-------------|-------------|--|
| NodeID width | “QPIPCTRL: Intel QuickPath Interconnect Protocol Control” | 5-bit | 5-bit/6-bit | Other NodeID bits will be set to zero, and will be interpreted as zero when received. 6-bit NodeID is supported only with Extended headers are enabled. |
| Remote P2P | <I/O SAD> ¹ | enable | enable | |
| Extended Header | “QPIPCTRL: Intel QuickPath Interconnect Protocol Control” | disable | <prog> | Extended headers are optionally supported in EMP profile. To achieve more the 46 bits of addressing this is required. |
| Poison | “QPIPCTRL: Intel QuickPath Interconnect Protocol Control” | enable | enable | When disabled any uncorrectable data error will be treated identically to a header parity. |
| Routing Table | “QPIRTBL : Intel QuickPath Interconnect Routing Table” | <prog> | <prog> | |
| Viral support | “QPIPCTRL: Intel QuickPath Interconnect Protocol Control” | enable | enable | Masks both sending and receiving viral. |

Notes:

1. See Table 4-5 for details on which registers are affected.

4.7 Lock Arbiter

Lock Arbiter is a central system resource used for coordinating lock and quiescence flows on Intel QuickPath Interconnect. There is a single lock arbiter in the IOH which can accommodate a maximum of 8 simultaneous issues and 63 NodeID targets. IOH will not support sending PHold on Intel QuickPath Interconnect.

The requests from Intel QuickPath Interconnect that correspond to a System Lock are: NcMsgS-ProcLock, and NcMsgS-ProcSplitLock. The System Unlock message corresponds to NcMsgS-Unlock.

PHold is also supported from ESI and is queued sequentially with the System Lock requests from Intel QPI. PHold support in MP is supported, but is restricted to the ESI port connected to the legacy IOH.

VC1 traffic from DMI is allowed to proceed under ProcLock, ProcSplitLock, and PHold. The “Quiesce” flow requirement will result in blocking of VC1, which can be a result of config register controlled quiesce flow or from the Intel QPI message NcMsgS-Quiesce. This blocking of VC1 for “Quiesce” is necessary when it’s necessary to ensure no traffic is flowing on Intel QPI.

The lock arbiter can support a maximum of 8 CPU lock sources and one PHold source, where each source can only have a single lock or hold pending at a time. The queue receives a System Lock or PHold and sends it to the issue/control state-machine. The queue must send the System Lock and PHold requests in FIFO order to the state-machine. The System Lock queue will store only the basic information about the System Lock: SrcNodeID, Tag, and Lock Type (2-bit).

The issue/control state-machine is shown in Figure 4-4. Each System Lock must have a corresponding Unlock after the System Lock completion is sent. After the Unlock completion is sent, the state-machine is ready to accept the next System Lock or PHold request from the System Lock queue. In each of the states with the word “proceeding” a StartReq/StopReq message is broadcast to a set of Intel QPI participants and/or to



internal “south agent” targets within the IOH. Which participants are included is specified in [Table 4-12](#). For details on how the StartReq/StopReq messages are broadcast see [Section 4.5.7.1](#).

The lock arbiter also provides quiescence and de-quiescence of the system for debug and RAS operations via configuration registers. The registers used are referred to as Q[2:0] & DeQ[2:0] in [Figure 4-4](#). Configuration register is defined in [Chapter 21](#). When the Q[2:0] register bits are set quiescence is initiated. There are bits to allow each startreq phase to be initiated individually. The DeQ[2:0] bits allow each StartReq phase to be individually controlled. Before DeQ[2:0] are set the Q[2:0] bit should be cleared to prevent the Quiesce flow from starting up immediately as it enters idle. The same is true for DeQ[2:0] which should be cleared before Q[2:0] should be set. The state of the “lock arbiter” is used by software to identify when each phase is complete. This information is exported to the register defined in [Section 21.11.2.21](#).

Prior to quiesce software must ensure that no System Locks or PHold requests are active. System Locks are avoided by bring the processors into SMM mode. In a platform that supports PHold, firmware must first set the quiesce control bit to block PHold requests at the lock arbiter. Then poll to ensure that the Lock Arbiter is in the “Idle” state before proceed with the flow described above.

During system quiesce, no MSI is expected to be generated internally by IOH. If there is an MSI not blocked at PCIe root ports (by setting MSI ENABLE bit of MSICTRL register to 0) during system quiesce, then the hardware is going to push it out.



Figure 4-4. Lock Arbiter Issue/Control State-Machine

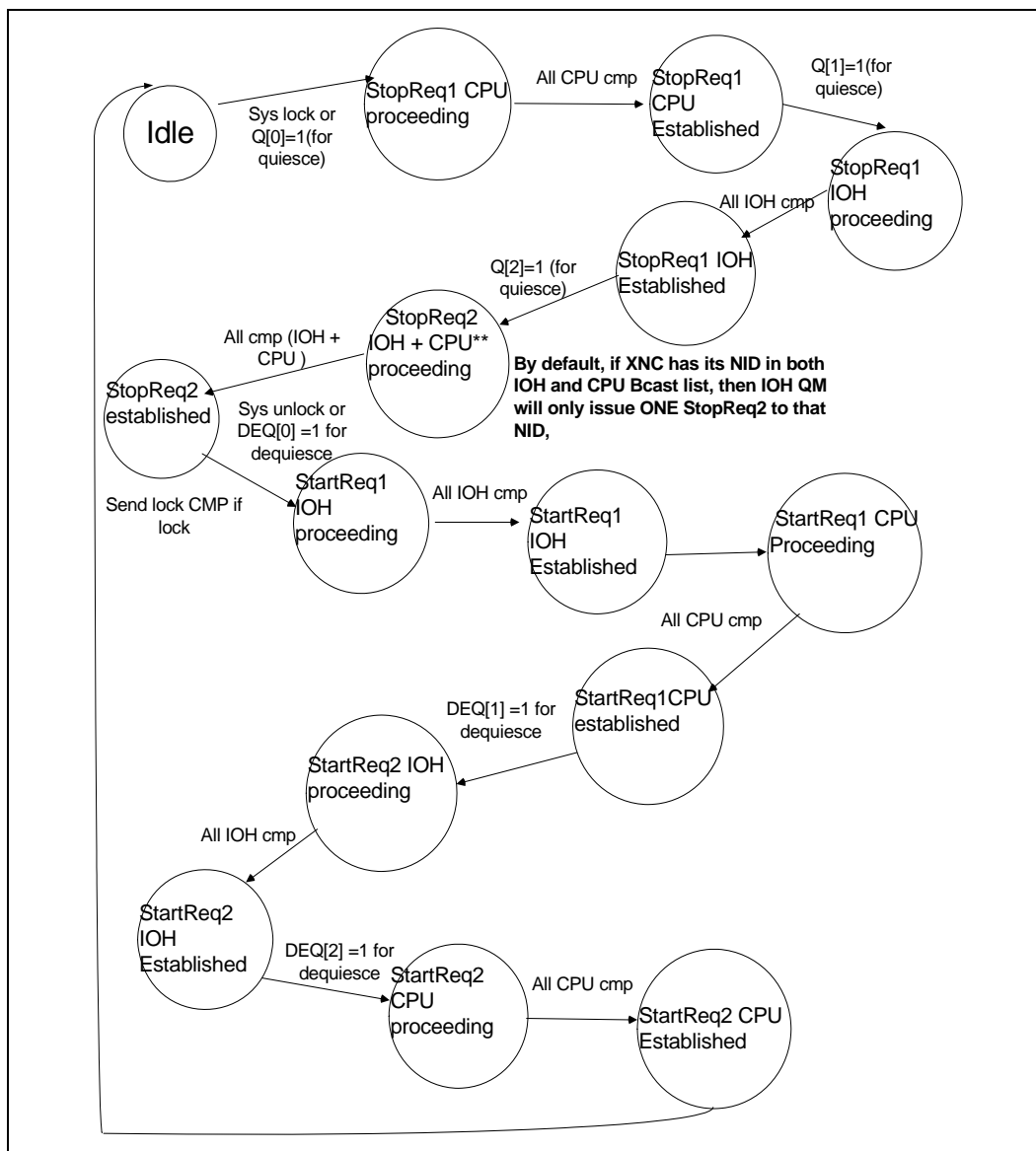


Table 4-12. Lock Master Participant List Usage

| Broadcast State | Participant List used | Register Used |
|--|-----------------------|---|
| StopReq1-CPU Proceeding | CPU Participants | "QPIQBPCPU: QPI Protocol Quiesce Broadcast CPU" |
| StopReq1-IOH Proceeding, StopReq2 Proceeding | IOH Participants | "QPIQBIOH: QPI Protocol Quiesce Broadcast IOH" |
| StartReq1 Proceeding, StartReq2-IOH Proceeding | IOH Participants | "QPIQBIOH: QPI Protocol Quiesce Broadcast IOH" |
| StartReq2-CPU Proceeding | CPU Participants | "QPIQBPCPU: QPI Protocol Quiesce Broadcast CPU" |



4.7.1 Lock Arbiter Time-Out

Requests generated to the local IOH by the lock arbiter will use the same time-out hierarchy as requests issued on Intel QuickPath Interconnect.

4.8 Write Cache

The IOH write cache is used for pipelining of inbound coherent writes. This is done by obtaining exclusive ownership of the cache line prior to ordering. Then writes are made observable (M-state) in I/O order.

4.8.1 Write Cache Depth

The write cache depth calculation is based on the latency from allocation of the RFO until the EWB causes de-allocation of the entry.

A 128-entry Write Cache meets the bandwidth requirement. This cache size assumes it is only used for inbound writes and any space for inbound read completions or read caching would be independent.

4.8.2 Coherent Write Flow

Inside the IOH, coherent writes follow a flow that starts with a RFO (Request for Ownership) followed by write a promotion to M-state.

IOH will issue an RFO command on the Intel QuickPath Interconnect when it finds the write cache in I-state. The command used for the RFO phase depends on the a configuration mode bit that selects between “Normal” or “Invalidating Write” flow. In the “Standard” flow uses the InvItoE request while the “Invalidating Write” flow uses InvWbMtoI command. These requests returns E-state with no data.

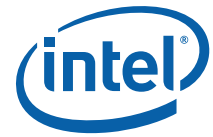
In the case where a RFO hits an M-state line in the write cache, ownership is granted immediately with no request appearing from Intel QPI. This state will be referred to as MG (M-state with RFO Granted). This state is necessary for resolving local conflicts because M-state can exist in the cache without any RFO ownership being granted. For the case of E-state in the write cache requires no additional sub-state because it always has an RFO granted.

An RFO hitting E-state or MG-state in the write cache, indicates that another write has already received RFO completion. Refer to [Section 4.10](#) for details on how this is handled.

The write promotion phase causes an M-state transition in the write cache. For partial writes, the modified bytes are tracked. In the case of MG->M state transition the data must be merged and the modified bytes being tracked are also merged. Local and remote conflict scenarios described in [Section 4.5.7](#).

In both cases IOH will hold the RTID allocation from the time the RFO is sent over Intel QPI until the EWB phase is completed. Holding of the allocation in this manner is a requirement the Invalidating Write flow because the tracker in the home is being used to track E-state in the IOH. The Standard flow will use the same RTID policy to minimize difference between the two modes.

As part of the Invalidating write flow the IOH must evict lines immediately when M-state is reached to prevent starvation in the CPU. IOH applies a single eviction policy for both modes of operation.



4.8.3 Cache State

If RFO is received by the cache and the current state is I, then either an InvItoE or InvWbMtoI occurs on the Intel QuickPath Interconnect (depending on the mode). This request always returns E-state with no data. Before this request can be issued a data resource is pre-allocated in the write cache to ensure forward progress. When the RFO request is completed then E-state has been granted for a write. Once all the I/O ordering requirements have been met, the promotion phase occurs and the state of the line becomes M.

4.8.4 System Directory Support

Under normal operation the IOH will not drop E/M cache state, and will thus have no need for system directory update. In special cases that require dropping of E state, the IOH will issue a WbMtoI + WbDataPtl with 0 byte enables. This allows inclusive snoop filter to remain in sync with the IOH cache.

For inbound reads in RdCode mode, the F/S-state data returned is immediately degraded to I-state. Any inclusive snoop filter must be aware of this policy within IOH.

4.9 Outgoing Request Buffer (ORB)

When a transaction is issued onto the Intel QuickPath Interconnect, an ORB entry is allocated. This list keeps all pertinent information about the transaction header needed to complete the request. It also stores the cacheline address for coherent transactions to allow conflict checking with snoops and other local requests. See [Section 4.10](#) for details. An Intel QuickPath Interconnect response may come as multiple phases. The ORB tracks each completion phase (that is, Data* and Cmp) and any conflicts (that is, snoops, FrcAckCnflt Response) in the ORB.

When a request is issued, a RTID (Requestor Transaction ID) is assigned based on Home NodeID. Limitations are placed on the RTID based on how many transactions each home agent can support. The RTID allocation limitation binds the tag to a specific range; this range is referred to as *MaxRequests*, where the range is 0 to *MaxRequests-1*. This value is programmable in the IOH configuration registers.

The Home NodeID and RTID are returned in the responses. The ORB must provide a way to associate the response's RTID and Home NodeID with a ORB entry. This can be done through a reverse lookup table based on RTID and Home NodeID or by searching the table for matching RTID and NodeID. Simplification of the reverse lookup function is possible by limiting the scope of the RTID allocation as described in [Section 4.9.2](#).

4.9.1 ORB Depth

The ORB depth is based on the number of requests the IOH needs to be pending on Intel QuickPath Interconnect to achieve full bandwidth, given worst-case average latency for a 4S system. This latency is calculated from allocation and de-allocation of tag. The ORB depth is 256 entries. The IOH will further partition the 256 entries into 128 entries per port. For evenly distributed traffic, this will provide the equivalent of 256 entries.



4.9.2 Requestor Transaction ID (RTID)

RTID allocation is broken into separate pools, 4 pools for each Intel QPI port (8 pools total). The size of pools, which correspond to the MaxRequest variable in Intel QPI, are programmable to 1, 2, 4, 8, 16, 24, 32. This MaxRequest value is configurable per allocation pool. The RTIDs generated from the given pool are between 0 and MaxRequests-1. Attention must be paid that RTID changes are done in one step. Programming MaxRequest RTID in multiple steps may cause a CATERR. Which pool is selected depends on the home NodeID target for the request. Configuration registers set which bits from the NodeID select the pool to use. The index setting should be chosen to maximize use of all allocation pools when traffic is distributed across all home agents. This selection depends on the NodeIDs of the agents connected to that port and the routing table map to get to that port. See [Chapter 21](#) for additional details.

4.9.3 Time-Out Counter

Each entry in the ORB is tagged with a timeout value when it is allocated; the timeout value is dependent on the transaction type. This separation allows for isolation a failing transaction when dependence exists between transactions. [Table 4-13](#) shows the timeout levels of transactions the IOH supports. Levels 2 and 6 are for transactions that the IOH does not send. The levels should be programmed such that they are increasing to allow the isolation of failing requests, and they should be programmed to consistent values across all components in the system.

The ORB implements a 2-bit timeout tag value per entry that starts out being set to 0x0. The timeout counter rate value x is programmable per time-out level. It is controllable through configuration in powers of 2. The timeout counter rate x will result in a time-out for a given transaction in that level between $3x-4x$ based on the 2-bit timeout tag. The configuration values should be programmed to increase as the level increases to support longer timeout values for the higher levels.

On each global timeout counter expiration, every ORB entry with a matching request for that level is checked. If a match is found on a valid transaction and the timeout tag is equal to 0x2, then it logged as a timeout, else the timeout tag is incremented. If timeout occurs, a failed response status is then sent to the requesting south agent for non-posted requests, and all Intel QuickPath Interconnect structures will be cleared of this request.

A request in the ORB which receives an AbortTO response, results in resetting of the timeout tag for that request. The only usage model for this case corresponds to configuration transactions to PCI Express targets coming out of reset.



Table 4-13. Time-Out Level Classification for IOH

| Level | Request Type |
|----------------|---|
| 1 | WbMtoI |
| 2 | None |
| 3 | NonSnprd, NonSnprWr, RdCur ³ , RdCode, InvltoE, NcP2PB, IntPhysical, IntLogical ² , NcMsgS-StartReq1, NcMsgB-StartReq2, PrefetchHint ² |
| 4 | NcP2PS, NcMsgB-VLW, NcMsgB-PmReq |
| 5 ¹ | NcMsgS-StopReq1, NcMsgS-StopReq2 |
| 6 | None |

Note:

1. Intel Itanium processor 9300 series will put PTC.g requests at level 5, which put the StopReq* messages at level 6. Because IOH does not support PTC.g this can be represented programming the IOH level 5 timeout adequately larger than the Intel Itanium processor 9300 series level 5.
2. Intel Xeon processor 7500 series-based platform (Boxboro-EX platform) only
3. Intel Itanium processor 9300 series-based platform only

Timeout values are specified for each level independently. The values are specified in core clocks which is proportional to Intel QuickPath Interconnect operational frequency.

4.10 Conflict Handling

A coherent conflict occurs in the Intel QuickPath Interconnect when two requests are trying to access the same cache line. This can occur when a snoop hits a pending outstanding request or ownership grant. This type of conflict is referred to as a Remote-Local Conflict. The other type of conflict is a local-local conflict, where a local read or RFO hits a pending outstanding request on the Intel QuickPath Interconnect, or a write cache entry. Local-Local conflicts also apply to non-coherent requests to DRAM (NonSnprd & NonSnprWr*).

Intel QuickPath Interconnect also has a number of rules to prevent general network deadlock that apply to all transactions. This section will refer to this class of resource deadlock.

In this section, “local requests” are requests originating from PCI Express, or ESI coming from the Intel QuickPath Interconnect interface, originating from a processor or IOH.

4.10.1 Coherent Local-Local Conflicts

Local-local conflicts occur when a local request finds state for the same cache line in the Write Cache or ORB. There are three possible outcomes of the conflict detection: stall the request until the conflicting transaction completes, eviction of the line from the write cache, or completion of the transaction.



Table 4-14. Local-Local Conflict Actions

| Local Request | ORB or Write Cache State | Action |
|-----------------------------|--|--|
| Rd or NonSnpRd or NonSnpWr* | Rd or NonSnpRd or NonSnpWr* | <ul style="list-style-type: none"> Stall until completed. |
| | RFO | <ul style="list-style-type: none"> Stall until EWB completed. Force EWB on Promotion. |
| | EWB | <ul style="list-style-type: none"> Stall until EWB completed. |
| | E- or MG-state | <ul style="list-style-type: none"> Stall until EWB completed. Force EWB on Promotion. |
| | M-state | <ul style="list-style-type: none"> Stall until EWB completed. Force EWB. |
| | Non-Coh VT-d table data | <ul style="list-style-type: none"> Data may be snarfed locally for other Intel VT-d Reads. |
| RFO | Rd or NonSnpRd or NonSnpWr* | <ul style="list-style-type: none"> Stall until completed. |
| | RFO | <ul style="list-style-type: none"> Stall until Promotion. If promotion does not evict the line then Complete RFO. If promotion causes EWB, then send RFO after EWB completed. |
| | EWB | <ul style="list-style-type: none"> Stall until EWB completed. |
| | E- or MG-state | <ul style="list-style-type: none"> Stall until Promotion. If promotion does not evict the line then Complete RFO. If promotion causes EWB, then send RFO after EWB completed. |
| | M-state | <ul style="list-style-type: none"> Complete immediately, no stall condition. |
| M-Promote | Rd or NonSnpRd or NonSnpWr* or RFO or EQB or M-state | <ul style="list-style-type: none"> Impossible. EWB can only be received in E or MG state. |
| | E- or MG-state | <ul style="list-style-type: none"> Normal flow to M-state and Eviction. If M-state does not cause eviction under normal rules then Conflict queue is checked to see if EWB required or RFO completion required. EWB: on EWB completion, next conflict is cleared. If multiple ownerships are granted simultaneously, then state will change to MG state. |

4.10.1.1 Local Conflict Bypassing

In the condition where the request is stalled, other requests are allowed to bypass from all clusters. Although conflicts are somewhat rare, a single conflict can not block traffic from other streams. The bypass buffer can absorb one stalled conflicting request per active cache line. Upon receiving a second conflict, all Read and RFO requests are blocked. Write promotion will never conflict which is guaranteed by E/M state. Promotions will not be blocked by conflicted RFO or Read request.



4.10.2 Coherent Remote-Local Conflicts

Because the IOH only supports a sub-set of coherency states and coherent transactions, it requires only a limited subset of full conflict handling.

It is assumed that conflicts occur on less than one percent of total transactions. This implies that performance of these individual transactions are of little importance, however blocking on a single conflict can not be allowed to block forward progress of other “remote requests”.

Table 4-15. Remote-Local Conflict Actions

| Snoop Request | Local Transaction Phase | Local Transaction Pending | Snoop Response |
|--|-------------------------|---------------------------|----------------|
| SnpCode, SnpData, SnpInvItoE, SnpInvXtol | Request | Rd (RdCode) RFO EWB | RspCnflt |
| SnpCode, SnpData, SnpInvItoE, SnpInvXtol | AckCnflt | Rd (RdCode) RFO EWB | <Buffer/Block> |

The AckCnflt phase is completed by a Cmp or Cmp_Fwd*. [Table 4-16](#) shows the responses and final state in the IOH on receiving different Cmp_Fwd* completions. Once the AckCnflt phase is completed, all buffered snoops are cleared.

Table 4-16. Conflict Completions Actions

| Local Transaction Pending | Conflict Completion | Response to Requestor | Send to Home | Final State | Notes |
|---|--|-----------------------|------------------|-------------|---|
| Rd (RdCur) | Cmp_Fwd* | N/A | RspI | I | RdCur results in I-state |
| Rd (RdCode) | Cmp_Fwd* | N/A | RspI | I | RdCode in IOH will always degrade to I state on RdCode Completion. |
| RFO (Write not at the head of IOQ) | Cmp_Fwd* | N/A | RspI | I | E-state will always degrade to I-state because data may not exist on E-state. |
| RFO (Write at the head of IOQ ready to atomically go M) | Cmp_FwdCode Cmp_FwdInvItoE Cmp_FwdInvOwn | N/A | RspIWb + WblData | I | Write back to home |
| | Cmp_FwdInvOwn | DataC_M | RspFwdI | I | |
| EWB | Cmp_Fwd* | N/A | RspI | N/A | The Cmp_Fwd* message could be avoided by the home agent in this case, but some home agents may not use this optimization. |



4.10.3 Resource Conflicts

Resource conflicts are generally not a problem in the Intel QuickPath Interconnect because of the independent nature of the message classes. Two cases are explicitly stated here for how resources are managed to prevent resource problems.

The first basic Intel QuickPath Interconnect rule is that completions are absorbed at the source, unconditional on any other message classes. This generally requires pre-allocation of completion resources before a request is sent. See [Section 4.11.1](#) for more details on message class ordering details.

The ORB ensures that peer-to-peer non-posted requests do not fill the ORB (per allocation pool). This ensures that posted requests (with respect to PCI Express) will never be blocked by non-posted peer-to-peer requests.

The ORB ensures that Reads are allowed fair access into the ORB.

4.11 Deadlock Avoidance

Following section calls out specific IOH ordering requirements.

4.11.1 Protocol Channel Dependence

[Section 4.11.1.1](#) through [Section 4.11.1.3](#) concentrate on potential deadlock situations between outbound and inbound traffic, and vice versa.

4.11.1.1 Outbound NC Request versus Inbound NC Request

Completions are always allowed to bypass deferred requests in the PCI ordered domain.

4.11.1.2 Inbound Response versus Inbound AckCnflt (Home Channel)

Inbound responses (responses received by the IOH) are never blocked because of blocking on the home channel in the inbound direction.

4.11.1.3 Snoop Stall on Hit, E-State

Any snoop that hits a line being promoted to M-state will be stalled while the promotion request for the M-state data is received from the IOQ to the write cache.

The write cache will revoke ownership if a snoop hits a blocked write in E-state and the RFO will be re-issued. See [Section 4.10.2](#) for more details.

§



5 PCI Express* and ESI Interfaces

5.1 Introduction

PCI Express* is the next generation I/O interface extending I/O solutions beyond PCI-X. It offers a very high bandwidth to pin interface for general-purpose adapters interfacing a wide variety of I/O devices. The *PCI Express Base Specification, Revision 2.0* provides the details of the PCI Express protocol. This chapter is complementary to [Chapter 3](#) and should be used as additional reference.

5.2 PCI Express* Link Characteristics - Link Training, Bifurcation, Downgrading and Lane Reversal Support

5.2.1 Link Training

The IOH PCI Express port 0 and port 1 will support the following Link widths: x16, x8, x4, x2 and x1. The IOH PCI Express port 2 will support link widths x4, x2 and x1. During link training, the IOH will attempt link negotiation starting from the highest and ramp down to the nearest supported link width that passes negotiation. Each of the widths (x16, x8, x4, x2, x1) are trained in both the non-lane-reversed and lane-reversed modes. For example, x16 link width is trained in both the non-lane-reversed and lane-reversed modes before attempting a dual x8 configuration.

5.2.2 Port Bifurcation

IOH supports port bifurcation via two different means:

- Using the hardware straps. [Table 19-12](#) illustrates the strapping options for ports 0, 1, and 2.
- Via BIOS by appropriately programming the PCIE_PRTx_BIF_CTRL register

5.2.2.1 Port Bifurcation via BIOS

When BIOS needs to control port bifurcation, the hardware strap needs to be set to "Wait_on_BIOS". This instructs the LTSSM to not train till BIOS explicitly enables port bifurcation by programming the PCIE_PRTx_BIF_CTRL register. The default of the latter register is such as to halt the LTSSM from training at poweron, provided the strap is set to "Wait_on_BIOS". When BIOS programs the appropriate bifurcation information into the register, it can initiate port bifurcation by writing to the "Start bifurcation" bit in the register. Once BIOS has started the port bifurcation, it cannot initiate any more *bifurcation* commands without resetting the IOH. Note that software can initiate link retraining within a sub-port or even change the width of a sub-port (by programming the PCIE_PRTx/ESI_LANE_MSK register) any number of times without resetting the IOH. Please refer to [Section 21.12.5.22-Section 21.12.6](#).

Here is a pseudo-code example for how the register and strap work together to control port bifurcation. Note that "strap to ltssm" indicates the IOH internal strap to the LTSSM.



```
If (PCIE_PRT<0,1>_BIF_CTRL[2:0]/PCIE_PRT2_BIF_CTRL[1:0] == 111/11) {  
    If (<PE0/1CFGSEL[2:0]>, <PE2CFGSEL[1:0]>!= <111>,<11>) {  
        Strap to Itssm = strap  
    } else {  
        Wait for "PCIE_PRTx_BIF_CTRL[3]" bit to be set  
        Strap to Itssm = csr  
    }  
} else {  
    Strap to Itssm = csr  
}
```

Note that the bifurcation control registers are sticky and BIOS can chose to program the register and cause an IOH reset and the appropriate bifurcation will take effect on exit from that reset.

5.2.3 Degraded Mode

Degraded mode is supported for x16, x8, x4 and x2 link widths. IOH supports degraded mode operation at half the original width and quarter of the original width or a x1. This mode allows one half or one quarter of the link to be mapped out if one or more lanes should fail during normal operation. This allows for continued system operation in the event of a lane failure. Without support for degraded mode, a failure on a critical lane like lane 0 could bring the entire link down in a fatal manner. This can be avoided with support for degraded mode operation. For example, if lane 0 fails on a x8 link, then the lower half of the link will be disabled and the traffic will continue at half the performance on lanes 4-7. Similarly, a x4 link would degrade to a x2 link. This remapping should occur in the physical layer and the link and transaction layers are transparent to the link width change. The degraded mode widths are automatically attempted every time the PCI Express link is trained. The events that trigger the PCI Express link training are per the *PCI Express Base Specification, Revision 2.0*. For example, if a packet is retried on the link N times (where N is per the *PCI Express Base Specification, Revision 2.0*) then a physical layer retraining is automatically initiated. When this retraining happens, IOH starts out with negotiating a link width that it is currently operating at and if that fails, starts out with negotiating a lower link width per the degraded mode operation.

IOH supported degraded modes are shown below. The [Table 5-1](#) should be read such that the various modes indicated in the different rows would be tried by IOH, but not necessarily in the order shown in the table. IOH would try a higher width degraded mode before trying any lower width degraded modes.



Table 5-1. Supported Degraded Modes

| Original Link Width ¹ | Degraded Mode Link width and Lanes Numbers |
|----------------------------------|--|
| x16 | x8 on lanes 7-0, 0-7, 15-8, 8-15 |
| | x4 on lanes 3-0, 0-3, 4-7, 7-4, 8-11, 11-8, 12-15, 15-12 |
| | x2 on lanes 1-0, 0-1, 4-5, 5-4, 8-9, 9-8, 12-13, 13-12 |
| | x1 on lanes 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 |
| x8 | x4 on lanes 7-4, 4-7, 3-0, 0-3 |
| | x2 on lanes 5-4, 4-5, 1-0, 0-1 |
| | x1 on lanes 0, 1, 2, 3, 4, 5, 6, 7 |
| x4 | x2 on lanes 1-0, 0-1 |
| | x1 on lanes 0, 1, 2, 3 |
| x2 | x1 on lanes 0, 1 |

Notes:

1. This is the native width of the link before degraded mode operation

IOH reports entry into or exit from degraded mode to software (see Section 21.12.5.19) and also records which lane failed.

5.2.4 Lane Reversal

IOH supports lane reversal on all its PCI Express ports, regardless of the link width that is, lane reversal works in x16, x8, x4 and x2 link widths. Note that IOH supports logic that allows a x4, x8 or x16 card to be plugged into a x16 slot that is lane-reversed on the motherboard, and operate at the max width of the card. Similarly for a x4, x8 card plugged into a x8 lane-reversed slot, x4 card plugged into a lane-reversed x4 slot and a x2 card plugged into a lane-reversed x2 slot. Note that for the purpose of this discussion, a “xN slot” means a CEM/SIOM slot that is capable of any width higher than or equal to xN but is electrically wired on the board for only a xN width. A x2 card can be plugged into a x16, x8 or x4 slot and work as x2 only if lane-reversal is not done on the motherboard otherwise it would operate in x1 mode.

5.2.5 PCI Express* Gen1/Gen2 Speed Selection

In general, the IOH will negotiate PCI Express Gen1 versus Gen2 link speed inband during link training.

5.2.6 Form-Factor Support

The IOH supports Cardedge and Server I/O Module (SIOM) form-factors. Form-factor specific differences that exist for hot-plug and power management are captured in their individual sections.

5.3 IOH Performance Policies

5.3.1 Max_Payload_size

IOH will support a Max_Payload_Size of 256B.



5.3.2 Isochronous Support and Virtual Channels

IOH supports the default virtual channel (virtual channel 0) and any TC on the PCI Express interfaces.

5.3.3 Non-Coherent Transaction Support

5.3.3.1 Inbound

Non-coherent transactions are identified by the NoSnoop attribute in the PCI Express request header being set. PCI Express ports in IOH must provide support for converting these transactions to Non-coherent read/writes on Intel QuickPath Interconnect. For writes the NoSnoop attribute is used in conjunction with the Relaxed Ordering attribute to reduce snoops on Intel QuickPath Interconnect interface. For inbound reads with NoSnoop attribute set, IOH does not snoop on Intel QuickPath Interconnect. This optimization for reads and writes can be individually disabled.

5.3.3.2 Outbound

IOH always clears the NoSnoop attribute bit in the PCI Express header for transactions that it forwards from the CPU. For peer 2 peer transactions from other PCI Express ports and ESI, the NoSnoop attribute is passed as is from the originating port.

5.3.4 Completion Policy

The *PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*'s* requires that completions for a specific request must occur in linearly-increasing address order. However, completions for different requests are allowed to complete in any order. As long as the above rules are followed, the IOH will send the completions on the PCI Express interface in the order received from the Intel QuickPath Interconnect interface and never artificially delay completions received from Intel QuickPath Interconnect to PCI Express.

5.3.4.1 Read Completion Combining

The *PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*'s* allows that a single request can be satisfied with multiple "sub-completions" as long as they return in linearly-increasing address order. Therefore, since the IOH must split requests into cacheline quantities before issue on Intel QuickPath Interconnect, the IOH will often complete a large request in cacheline-sized sub-completions.

As a performance optimization, the IOH implements an opportunistic read completion combining algorithm for all reads towards main memory. When the downstream PCI Express interface is busy (e.g. with another transaction) and multiple cachelines have returned before completion on PCI Express is possible, the PCI Express interface will combine the cacheline sub-completions into larger quantities up to MAX_PAYLOAD.

5.3.5 Read Prefetching Policies

The IOH will not perform any prefetching on behalf of interfacing PCI Express component reads. The PCI Express component is solely responsible for its own prefetch algorithms since those components are best suited to make appropriate trade-offs.

The IOH will not perform any outbound read prefetching.



5.3.6 Error Reporting

PCI Express reports many error conditions through explicit error messages: ERR_COR, ERR_NONFATAL, ERR_FATAL. The IOH can be programmed to do one of the following when it receives one of these error messages:

- Generate MSI
- Assert pins ERR[2:0]
- Forward the messages to the ICH

Refer to the *PCI Express Base Specification, Revision 2.0* for details of the standard status bits that are set when a root complex receives one of these messages.

5.3.7 Intel Chipset-Specific Vendor-Defined Messages

Intel chipset-specific vendor-defined messages (VDMs) are identified with a Vendor ID of 8086 in the message header and a specific message code. Refer to Enterprise Southbridge Interface Specification for more details.

5.3.7.1 ASSERT_GPE / DEASSERT_GPE

Upon receipt of an Assert_GPE message from a PCI Express port, the IOH forwards the message to the ICH. When the GPE event has been serviced, the IOH will receive a Deassert_GPE message on the PCI Express port. At this point the IOH can send the deassert_GPE message on ESI.

When an IOH does not have its ESI port enabled for legacy, it forwards the messages over the Intel QuickPath Interconnect.

5.4 Inbound Transactions

This section discusses the IOH behavior towards transactions that originate from PCI Express. Throughout this section, inbound refers to the direction towards main memory from I/O.

5.4.1 Inbound Memory, I/O and Configuration Transactions Supported

Table 5-2 lists the memory, I/O and configuration transactions supported by the IOH which are expected to be received from the PCI Express.

Table 5-2. Incoming PCI Express Memory, I/O and Configuration Request/Completion Cycles (Sheet 1 of 2)

| PCI Express* Transaction | Address Space or Message | IOH Response |
|--------------------------|--------------------------|---|
| Inbound Write Requests | Memory | Forward to Main Memory, PCI Express port (local or remote) or ESI (local or remote) depending on address. |
| | I/O | Forward to PCI Express port (local or remote) or ESI (local or remote). |
| | Type 0 Configuration | Forward to the peer-to-peer port whose device number matches the device number in the Type 0 transaction, if enabled. |
| | Type 1 Configuration | Forward to peer PCI Express port (local or remote) or ESI (local or remote). |



Table 5-2. Incoming PCI Express Memory, I/O and Configuration Request/Completion Cycles (Sheet 2 of 2)

| PCI Express* Transaction | Address Space or Message | IOH Response |
|----------------------------|--------------------------|--|
| Outbound Write Completions | I/O | Forward to processor, PCI Express port (local or remote) or ESI (local or remote). Refer to Section 5.4.2 for handling of Configuration retry completions that target the processor. |
| | Configuration | |
| Inbound Read Requests | Memory | Forward to Main Memory, PCI Express port (local or remote), ESI (local or remote). |
| | I/O | Forward to peer PCI Express port (local or remote), ESI (local or remote). |
| | Type 0 Configuration | Forward to the peer-to-peer port whose device number matches the device number in the Type 0 transaction, if enabled. |
| | Type 1 Configuration | Forward to peer PCI Express port (local or remote) or ESI (local or remote). |
| Outbound Read Completions | Memory | Forward to CPU, PCI Express port (local or remote) or ESI (local or remote). Refer to Section 5.4.2 for handling of Configuration retry completions that target the processor. |
| | I/O | |
| | Configuration | |

5.4.2 Configuration Retry Completions

When a PCI Express port receives a configuration completion packet with a configuration retry status, it reissues the transaction on the affected PCI Express port or completes it. There is an ECN to 1.0a spec that allows for Configuration retry from PCI Express to be visible to software by returning a value of 0x01 on configuration retry (CRS status) on configuration reads to the VendorID register. This ECN provides details of when a root port reissues a configuration transaction and when it is required to complete the transaction.

Here is the summary of when IOH decides to reissue a configuration request.

- When configuration retry software visibility is disabled via the root control register:
 - A configuration request (read or write and regardless of address) is reissued when a CRS response is received for the request and the Configuration Retry Timeout timer has not expired. If the timer has expired, a CRS response received after that will be aborted and a UR response is sent.
 - An “Timeout Abort” response is sent on the Intel® QuickPath Interconnect at the expiry of every 48 ms from the time the request has been first sent on PCI Express until the request has been retired.
- When configuration retry software visibility is enabled via the root control register:
 - The reissue rules as stated previously apply to all configuration transactions, except for configuration reads to vendor ID field at DWORD offset 0x0. When a CRS response is received on a configuration read to VendorID field at word address 0x0, IOH completes the transaction normally with a value of 0x01 in the data field and all 1s in any other bytes included in the read. Refer to *PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s* for more details.

Note: An IOH-aborted configuration transaction is treated as if the transaction returned a UR status on PCI Express, except that the associated PCI header space status and the AER status/log registers are not set.



5.4.3 Inbound PCI Express Messages Supported

Table 5-3 lists all inbound messages that IOH supports receiving on a PCI Express downstream port (does not include ESI messages). In a given system configuration, certain messages are not applicable being received inbound on a PCI Express port. They will be called out as appropriate.

Table 5-3. Incoming PCI Express* Message Cycles

| PCI Express Transaction | Address Space or Message | IOH Response |
|-------------------------|--|--|
| Inbound Message | ASSERT_INTA DEASSERT_INTA ASSERT_INTB DEASSERT_INTB ASSERT_INTC DEASSERT_INTC ASSERT_INTD DEASSERT_INTD | Inband interrupt assertion/deassertion emulating PCI interrupts. Forward to ESI. |
| | ERR_COR ERR_NONFATAL ERR_FATAL | PCI Express error messages propagate as an interrupt to system or cause the ERR[2:0] pins to toggle. |
| | PM_PME | Propagate as an interrupt/general purpose event to the system. |
| | PME_TO_ACK | Received PME_TO_ACK bit is set when IOH receives this message. |
| | PM_ENTER_L1 (DLLP) | Block subsequent TLP issue and wait for all pending TLPs to Ack. Then, send PM_REQUEST_ACK. Refer to <i>PCI Express Architecture Specification</i> , Revision 1.0a for details of the L1 entry flow. |
| | ATC Invalidation Complete | When an end point device completes an ATC invalidation, it will send an Invalidate Complete message to the IOH (RC). This message will be tagged with information from the Invalidate message so that the IOH can associate the Invalidate Complete with the Invalidate Request. |
| Vendor-defined | ASSERT_GPE DEASSERT_GPE (Intel-specific) | Vendor-specific message indicating assertion/deassertion of PCI-X* hot-plug event in PXH. Message forwarded to ESI port. Refer to Section 5.3.7.1, "ASSERT_GPE / DEASSERT_GPE" for further details. |
| | All Other Messages | Silently discard if message type is type 1 and drop and log error if message type is type 0 |

5.5 Outbound Transactions

This section describes the IOH behavior towards outbound transactions. Throughout the rest of the chapter, outbound refers to the direction from processor towards I/O.

5.5.1 Memory, I/O and Configuration Transactions Supported

The IOH generates the outbound memory, I/O and configuration transactions listed in Table 5-4.



Table 5-4. Outgoing PCI Express* Memory, I/O and Configuration Request/Completion Cycles

| PCI Express* Transaction | Address Space or Message | Reason for Issue |
|---------------------------|--------------------------|--|
| Outbound Write Requests | Memory | Memory-mapped I/O write targeting a PCI Express device. |
| | I/O | Legacy I/O write targeting a PCI Express device. |
| | Configuration | Configuration write targeting a PCI Express device. |
| Inbound Write Completions | I/O | Response for an inbound write to a peer I/O device. |
| | Configuration | Response for an inbound write to a peer I/O device or to the originating port. |
| Outbound Read Requests | Memory | Memory-mapped I/O read targeting a PCI Express device. |
| | I/O | Legacy I/O read targeting a PCI Express device. |
| | Configuration | Configuration read targeting PCI Express device. |
| Inbound Read Completions | Memory | Response for an inbound read to main memory or a peer I/O device. |
| | I/O | Response for an inbound read to a peer I/O device. |
| | Configuration | Response for an inbound read to a peer I/O device or to the originating port. |

5.5.2 Lock Support

For legacy PCI functionality, the IOH supports bus locks through an explicit sequence of events. The IOH can receive a locked transaction sequence on the Intel QuickPath Interconnect interface directed to a PCI Express port.

5.5.3 Outbound Messages Supported

Table 5-5 provides a list of all the messages supported by the IOH as an initiator on a PCI Express port.

Table 5-5. Outgoing PCI Express Message Cycles

| PCI Express* Transaction | Address Space or Message | Reason for Issue |
|---------------------------------------|--------------------------|---|
| Outbound Messages | Unlock | Releases a locked read or write transaction previously issued on PCI Express. |
| | PME_Turn_Off | When PME_TO bit in the MISCCTRLSTS register is set, send this message to the associated PCI Express port. |
| | PM_REQUEST_ACK (DLLP) | Acknowledges that the IOH received a PM_ENTER_L1 message. This message is continuously issued until the receiver link is idle. Refer to the <i>PCI Express Base Specification</i> , Revision 2.0 for details. |
| | PM_Active_State_Nak | When the IOH receives a PM_Active_State_Request_L1. |
| | Set_Slot_Power_Limit | Message that is sent to a PCI Express device when software writes to the Slot Capabilities Register or the PCI Express link transitions to DL_Up state. Refer to <i>PCI Express Base Specification</i> , Revision 2.0 for more details. |
| Intel Chipset-specific Vendor-defined | EOI | End-of-interrupt cycle received on the Intel QuickPath Interconnect. The IOH broadcasts this message to all downstream PCI Express and ESI ports. |

5.5.3.1 Unlock

This message is transmitted by the IOH at the end of a lock sequence. This message is transmitted regardless of whether PCI Express lock was established or whether the lock sequence terminated in an error.



5.5.3.2 EOI

EOI messages will be multicast from the Intel QuickPath Interconnect to all the PCI Express interfaces/ESI ports that have an APIC below them. Presence of an APIC is indicated by the EOI enable bit (refer to [Chapter 21](#)). This ensures that the appropriate interrupt controller receives the end-of-interrupt.

5.6 32-/64-Bit Addressing

For inbound and outbound memory reads and writes, the IOH supports the 64-bit address format. If an outbound transaction's address is less than 4 GB, the IOH will issue the transaction with a 32-bit addressing format on PCI Express. Only when the address is greater than 4 GB will the IOH initiate transactions with 64-bit addressing format. Refer to [Chapter 7](#) for details of addressing limits imposed by the Intel[®] QuickPath Interconnect and the resultant address checks that IOH does on PCI Express packets it receives.

5.7 Transaction Descriptor

The *PCI Express Base Specification*, Revision 2.0 defines a field in the header called the Transaction Descriptor. This descriptor comprises three sub-fields:

- Transaction ID
- Attributes
- Traffic class



5.7.1 Transaction ID

The Transaction ID uniquely identifies every transaction in the system. The Transaction ID comprises four sub-fields described in Table 5-6. The table provides details on how this field in the PCI Express header is populated by the IOH:

Table 5-6. PCI Express Transaction ID Handling

| Field | Definition | IOH as Requester | IOH as Completer |
|-----------------|---|--|--|
| Bus Number | Specifies the bus number that the requester resides on. | The IOH fills this field in with zeros. | The IOH preserves this field from the request and copies it into the completion. |
| Device Number | Specifies the device number of the requester. | For CPU requests, the IOH fills this field in with its Device Number that the PCI Express cluster owns. | |
| Function Number | Specifies the function number of the requester. | The IOH fills this field in with its Function Number that the PCI Express cluster owns (zero). | |
| Tag | Denotes a unique identifier for every transaction that requires a completion. Since the PCI Express ordering rules allow read requests to pass other read requests, this field is used to reorder separate completions if they return from the target out-of-order. | <p>Non-Posted (NP) Transaction: The IOH fills this field in with a value such that every pending request carries a unique Tag. NP Tag[7:5]=Intel QuickPath Interconnect Source NodeID[4:2]. Note that bits 7:5 can be non-zero only when 8-bit tag usage is enabled. Otherwise, the IOH always zeros out 7:5. NP Tag[4:0]=the IOH guarantees uniqueness across all pending NP requests from the port.</p> <p>Posted Transaction: No uniqueness is guaranteed. Tag[7:0]=Intel QuickPath Interconnect Source NodeID[7:0] for processor requests. Note that bits 7:5 can be non-zero only when 8-bit tag usage is enabled. Otherwise, the IOH always zeros out bits 7:5.</p> | |

Note: Follow the TransactionID rules outlined in the table above for compliance to features like Intel VT-d and DCA.



5.7.2 Attributes

PCI Express supports two attribute hints described in Table 5-7. This table describes how the IOH populates these attribute fields for requests and completions it generates.

Table 5-7. PCI Express Attribute Handling

| Attribute ¹ | Definition | IOH as Requester | IOH as Completer |
|------------------------|--|---|--|
| Relaxed Ordering | Allows the system to relax some of the standard PCI ordering rules. | This bit is not applicable and set to zero. | The IOH preserves this field from the request and copies it into the completion. |
| Snoop Not Required | This attribute is set when an I/O device controls coherency through software mechanisms. This attribute is an optimization designed to preserve processor snoop bandwidth. | | |

Notes:

1. Refer to the Chapter 3 for how the IOH uses these attributes for performance optimizations.

5.7.3 Traffic Class

The IOH does not optimize based on traffic class. IOH can receive a packet with TC equal not to 0 and treat the packet as if it were TC equal to 0 from an ordering perspective. IOH forwards the TC field as-is on peer-to-peer requests and also returns the TC field from the original request on the completion packet sent back to the device.

5.8 Completer ID

The CompleterID field is used in PCI Express completion packets to identify the completer of the transaction. The CompleterID comprises three sub-fields described in Table 5-8.

Table 5-8. PCI Express CompleterID Handling

| Field | Definition | IOH as Completer |
|-----------------|---|---|
| Bus Number | Specifies the bus number that the completer resides on. | The IOH returns 00h as the Bus number |
| Device Number | Specifies the device number of the completer. | The IOH returns 00000b as the Device Number |
| Function Number | Specifies the function number of the completer. | 0 |



5.9 Miscellaneous Information

5.9.1 Number of Outbound Non-Posted Requests

Each x4 PCI Express link supports up to two outstanding non-posted outbound transactions issued by the processors. Each x8 link supports up to four and x16 supports up to eight outstanding non-posted transactions.

5.9.2 MSIs Generated from Root Ports and Locks

Once lock has been established on the Intel QuickPath Interconnect, the IOH cannot send any requests on the Intel QuickPath Interconnect, including MSI transactions generated from the root port of the PCI Express port that is locked.

5.9.3 Completions for Locked Read Requests

Both LkRdCmp and RdCmp completion types can terminate a locked or non-locked read request.

5.10 PCI Express RAS

The IOH supports the PCI Express Advanced Error Reporting (AER) capability. Refer to *PCI Express Base Specification*, Revision 2.0 for details.

5.10.1 ECRC Support

The IOH does not support the PCI Express end-to-end CRC (ECRC) feature. The IOH ignores and drops ECRC on all incoming packets and does not generate ECRC on any outgoing packet.

5.10.2 Completion Time-Out

For all non-posted requests that the IOH issues on PCI Express or ESI, IOH maintains a timer that times the max completion time for that request.

IOH follows the time-out mechanism ECN that is currently being proposed in the PCI Express Base Specification. The ECN provides a way for the OS to select a coarse range for the timeout value. The IOH then chooses a final value of the timeout within each coarse range. The timeout value is programmable from 50 ms all the way up to 64 seconds. Refer to [Chapter 21](#) for details and additional control that the IOH provides for the 17 second to 64 second timeout range.

Refer to the [Chapter 16](#) for details of responses returned by the IOH to various interfaces on a completion time-out event. AER-required error logging and escalation happen as well. In addition to the AER error logging, the IOH also sets the locked read time-out bit in the Miscellaneous Control and Status Register if the completion time-out happened on a locked read request. See the [Chapter 21](#) for details.



5.10.3 Data Poisoning

The IOH supports forwarding of poisoned data among its interfaces.

The IOH provides an optional mode where poisoned data is never sent out on PCI Express; any packet with poisoned data is dropped by the IOH and generate an error. See the [Chapter 16](#) for details

5.10.4 Role-Based Error Reporting

The IOH supports the new role-based error reporting feature being amended to the PCI Express base specification 1.1. Details of how the IOH handles various error cases under this role-based error reporting scheme are as follows.

A Poisoned TLP received on peer-to-peer packets is treated as an *advisory* non-fatal error condition. that is, ERR_COR is signaled and the poisoned information propagated peer-to-peer.

Poisoned TLP on packets destined for internal devices of the IOH are treated, from a PCI Express interface error reporting perspective, as a *normal*, non-fatal error condition.

Poisoned TLP on packets destined towards DRAM, or poisoned TLP packets that target the interrupt address range, are forwarded to the Intel QuickPath Interconnect with the poison bit set, provided the Intel QuickPath Interconnect interface is enabled to set the poisoned bit via QPIPC[12]. In such a case the received poisoned TLP condition is treated as *advisory* non-fatal error on the PCI Express interface. If that bit is not set, the IOH treats the received poisoned TLP condition as a *normal*, non-fatal error. The packet is dropped if it is a posted transaction. A “master abort” response is sent on the Intel® QuickPath Interconnect if the poisoned TLP received was for an outstanding non-posted request.

When the IOH times out, or receives a UR/CA response on a request outstanding on PCI Express, it does not attempt recovery in hardware. Also, it would treat the completion time-out condition as a *normal*, non-fatal error condition. UR/CA received does not cause an error escalation.

5.11 Link Layer Specifics

5.11.1 Ack/Nak

The Data Link layer is responsible for ensuring that TLPs are successfully transmitted between PCI Express agents. PCI Express implements an Ack/Nak protocol to accomplish this. Every TLP is decoded by the Physical layer (8b/10b) and forwarded to the Link layer. The CRC code appended to the TLP is then checked. If this comparison fails, the TLP is retried. Refer to [Section 5.11.2](#) for details.

If the comparison is successful, an Ack is issued back to the transmitter and the packet is forwarded for decoding by the receiver’s Transaction layer. The PCI Express protocol allows that Acks can be combined and the IOH implements this as an efficiency optimization.

Generally, Naks are sent as soon as possible. Acks, however, will be returned based on a timer policy such that when the timer expires, all unacknowledged TLPs to that point are Aacked with a single Ack DLLP. The timer is programmable.



5.11.2 Link Level Retry

The *PCI Express Base Specification*, Revision 2.0 lists all the conditions where a TLP gets Nak'd. One example is on a CRC error. The Link layer in the receiver is responsible for calculating 32 bit CRC (using the polynomial defined in *PCI Express Base Specification*, Revision 2.0) for incoming TLPs and comparing the calculated CRC with the received CRC. If they do not match, then the TLP is retried by Nak'ing the packet with a Nak DLLP specifying the sequence number of the corrupt TLP. Subsequent TLPs are dropped until the reattempted packet is observed again.

When the transmitter receives the Nak, it is responsible for retransmitting the TLP specified with the Sequence number in the DLLP + 1. Furthermore, any TLPs sent after the corrupt packet will also be resent since the receiver has dropped any TLPs after the corrupt packet.

5.11.2.1 Retry Buffer

The IOH transmitter retry buffer is designed such that under normal conditions there is no performance degradation. Unless there is a CRC error at the receiver, the transmitter will never back up (at Gen2 speeds) due to insufficient room in the retry buffer. The following environment is assumed:

- 3 m of cable + 25" FR4 total
- Two repeaters
- Four connectors

5.11.3 Ack Time-Out

Packets can get "lost" if the packet is corrupted such that the receiver's Physical layer does not detect the framing symbols properly. Frequently, lost TLPs are detectable with non-linearly incrementing sequence numbers. A time-out mechanism exists to detect (and bound) cases where the *last* TLP packet sent (over a long period of time) was corrupted. A replay timer bounds the time a retry buffer entry waits for an Ack or Nak. Refer to the *PCI Express Base Specification*, Revision 2.0 for details on this mechanism.

5.11.4 Flow Control

The PCI Express flow control types are described in following tables.

Table 5-9. PCI Express Credit Mapping for Inbound Transactions (Sheet 1 of 2)

| Flow Control Type | Definition | Initial IOH Advertisement |
|---|---|--------------------------------|
| Inbound Posted Request Header Credits (IPRH) | Tracks the number of posted requests the agent is capable of supporting. Each credit accounts for one posted request. | 24(x4) 48(x8) 96(X16) |
| Inbound Posted Request Data Credits (IPRD) | Tracks the number of posted data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data. | 108(x4) 216(x8) 432(X16) |
| Inbound Non-Posted Request Header Credits (INPRH) | Tracks the number of non-posted requests the agent is capable of supporting. Each credit accounts for one non-posted request. | 24(x4) 48(x8) 96(X16) |



Table 5-9. PCI Express Credit Mapping for Inbound Transactions (Sheet 2 of 2)

| Flow Control Type | Definition | Initial IOH Advertisement |
|---|---|---------------------------|
| Inbound Non-Posted Request Data Credits (INPRD) | Tracks the number of non-posted data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data. | 4(x4) 8(x8) 16(X16) |
| Completion Header Credits (CPH) | Tracks the number of completion headers the agent is capable of supporting. | infinite |
| Completion Data Credits (CPD) | Tracks the number of completion data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data. | infinite |

Every PCI Express device tracks the above six credit types for both itself and the interfacing device. The rules governing flow control are described in *PCI Express Base Specification, Revision 2.0*.

Note: The credit advertisement in [Table 5-9](#) does not necessarily imply the number of *outstanding* requests to memory.

The IOH keeps a pool of credits that are allocated between the ports based on their partitioning. For example, assume the NPRH credit pool is N for the x8 port. If this port is partitioned as two x4 ports, the credits advertised are N/2 per port.

Table 5-10. PCI Express Credit Mapping for Outbound Transactions

| Flow Control Type | Definition | Initial IOH Advertisement |
|--|---|-----------------------------|
| Outbound Posted Request Header Credits (OPRH) | Tracks the number of posted requests the agent is capable of supporting. Each credit accounts for one posted request. | 4(x4) 8(x8) 16(X16) |
| Outbound Posted Request Data Credits (OPRD) | Tracks the number of posted data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data. | 8(x4) 16(x8) 32(X16) |
| Outbound Non-Posted Request Header Credits (ONPRH) | Tracks the number of non-posted requests the agent is capable of supporting. Each credit accounts for one non-posted request. | 14(x4) 24(x8) 56(X16) |
| Onbound Non-Posted Request Data Credits (ONPRD) | Tracks the number of non-posted data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data. | 12(x4) 24(x8) 48(X16) |
| Completion Header Credits (CPH) | Tracks the number of completion headers the agent is capable of supporting. | 6(x4) 12(x8) 24(X16) |
| Completion Data Credits (CPD) | Tracks the number of completion data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data. | 12(x4) 24(x8) 48(X16) |

5.11.4.1 Flow Control Credit Return by IOH

After reset, credit information is initialized with the values indicated in [Table 5-9](#) by following the flow control initialization protocol defined in the *PCI Express Base Specification, Revision 2.0*. Since the IOH supports only VC0, only this channel is initialized. As a receiver, the IOH is responsible for updating the transmitter with flow control credits as the packets are accepted by the Transaction Layer. Credits will be returned as follows:

- If infinite credits advertised, there are no flow control updates for that credit class, as per the *PCI Express Base Specification, Revision 2.0*



- For non-infinite credits advertised, the IOH will send flow control updates if none were sent previously, for example, after 28 usec (to comply with the specification's 30 usec requirement). This 28 us is programmable down to 6 us.
- If, and only when, there are credits to be released, the IOH will wait for a configurable/programmable number of cycles (in the order of 30-70 cycles) before the flow control update is sent. This is done on a per flow-control credit basis. This mechanism ensures that credits updates are not sent when there is no credit to be released.

5.11.4.2 Flow Control Update DLLP Time-Out

IOH supports the optional flow control update DLLP time-out timer.

5.12 Power Management

IOH does not support the beacon wake method on PCI Express. IOH supports Active State Power Management (ASPM) transitions into L0s and L1 state. Also, IOH supports the D0 and D3hot power management states per PCI Express port and also supports a wake event from these states on a PCI Express hot-plug event. In D3hot, IOH will master abort all configuration TX targeting the PCI Express link.

5.13 Enterprise South Bridge Interface (ESI)

The Enterprise South Bridge Interface (legacy port) in the IOH is responsible for sending and receiving packets/commands to legacy components in the system like the ICH. PCI Express link and uses the same physical/link layers as described in the PCI Express Chapter.

Features:

- The ESI port supports x4, x2 and x1 and the width is auto-negotiated at power-on.
- Port bifurcation is NOT supported on ESI i.e. the ESI port is always negotiated as one single port.
- Downgrading is supported from x4 to x2 and x1. Lane reversal is also supported.
- The IOH only supports Gen1 speed.

5.13.1 ESI Port as a PCI Express Gen1 Port

When the LEGACYIOH strap is set to "0" and FWAGNT_ESIMODE strap is set to "1", the ESI port will work as a Gen1 PCI Express port.

5.13.2 Configuration Retry Completion

The IOH handles configuration retry from ESI similar to configuration retry from PCI Express. Refer to [Section 5.4.2](#) for details of how a PCI Express port handles configuration retry completions.

5.13.3 Inbound Transactions

This section reviews transactions that are supported by IOH coming in from the ESI interface.



5.13.3.1 Memory, I/O and Configuration Transactions Supported

Table 5-11 lists the memory, I/O and Configuration transactions in the ESI port which are expected to be received by the IOH.

Table 5-11. Incoming ESI Memory, I/O and Configuration Requests/Completions

| ESI Transaction | Transaction Type | IOH Response |
|----------------------------|------------------|---|
| Inbound Write Requests | Memory | Forward to Main Memory, integrated device or PCI Express port depending on address. |
| | Config | Forward to PCI Express ports (peer or local) or IOH internal config space depending on bus number and device number. |
| Inbound Read Requests | Memory | Forward to Main Memory, integrated device or PCI Express port depending on address. |
| | Config | Forward to PCI Express ports (peer or local) or IOH internal config space depending on bus number and device number. |
| Outbound Write Completions | I/O | Forward to the interface that originated the request which is determined based on the requestor bus number and device number. |
| | Configuration | |
| Outbound Read Completions | Memory | Forward to the interface that originated the request which is determined based on the requestor bus number and device number. |
| | I/O | |
| | Configuration | |

5.13.3.2 Messages Supported

Table 5-12 below lists all messages that are supported by IOH inbound from ESI.

Table 5-12. Incoming ESI Messages

| ESI Transaction | Transaction Type | IOH Response |
|-------------------------------|--------------------|--|
| Standard PCI Express Messages | ERR_COR | The ICH will send ERR* messages to the IOH in response to error conditions. |
| | ERR_NONFATAL | |
| | ERR_FATAL | |
| Intel Vendor-defined Messages | Go_C0/S3 | Power management related messages. |
| | Go_C2 | |
| | CPU_Reset_Done_Ack | Reset related messages. |
| | Rst_Warn | |
| | INTR_Ack_Reply | Issued by the ICH in response to an "INTR_Ack" message from the IOH. The INTR_Ack_Reply message contains the interrupt vector (from the 8259 controller) and the IOH completes the pending INTR_ACK transaction on Intel® QuickPath Interconnect with the required information. Note that there can be only one outstanding INTR_Ack transaction from the processor. |
| | Assert_PHLD | The Phold handshake is used to flush the contents of the IOH's downstream write buffers prior to granting the ICH's ISA-legacy bus master logic ownership. ICH generates the Assert_PHLD message to the IOH. IOH flushes and disables downstream write buffers and then generates the Assert_PHLDA message to the ICH. When the ICH's ISA-legacy bus master operations are complete, the ICH generates the Deassert_PHLD message to IOH. |
| Deassert_PHLD | | |

5.13.3.3 Configuration Retry Completion

IOH handles configuration retry from ESI similar to configuration retry from PCI Express. Refer to [Section 5.4.2, "Configuration Retry Completions" on page 92](#) for details of how an Express port handles configuration retry completions.

5.13.4 Outbound Transactions

This section describes outbound transactions supported by the IOH on the ESI link.



5.13.4.1 Outbound Memory, I/O and Configuration Transactions Supported

Table 5-13 lists the outbound memory, I/O and Configuration requests and completions supported by the IOH on ESI.

Table 5-13. Outgoing ESI Memory, I/O and Configuration Requests/Completions

| ESI Transaction | Transaction Type | Reason for Issue |
|--------------------------|------------------|--|
| Outbound Write Requests | Memory | Processor or peer memory-mapped I/O write targeting ICH. |
| | I/O | Processor or peer legacy I/O write targeting ICH. |
| | Configuration | Processor or peer Configuration write targeting ICH. |
| Outbound Read Requests | Memory | Processor or memory-mapped I/O read targeting ICH |
| | I/O | Processor or PCI Express I/O read targeting ICH. |
| | Configuration | Configuration read targeting ICH. |
| Inbound Read Completions | Memory | Response for an inbound read to main memory, integrated device or PCI Express. |

5.13.4.2 Outbound Messages Supported

Table 5-14 lists all outbound messages supported by the IOH on ESI.

Table 5-14. Outgoing ESI Messages (Sheet 1 of 2)

| ESI Transaction | Transaction Type | Reason for Issue |
|-------------------------------|------------------|---|
| Standard PCI Express Messages | Unlock | When a locked read or write transaction was previously issued to the ESI, "Unlock" releases the PCI lock. |
| | Assert_INTA | Issued by the IOH through the ESI port when a PCI Express interface receives a legacy interrupt message on its standard PCI Express ports or generates them internally. Note that these events are level sensitive at the source and the IOH will send an aggregated message (wired-or) to the ESI for each interrupt level. Refer to the Chapter 8, "Interrupts" for further details of how these messages are handled through the IOH from the receiving PCI Express interface. |
| | Assert_INTB | |
| | Assert_INTC | |
| | Assert_INTD | |
| | Deassert_INTA | |
| | Deassert_INTB | |
| | Deassert_INTC | |
| Deassert_INTD | | |



Table 5-14. Outgoing ESI Messages (Sheet 2 of 2)

| ESI Transaction | Transaction Type | Reason for Issue |
|-------------------------------|----------------------------------|--|
| Intel Vendor-defined Messages | PM_Active_State_NAK | The IOH will generate the "PM_Active_State_NAK" message to the ICH in response to receiving a "PM_Active_State_Request_L1" DLLP because the IOH cannot transition to the L1 state. Refer to <i>PCI Express Base Specification, Revision 2.0</i> for further details on the L1 ASPM flow. |
| | Rst_Warn_Ack | The IOH sends the Acknowledge in response to a prior "Rst_Warn" message. Refer to the Chapter 14, "Reset" for details of the IOH reset flow and how this message is handled. |
| | EOI | The IOH will broadcast an EOI encoded as a TLP Data message with EOI vector embedded in the payload. Refer Section 5.5.3.2 for details of the EOI broadcast. |
| | Assert_GPE Deassert_GPE | The IOH will forward a collapsed version of the Assert_GPE and Deassert_GPE it receives from its PCI Express ports. Refer to Section 5.3.7.1 for further details. |
| | Assert_HPGPE Deassert_HPGPE | The IOH will send a hot-plug GPE message, "Assert_HPGPE" when a hot-plug event is detected (and native OS handling of hot-plug is disabled). The "Deassert_HPGPE" is sent when the hot-plug event has been serviced. |
| | Assert_PMEGPE Deassert_PMEGPE | The IOH will send a "Assert_PMEGPE" when a Power Management event is detected (and native OS handling of hot-plug is disabled). The "Deassert_PMEGPE" is sent when the Power Management event has been completed. |
| | Ack_C0 | Refer to the Chapter 10, "Power Management" for details. |
| | Ack_S3 | |
| | Ack_C2 | |
| | CPU_Reset_Done | Reset related message. Refer to the Chapter 14, "Reset" for further details. |
| | INTR_Ack | Issued by the IOH when the Processor sends an interrupt acknowledge command on Intel QuickPath Interconnect. This is treated as an outbound posted message and sent to the ICH. There can be only one INTR_Ack outstanding from the processor at a time. |
| | DO_SCI | Needed for Intel QuickPath Interconnect-based ACPI events |

5.13.4.3 Lock Support

For legacy PCI functionality, the IOH supports bus locks to the ESI port.

5.13.4.4 PHOLD Support

The IOH supports the PHOLD protocol. This protocol is used for legacy ISA devices which do not allow the possibility for being both a master and a slave device simultaneously. Example devices that use the PHOLD protocol are legacy floppy drives, and so on.

5.13.4.5 PHOLD/PHOLDA

A PHOLD regime is established when IOH issues an Assert_PHLDA message to ESI and is terminated when the IOH receives a Deassert_PHLD message on ESI. The IOH will not send posted or non-posted requests to ESI port during a PHOLD regime and will only allow downstream completions. All non-posted peer-to-peer traffic should be disabled during the PHOLD regime to avoid deadlock situations within IOH.

5.13.4.6 ICH Behavior

Once ICH has sent an Assert_PHLD message, it will not send a Deassert_PHLD message until the IOH has sent an Assert_PHLDA message.



5.13.4.6.1 Intel® QuickPath Interconnect Lock Request

When the IOH receives an Assert_PHLDA message on ESI, it will generate a request to lock arbiter.

5.13.4.6.2 Block All Sources of Transactions

Once the Intel QuickPath Interconnect lock is established, the IOH flushes the queues and sends an Assert_PHLDA message to ICH on the ESI.

5.13.5 64-Bit Addressing

For processor and peer-to-peer writes and reads, the IOH supports 64-bit address format on the ESI to and from the ICH.

5.13.6 Transaction Descriptor

The Transaction Descriptor comprises three sub-fields:

- Transaction ID
- Attributes
- Traffic Class

5.13.6.1 Transaction ID

The Transaction ID uniquely identifies every transaction in the system. The Transaction ID comprises four sub-fields described in [Table 5-15, "ESI Transaction ID Handling"](#).



Table 5-15. ESI Transaction ID Handling

| Field | Definition | IOH as Requester | IOH as Completer |
|-----------------|--|--|--|
| Bus Number | Specifies the bus number that the requester resides on. | The IOH fills this field in with its internal Bus Number that the ESI cluster resides on (refer to the Chapter 21, "Configuration Register Space" for details). | The IOH preserves this field from the request and copies it into the completion. |
| Device Number | Specifies the device number of the requester. | 9 | |
| Function Number | Specifies the function number of the requester. | 0 | |
| Tag | Unlike PCI Express this field is allowed to be non-unique on ESI. IOH as a requester will not use non-unique TID on ESI but as a completer will preserve any original request TID. | <p>Non-Posted (NP) Transaction: The IOH fills this field in with a value such that every pending request carries a unique Tag.</p> <p>NP Tag[7:4]=Intel QuickPath Interconnect Source NodeID[4:1]. Note that bits [7:5] can be non-zero only when 8-bit tag usage is enabled (in the P2P register corresponding to the ESI port). Otherwise, the IOH always zeros out bits [7:5].</p> <p>NP Tag[3:0]=Any algorithm that guarantees uniqueness across all pending NP requests from the port. Posted Transaction: No uniqueness guaranteed.</p> <p>Tag[7:0]=Intel QuickPath Interconnect Source NodeID[7:0] for processor requests. Note that bits [7:5] can be non-zero only when 8-bit tag usage is enabled. Otherwise, IOH always zeros out bits [7:5].¹</p> | |

Notes:

1. The IOH never uses non-unique tag as requester on ESI.

5.13.6.2 Attributes

ESI supports two attribute hints described in [Table 5-16](#).

Table 5-16. ESI Attribute Handling

| Attribute | Definition | IOH as Requester | IOH as Completer |
|--------------------|--|---|--|
| Relaxed Ordering | Allows the system to relax some of the standard PCI ordering rules. | For outbound transactions, this bit is not applicable and set to zero. For peer-to-peer requests, preserve this field from the source PCI Express port to the destination port. | The IOH preserves this field from the request and copies it into the completion. |
| Snoop Not Required | This attribute is set when an I/O device controls coherency through software mechanisms. This attribute is an optimization designed to preserve processor snoop bandwidth. | | |

The IOH supports ESI virtual channels VC0, VC1, and VCp.



5.13.7 Completer ID

The CompleterID field is used in ESI completion packets to identify the completer of the transaction. The CompleterID comprises three sub-fields described in [Table 5-17](#). The table provides details on how this field is populated by the IOH for completions it generates to ESI.

Table 5-17. ESI CompleterID Handling

| Field | Definition | IOH as Completer |
|-----------------|---|---|
| Bus Number | Specifies the bus number that the completer resides on. | The IOH fills this field in with its internal Bus Number that the ESI cluster resides on. |
| Device Number | Specifies the device number of the completer. | 9 |
| Function Number | Specifies the function number of the completer. | 0 |

5.14 Flow Control Credits Advertised on ESI

The ESI port flow control credits advertised are described in [Table 5-18](#).

Table 5-18. ESI Credit Mapping

| Flow Control Type | Definition | Initial IOH Advertisement |
|--|---|---------------------------|
| Posted Request Header Credits (PRH) | Tracks the number of posted requests the agent is capable of supporting. Each credit accounts for one posted request. | 24 |
| Posted Request Data Credits (PRD) | Tracks the number of posted data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data. | 48 |
| Non-Posted Request Header Credits (NPRH) | Tracks the number of non-posted requests the agent is capable of supporting. Each credit accounts for one non-posted request. | 16 |
| Non-Posted Request Data Credits (NPRD) | Tracks the number of non-posted data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data. | 1 |
| Completion Header Credits (CPH) | Tracks the number of completion headers the agent is capable of supporting. | infinite |
| Completion Data Credits (CPD) | Tracks the number of completion data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data. | infinite |

§



6 Ordering

6.1 Introduction

The IOH spans two different ordering domains: one that adheres to Producer-Consumer ordering (PCI Express) and one that is completely unordered (Intel QuickPath Interconnect). One of the primary functions of the IOH is to ensure that the Producer-Consumer ordering model is maintained in the unordered, Intel QuickPath Interconnect domain.

This section describes the rules that are required to ensure that both PCI Express and Intel QuickPath Interconnect ordering is preserved. Throughout this chapter, the following terms are used:

Table 6-1. Ordering Term Definitions

| Term | Definition |
|--|---|
| Intel QuickPath Interconnect Ordering Domain | The Intel QuickPath Interconnect has a relaxed ordering model allowing reads, writes and completions to flow independent of each other. Intel QuickPath Interconnect implements this through the use of multiple, independent virtual channels. In general, the Intel QuickPath Interconnect ordering domain is considered unordered. |
| PCI Express Ordering Domain | PCI Express (and all other prior PCI generations) have specific ordering rules to enable low cost components to support the Producer-Consumer model. For example, no transaction can pass a write flowing in the same direction. In addition, PCI implements ordering relaxations to avoid deadlocks (for example, completions must pass non-posted requests). The set of these rules are described in <i>PCI Express Base Specification, Revision 1.0a</i> . |
| Posted | A posted request is a request which can be considered ordered (per PCI rules) upon the issue of the request and therefore completions are unnecessary. The only posted transaction is PCI memory writes. Intel QuickPath Interconnect does not implement posted semantics and so to adhere to the posted semantics of PCI, the rules below are prescribed. |
| Non-posted | A non-posted request is a request which cannot be considered ordered (per PCI rules) until after the completion is received. Non-posted transactions include all reads and some writes (I/O and configuration writes). Since Intel QuickPath Interconnect is largely unordered, all requests are considered to be non-posted until the target responds. Throughout this chapter, the term non-posted applies only to PCI requests. |
| Outbound Read | A read issued toward a PCI Express device. This can be a read issued by a processor, an SMBus master, or a peer IOH (in the context of the target IOH). |
| Outbound Read Completion | The completion for an outbound read. For example, the read data which results in a processor read of a PCI Express device. <i>Note that while the data flows inbound, the completion is still for an outbound read.</i> |
| Outbound Write | A write issued toward a PCI Express device. This can be a write issued by a processor, an SMBus master, or a peer IOH (in the context of the target IOH). |
| Outbound Write Completion | The completion for an outbound write. For example, the completion from a PCI Express device which results in a processor-initiated I/O or configuration write. <i>Note that while the completion flows inbound, the completion is still for an outbound write.</i> |
| Inbound Read | A read issued toward an Intel® QuickPath Interconnect component. This can be a read issued by a PCI Express device. An obvious example is a PCI Express device reading main memory. |
| Inbound Read Completion | The completion for an inbound read. For example, the read data which results in a PCI Express device read to main memory. <i>Note that while the data flows outbound, the completion is still for an inbound read.</i> |
| Inbound Write | A write issued toward an Intel QuickPath Interconnect component. This can be a write issued by a PCI Express device. An obvious example is a PCI Express device writing main memory. In the Intel QuickPath Interconnect domain, this write is often fragmented into a request-for-ownership followed by an eventual writeback to memory. |
| Inbound Write Completion | Does not exist. All inbound writes are considered posted (in the PCI Express context) and therefore, this term is never used in this chapter. |

6.2 Inbound Ordering Rules

Inbound transactions originate from PCI Express and target main memory. In general, the IOH forwards inbound transactions in FIFO order. There are exceptions to this under certain situations. For example, PCI Express requires that read completions are allowed to pass stalled read requests. This forces any read completions to bypass any reads which might be back pressured on the Intel QuickPath Interconnect. Sequential, non-posted requests are not required to be completed in the order they were requested.¹

Inbound writes cannot be posted beyond the PCI Express ordering domain. The posting of writes relies on the fact that the system maintains a certain ordering relationship. Since the IOH cannot post inbound writes beyond the PCI Express ordering domain, the IOH must wait for snoop responses before issuing subsequent, order-dependent transactions.

Each of the Intel QuickPath Interconnect ports have no ordering relationship to each other. The IOH relaxes ordering between different PCI Express ports (aside from the peer-to-peer restrictions below).

6.2.1 Inbound Ordering Requirements

In general, there are no ordering requirements between transactions received on different PCI Express interfaces. However, the rules below apply to inbound transactions received on the same interface.

- Rule 1. Outbound non-posted read and non-posted write completions must be allowed to progress past stalled inbound non-posted requests.
- Rule 2. Inbound posted write requests and messages must be allowed to progress past stalled inbound non-posted requests.
- Rule 3. Inbound posted write requests, inbound messages, inbound read requests, outbound non-posted read and outbound non-posted write completions cannot pass enqueued inbound posted write requests.
The Producer - Consumer model prevents read requests, write requests, and non-posted read or non-posted write completions from passing write requests. Refer to *PCI Local Bus Specification, Revision 2.3* for details on the Producer - Consumer ordering model.
- Rule 4. Outbound non-posted read or outbound non-posted write completions must push ahead *all* prior inbound posted transactions from that PCI Express port.
- Rule 5. The IOH is unaware of which destination I/O bus (for example, PCI-X* on the PXH) the read completion comes for outbound transactions. Therefore, the IOH prevents forwarding the read or non-posted write completion to the Intel QuickPath Interconnect until all currently enqueued inbound writes are complete (independent of the VC value).
- Rule 6. Inbound, coherent, posted writes will issue requests for ownership (RFO) without waiting for prior ownership requests to complete. Local-local address conflict checking still applies.
- Rule 7. Inbound messages follow the same ordering rules as inbound posted writes (FENCE messages have their own rules).
Similarly to inbound posted writes, reads should push these commands ahead.

1. The ESI interface has exceptions to this rule specified in [Section 6.2.1](#).



Rule 8. If an inbound read completes with multiple sub-completions (for example, a cache line at a time), those sub-completions must be returned on PCI Express in linearly increasing address order.

The above rules apply whether the transaction is coherent or non-coherent. Some regions of memory space are considered non-coherent (for example, the Don't Snoop attribute is set). The IOH will order all transactions regardless of its destination.

Rule 9. For PCI Express ports, different read requests should be completed without any ordering dependency. For the ESI interface, however, all read requests with the same Tag must be completed in the order that the respective requests were issued.

Different read requests issued on a PCI Express interface should be completed in any order. This attribute is beneficial for the Intel Xeon processor 7500 series-based platform and Intel Itanium processor 9300 series-based platform where the Intel QuickPath Interconnect is an unordered, multipath interface. However, the read completion ordering restriction on ESI implies that the IOH must guarantee stronger ordering on that interface.

6.2.2 Special Ordering Relaxations

The *PCI Express Base Specification*, Revision 1.0a specifies that reads do not have any ordering constraints with other reads. Therefore if one read is blocked (on either Intel[®] QuickPath Interconnect or PCI Express) then subsequent reads will proceed. An example of why a read would be blocked is the case of an Intel QuickPath Interconnect address conflict. Under such a blocking condition, subsequent transactions are allowed to proceed until the blocking condition is cleared.

PCI Express allows inbound write requests to pass outbound read and outbound non-posted write completions. For peer-to-peer traffic, this optimization allows writes to memory to make progress while a PCI Express device is making long read requests to a peer device on the same interface.

6.2.2.1 PCI Express* Relaxed Ordering

The relaxed ordering attribute (RO) is a bit in the header of every PCI Express packet and relaxes the ordering rules such that:

- Posted requests with RO set can pass other posted requests.
- Non-posted completions with RO set can pass posted requests.

The IOH relaxes write ordering for non-coherent, DRAM write transactions with this attribute set. The IOH does not relax the ordering between read completions and outbound posted transactions.

With the exception of peer-to-peer requests, the IOH clears the relaxed ordering for outbound transactions received from the Intel QuickPath Interconnect interface. For local transaction, the attribute is preserved for both requests and completions.

6.3 Outbound Ordering Rules

Outbound transactions through the IOH are memory, I/O or configuration read/write transactions originating on an Intel QuickPath Interconnect interface destined for a PCI Express or ESI device. Subsequent outbound transactions with different destinations



have no ordering requirements between them. Multiple transactions destined for the same outbound port are ordered according to the ordering rules specified in *PCI Express Base Specification, Revision 2.0*.

Note: On the Intel QuickPath Interconnect, non-coherent writes are not considered complete until the IOH returns a Cmp for the NcWr transaction. On PCI Express and ESI interfaces, memory writes are posted. The IOH returns this completion once the write is guaranteed to meet the PCI Express ordering rules and is part of the “ordered domain”. For outbound writes that are non-posted in the PCI Express domain (for example, I/O and configuration writes), the target device will post the completion.

6.3.1 Outbound Ordering Requirements

There are no ordering requirements between outbound transactions targeting different outbound interfaces. For deadlock avoidance, the following rules must be ensured for outbound transactions targeting the same outbound interface:

- Rule 1. Inbound non-posted completions must be allowed to progress past stalled outbound non-posted requests.
- Rule 2. Outbound posted requests must be allowed to progress past stalled outbound non-posted requests.
- Rule 3. Outbound non-posted requests and inbound completions cannot pass enqueued outbound posted requests.
The Producer - Consumer model prevents read requests, write requests, and read completions from passing write requests. Refer to *PCI Local Bus Specification, Revision 2.3* for details on the Producer - Consumer ordering model.
- Rule 4. If a non-posted inbound request requires multiple sub-completions, those sub-completions must be delivered on PCI Express in linearly addressing order.
This rule is a requirement of the PCI Express protocol. For example, if the IOH receives a request for 4 KB on the PCI Express interface and this request targets the Intel QuickPath Interconnect port (main memory), the IOH splits up the request into multiple 64 B requests. Since the Intel QuickPath Interconnect is an unordered domain, it is possible that the IOH receives the second cache line of data before the first. Under such unordered situations, the IOH must buffer the second cache line until the first one is received and forwarded to the PCI Express requester.
- Rule 5. If a configuration write transaction targets the IOH, the completion must not be returned to the requester until after the write has actually occurred to the register.
Writes to configuration registers could have side-effects and the requester expects that it has taken effect prior to receiving the completion for that write. The IOH will not respond to the configuration write until after the register is actually written (and all expected side-effects have completed).

6.3.2 Hinted Peer-to-Peer

There are no specific IOH requirements for hinted peer-to-peer since PCI ordering is maintained on each PCI Express port.

6.3.3 Local Peer-to-Peer

Local peer-to-peer transactions flow through the same inbound ordering logic as inbound memory transactions from the same PCI Express port. This provides a serialization point for proper ordering.



When the inbound ordering logic receives a peer-to-peer transaction, the ordering rules require that it must wait until all prior inbound writes from the same PCI Express port are completed on the Intel QuickPath Interconnect interface. Local peer-to-peer write transactions complete when the outbound ordering logic for the target PCI Express port receives the transaction and returns the completion to the initiating IOH. Local peer-to-peer read transactions are completed by the target device.

6.4 Interrupt Ordering Rules

SAPIC and IOxAPIC interrupts are either directed to a single processor or broadcast to multiple processors. The IOH treats interrupts as posted transactions. This enforces that the interrupt will not be observed until after all prior inbound writes are flushed to their destinations. For broadcast interrupts, order-dependent transactions received after the interrupt must wait until all interrupt completions are received by the IOH.

Interrupts are treated as posted transactions; therefore the ordering rule that read completions push interrupts naturally applies. For example:

- An interrupt generated by a PCI Express interface must be ordered with read completions from configuration registers within that same PCI Express root port.
- Read completions from the integrated IOxAPIC's registers (configuration and memory-mapped I/O space) must push all interrupts generated by the integrated IOxAPIC.
- Read completions from the Intel VT-d registers must push all interrupts generated by the Intel VT-d logic (for example, an error condition). Intel Xeon processor 7500 series-based platform only.

6.4.1 SpcEOI Ordering

When a processor receives an interrupt, it will process the interrupt routine. The processor will then clear the I/O card's interrupt by writing to that I/O device's register. The EOI request is treated as an outbound posted transaction with regard to ordering rules.

6.4.2 SpcINTA Ordering

The legacy 8259 controller can interrupt a processor through a virtual INTR pin (virtual legacy wire). The processor responds to the interrupt by sending an interrupt acknowledge transaction reading the interrupt vector from the 8259 controller. After reading the vector, the processor will jump to the interrupt routine.

The Intel QuickPath Interconnect implements an IntAck message to read the interrupt vector from the 8259 controller. With respect to ordering rules, the Intr_Ack message (always outbound) is treated as a posted request. The completion returns to the IOH on ESI as an Intr_Ack_Reply (also posted). The IOH translates this into a completion for the Intel® QuickPath Interconnect Intr_Ack message.

6.5 Configuration Register Ordering Rules

The IOH implements legacy PCI configuration registers. These registers are accessed with NcCfgRd and NcCfgWr transactions (using PCI Bus, Device, Function) received on the Intel QuickPath Interconnect interface.



For PCI configuration space, the ordering requirements are the same as standard, non-posted configuration cycles on PCI. Refer to [Section 6.2.1](#) and [Section 6.3.1](#) for details. Furthermore, on configuration writes to the IOH the completion is returned by the IOH only after the data is actually written into the register.

6.6 Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d) Ordering Exceptions

The transaction flow to support the address remapping feature of Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d) requires that the IOH reads from an address translation table stored in memory. This table read has the added ordering requirement that it must be able to pass all other inbound non-posted requests (including non-table reads). If not for this bypassing requirement, there would be an ordering dependence on peer-to-peer reads resulting in a deadlock.

§



7 System Address Map

This chapter provides a basic overview of the system address map and describes how the IOH comprehends and decodes the various regions in the system address map. The term “IOH” in this chapter refers to the IOH in multiprocessor End Point and multiprocessor Route-Through modes. This chapter does not provide the full details of the Intel Xeon processor 7500 series based and Intel Itanium processor 9300 series-based platform system address spaces as viewed by software and it also does not provide the details of processor address decoding.

The IOH supports the full 51 bits [50:0] of memory addressing on its Intel QuickPath Interconnect interface. The IOH also supports receiving and decoding 64 bits of address from PCI Express. Memory transactions received from PCI Express that go above the top of physical address space supported on Intel QuickPath Interconnect (which is dependent on the Intel QuickPath Interconnect profile but is always less than or equal to 2^{51} for the IOH) are reported as errors by IOH. The IOH as a requester would never generate requests on PCI Express with any of address bits 63 to 51 set. For packets the IOH receives from Intel QuickPath Interconnect and for packets the IOH receives from PCI Express that fall below the top of Intel QuickPath Interconnect physical address space, the upper address bits from top of Intel QuickPath Interconnect physical address space up to bit 63 must be considered as 0s for target address decoding purposes. The IOH always performs full 64-bit target address decoding.

The IOH supports 16 bits of I/O addressing on its Intel QuickPath Interconnect interface. The IOH also supports receiving and decoding the full 32 bits of I/O address from PCI Express. I/O requests received from PCI Express that are beyond 64 KB are reported as errors by the IOH. The IOH as a requester would never generate I/O requests on PCI Express with any of address bits 31 to 16 set.

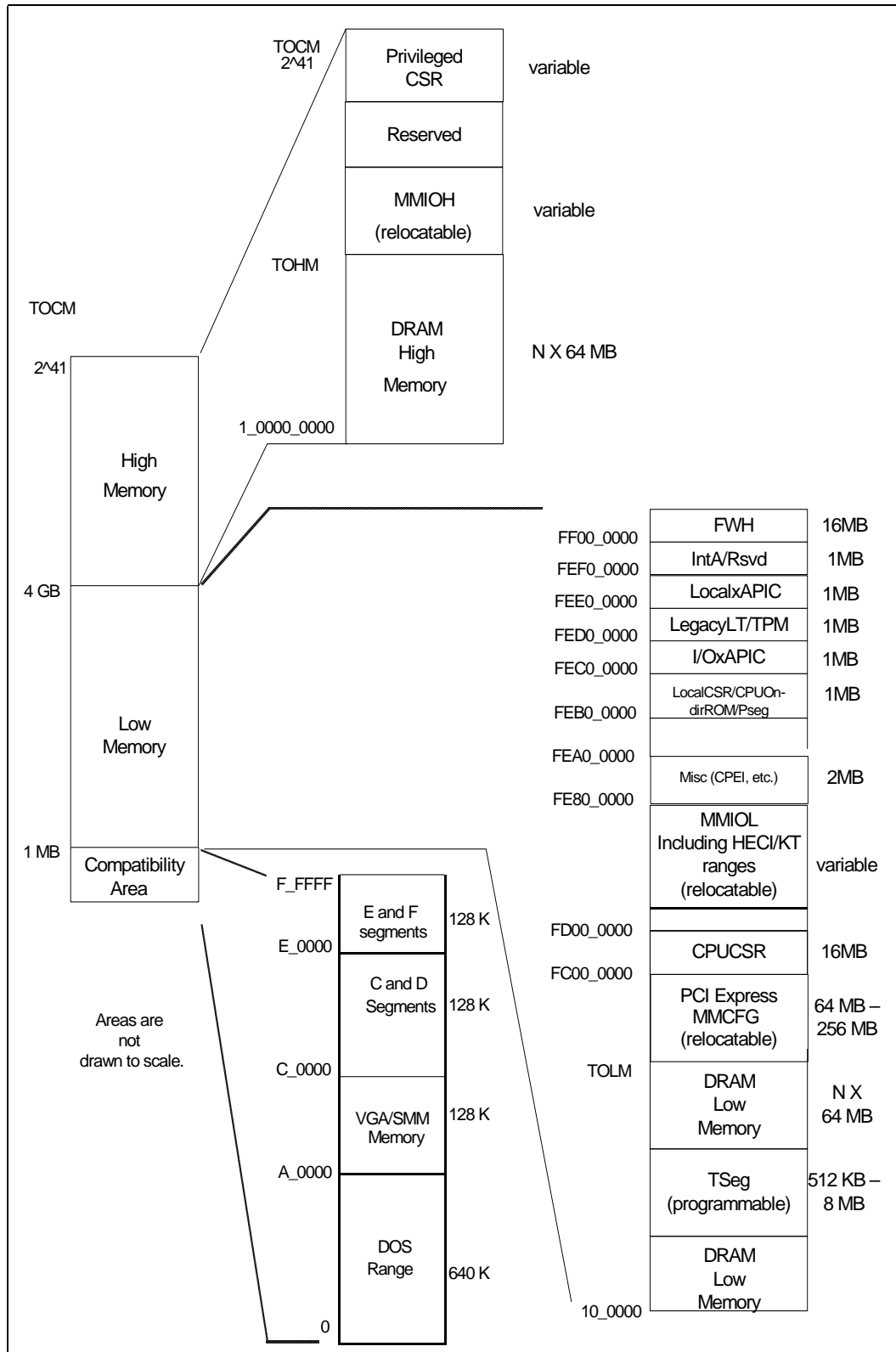
The IOH supports PCI configuration addressing up to 256 buses, 32 devices per bus and 8 functions per device. A single grouping of 256 buses, 32 devices per bus and 8 functions per device is referred to as a PCI *segment*. Intel Xeon processor 7500 series and Intel Itanium processor 9300 series source decoder supports multiple PCI segments in the system. However, all configuration addressing within an IOH and hierarchies below an IOH must be within one segment. The IOH does not support being in multiple PCI segments.

7.1 Memory Address Space

Figure 7-1 shows the Intel Xeon processor 7500 series-based platform and Intel Itanium processor 9300 series-based platform system memory address spaces. There are three basic regions of memory address space in the system: address below 1 MB, address between 1 MB and 4 GB, and address above 4 GB. These regions are described in the following sections.

Throughout this section, there will be references to the *subtractive decode port*. It refers to the port of the IOH that is attached to a legacy ICH or provides a path towards the legacy ICH. This port is also the recipient of all addresses that are not positively decoded towards any PCI Express device or towards memory.

Figure 7-1. System Address Map





7.1.1 System DRAM Memory Regions

| Address Region | From | To |
|---|---------------|---------------|
| 640 KB DOS Memory | 000_0000_0000 | 000_0009_FFFF |
| 1 MB to Top-of-low-memory | 000_0010_0000 | TOLM |
| Bottom-of-high-memory to Top-of-high-memory | 4 GB | TOHM |

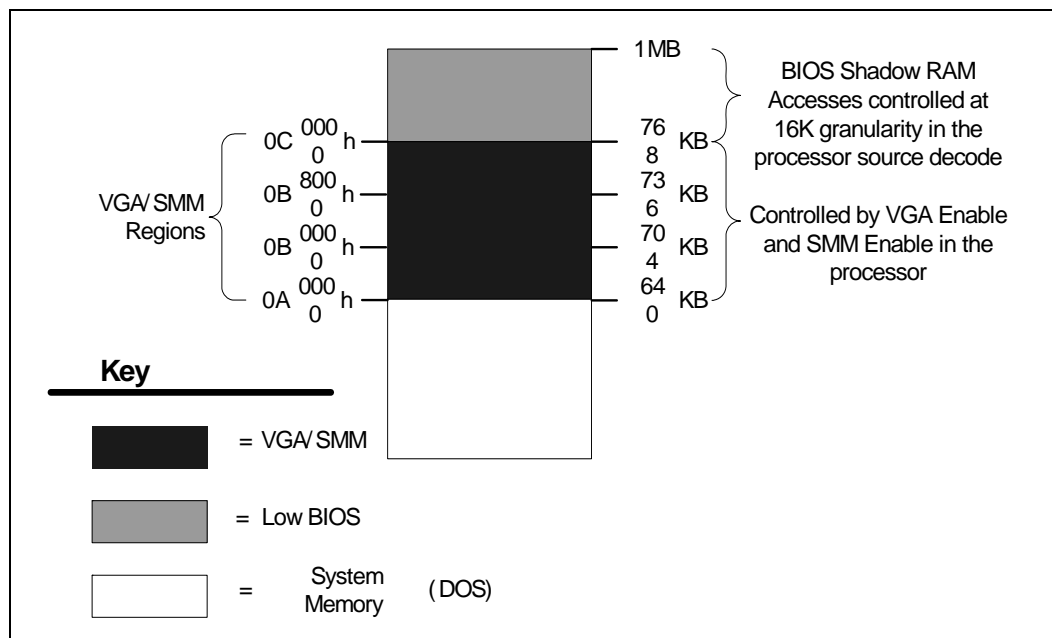
These address ranges are always mapped to system DRAM memory, regardless of the system configuration. The top of main memory below 4 G is defined by the Top of Low Memory (TOLM). Memory between 4 GB and TOHM is extended system memory. Since the platform may contain multiple processors, the memory space is divided amongst the CPUs. There may be memory holes between each processor's memory regions. These system memory regions are either coherent or non-coherent. A set of range registers in the IOH define a non-coherent memory region (NcMem.Base/NcMem.Limit) within the system DRAM memory region shown above. System DRAM memory region outside of this range but within the DRAM region shown in table above is considered coherent.

For inbound transactions, the IOH positively decodes these ranges via a couple of software programmable range registers. For outbound transactions, it would be an error for IOH to receive non-coherent accesses to these addresses from Intel QuickPath Interconnect. However, the IOH does not explicitly check for this error condition and simply forwards such accesses to the subtractive decode port, if one exists downstream, by virtue of subtractive decoding.

7.1.2 VGA/SMM and Legacy C/D/E/F Regions

Figure 7-2 shows the memory address regions below 1 MB. These regions are legacy access ranges.

Figure 7-2. VGA/SMM and Legacy C/D/E/F Regions





7.1.2.1 VGA/SMM Memory Space

| Address Region | From | To |
|----------------|---------------|---------------|
| VGA | 000_000A_0000 | 000_000B_FFFF |

This legacy address range is used by video cards to map a frame buffer or a character-based video buffer. By default, accesses to this region are forwarded to main memory by the processor. However, once firmware figures out where the VGA device is in the system, it sets up the processor's source address decoders to forward these accesses to the appropriate IOH. If the VGAEN bit is set in the IOH PCI bridge control register (BCR) of a PCI Express port, then transactions within the VGA space (defined above) are forwarded to the associated port, regardless of the settings of the peer-to-peer memory address ranges of that port. If none of the PCI Express ports have the VGAEN bit set (note that per the IOH address map constraints the VGA memory addresses cannot be included as part of the normal peer-to-peer bridge memory apertures in the root ports), then these accesses are forwarded to the subtractive decode port. Also refer to the *PCI-PCI Bridge 1.2 Specification* for further details on the VGA decoding. Note that only one VGA device may be enabled per system partition. The VGAEN bit in the PCIe bridge control register must be set only in one PCI Express port in a system partition. The IOH does not support the MDA (monochrome display adapter) space independent of the VGA space.

The VGA memory address range can also be mapped to system memory in SMM. The IOH is totally transparent to the workings of this region in the SMM mode. All outbound and inbound accesses to this address range are always forwarded to the VGA device of the partition, by the IOH. Refer to the [Table 7-4](#) and [Table 7-5](#) for further details of inbound and outbound VGA decoding.

For IOH based platforms, at boot time, the VGA port must be in the legacy IOH and can not be in any non-legacy IOH. Because legacy IO ranges are decoded only by the Legacy IOH, during BIOS boot and until the driver loads in the OS, the device must be used behind the non-legacy IOH. This is because the Legacy IO ranges are used to communicate to the device during initial boot. Once an OS level driver loads, the driver may know how to talk to the device using non-legacy resources. At that point, video behind a non-legacy IOH should work if a driver is loaded that supports that functionality and addressing.

7.1.2.2 C/D/E/F Segments

The E/F region could be used to address DRAM from an I/O device (processors have registers to select between addressing bios flash and dram). IOH does not explicitly decode the E/F region in the outbound direction and relies on subtractive decoding to forward accesses to this region to the legacy ICH. IOH does not explicitly decode inbound accesses to the E/F address region. It is expected that the DRAM low range that IOH decodes will be setup to cover the E/F address range. By virtue of that, the IOH will forward inbound accesses to the E/F segment to system DRAM. If it is necessary to block inbound access to these ranges, the Generic Memory Protection Ranges could be used.

C/D region is used in system DRAM memory for BIOS and option ROM shadowing. The IOH does not explicitly decode these regions for inbound accesses. Software must program one of the system DRAM memory decode ranges that the IOH uses for inbound system memory decoding to include these ranges.



All outbound accesses to the C through F regions are first positively decoded against all valid targets' address ranges and if none match, these address are forwarded to the subtractive decode port of the IOH, if one exists; else it is an error condition.

The IOH will complete locks to this range, but cannot guarantee atomicity when writes and reads are mapped to separate destinations.

7.1.3 Address Region Between 1 MB and TOLM

This region is always allocated to system DRAM memory. Software must set up one of the coarse memory decode ranges that IOH uses for inbound system memory decoding to include this address range. The IOH will forward inbound accesses to this region to system memory (unless any of these access addresses fall within a protected dram ranges protected as described in Chapter 7, "Protected System DRAM Regions"). It would be an error for IOH to receive outbound accesses to an address in this region, other than snoop requests from Intel QuickPath Interconnect links. However, the IOH does not explicitly check for this error condition, and simply forwards such accesses to the subtractive decode port.

Any inbound access that decodes within one of the two coarse memory decode windows with no physical DRAM populated for that address will result in a master abort response on PCI Express.

7.1.3.1 Relocatable TSeg

| Address Region | From | To |
|----------------|---------------------|---------------------|
| TSeg | FE00_0000 (default) | FE7F_FFFF (default) |

These are system DRAM memory regions that are used for SMM/CMM mode operation. IOH would completter abort all inbound transactions that target these address ranges. IOH should not receive transactions that target these addresses in the outbound direction, but IOH does not explicitly check for this error condition but rather subtractively forwards such transactions to the subtractive decode port of the IOH, if one exists downstream.

The location (1 MB aligned) and size (from 512 KB to 8 MB) in IOH can be programmed by software.

7.1.4 Address Region from TOLM to 4 GB

7.1.4.1 PCI Express Memory Mapped Configuration Space

This is the system address region that is allocated for software to access the PCI Express Configuration Space. This region is relocatable below 4 GB by BIOS/firmware; the IOH has no explicit knowledge of this address range. All inbound and outbound accesses to this region are sent to the subtractive decode port of the IOH by virtue of subtractive decoding. It is the responsibility of software to make sure that this system address range is not included in any of the system DRAM memory ranges that the IOH decodes inbound. Otherwise, these addresses could potentially be sent to the processor by the IOH.



7.1.4.2 MMIOL

| Address Region | From | To |
|----------------|-------------|--------------|
| MMIOL | GMMIOL.Base | GMMIOL.Limit |

This region is used for PCI Express device memory addressing below 4 GB. Each IOH in the system is allocated a portion of this address range; individual PCI Express ports within an IOH use sub-portions within that range. Each IOH has MMIOL address range registers (LMMIOL and GMMIOL) to support local peer-to-peer in the MMIOL address range. Refer to [Section 7.5](#) for details of how these registers are used in the inbound and outbound MMIOL range decoding.

7.1.4.3 CPU CSR Memory Space

| Address Region | From | To |
|----------------|-----------|-----------|
| CPU CSRs | FC00_0000 | FCFF_FFFF |

This range is used to accommodate the CSR registers in the processors. The IOH should not receive any inbound transactions from its PCI Express ports towards this address range. If such inbound accesses occur, they are aborted and IOH returns a completer abort response. The IOH should not receive any outbound transactions from any Intel QuickPath Interconnect link to this address range. However, the IOH does not explicitly check for this error condition, and simply forwards these outbound transactions to the subtractive decode port, if one exists downstream. Refer to [Section 7.5.1](#) for further details.

7.1.4.4 Miscellaneous (Misc)

This region is used by the processor for miscellaneous functionality including an address range that software can write to generate interrupt messages on Intel QuickPath Interconnect, and so on. The IOH aborts all inbound accesses to this region. Outbound accesses to this region is not explicitly decoded by IOH and are forwarded to downstream subtractive decode port, if one exists; it is otherwise master aborted.

| Address Region | From | To |
|----------------|-----------|-----------|
| Misc | FE80_0000 | FE9F_FFFF |

7.1.4.5 Processor Local CSR

| Address Region | From | To |
|----------------|-----------|-----------|
| CPU Local CSR | FE80_0000 | FEBF_FFFF |

This region accommodates processor's local CSRs. The IOH will block all inbound accesses from PCI Express to this address region and return a completer abort response. Outbound accesses to this address range are not part of the normal programming model and the IOH subtractively sends such accesses to the subtractive decode port of the IOH, if one exists downstream (else, error).



7.1.4.6 I/OxAPIC Memory Space

| Address Region | From | To |
|----------------|-----------|-----------|
| I/OxAPIC | FECO_0000 | FECF_FFFF |

This is a 1 MB range used to map I/OxAPIC Controller registers. The I/OxAPIC spaces are used to communicate with I/OxAPIC interrupt controllers that may be populated in the downstream devices, such as PXH, and the IOH's integrated I/OxAPIC. The I/OxAPIC space is divided among the IOHs in the system. Each IOH can be associated with an I/OxAPIC range. The range can be further divided by various downstream ports in the IOH and the integrated I/OxAPIC. Each downstream port in the IOH contains a Base/Limit register pair (APICBase/APICLimit) to decode its I/OxAPIC range. Addresses that fall within this range are forwarded to that port. Similarly, the integrated I/OxAPIC decodes its I/OxAPIC base address via the ABAR register (refer to [Chapter 21, "Configuration Register Space"](#)). The range decoded via the ABAR register is a fixed size of 256B. Note that the integrated I/OxAPIC also decodes a standard PCI-style 32-bit BAR (located in the PCI defined BAR region of the PCI header space) that is 4 KB in size, called the MBAR register (refer to [Chapter 21, "Configuration Register Space"](#)). The MBAR register is provided so that the I/OxAPIC can be placed anywhere in the 4 G memory space.

Only outbound accesses are allowed to this FEC address range and also to the MBAR region. Inbound accesses to this address range return a completer abort response. Outbound accesses to this address range that are not positively decoded towards any one PCI Express port are sent to the subtractive decode port of the IOH. Refer to [Section 7.5.1, "Outbound Address Decoding"](#) and [Section 7.5.2, "Inbound Address Decoding"](#) for details of outbound address decoding to the I/OxAPIC space.

Accesses to the I/OxAPIC address region (APIC Base/APIC Limit) of each root port, are decoded by the IOH irrespective of the setting of the MemorySpaceEnable bit in the root port peer-to-peer bridge register.

7.1.4.7 HPET/Others

| Address Region | From | To |
|----------------|-----------|-----------|
| HPET/Others | FED0_0000 | FEDF_FFFF |

This region covers the High performance event timers, and so on, in the ICH. All inbound/peer-to-peer accesses to this region are completer aborted by the IOH.

Outbound non-locked Intel QuickPath Interconnect accesses (that is, accesses that happen when Intel QuickPath Interconnect quiescence is not established) to the FED4_0xxx region are converted by IOH before forwarding to legacy ESI port. All outbound Intel QuickPath Interconnect accesses (that is, accesses that happen after Intel QuickPath Interconnect quiescence has been established) to FED4_0xxx range are aborted by non-IOH. Also IOH aborts all locked Intel QuickPath Interconnect accesses to the FED4_0xxx range. Other outbound Intel QuickPath Interconnect accesses in the FEDx_xxxx range, but outside of the FED4_0xxx range are forwarded to legacy ESI port by virtue of subtractive decoding.



7.1.4.8 Local XAPIC

| Address Region | From | To |
|----------------|-----------|-----------|
| Local XAPIC | FEEO_0000 | FEEF_FFFF |

The CPU Interrupt address space is used to deliver interrupts to the CPU(s). MSI from PCIe devices target this address and are forwarded as SpcInt messages to the CPU. Refer to [Chapter 8, “Interrupts”](#) for details of interrupt routing.

The CPUs may also use this region to send inter-processor interrupts (IPI) from one processor to another. The IOH is never a recipient of such an interrupt. Inbound reads to this address are considered errors and are completed with an unsupported request response by the IOH. Outbound accesses to this address are also considered as errors. However, the IOH does not explicitly check for this error condition but simply forwards the transaction subtractively to its subtractive decode port, if one exists downstream.

7.1.4.9 Firmware

| Address Region | From | To |
|----------------|-----------|-----------|
| HIGHBIO | FF00_0000 | FFFF_FFFF |

This ranges starts at FF00_0000 and ends at FFFF_FFFF. It is used for BIOS/Firmware. Outbound accesses within this range are forwarded to the firmware hub devices. During boot initialization, IOH with firmware connected south of it will communicate this on all Intel QuickPath Interconnect ports so that CPU hardware can configure the path to firmware. The IOH does not support accesses to this address range inbound that is, those inbound transactions are aborted and a completer abort response is sent back.

7.1.5 Address Regions above 4 GB

7.1.5.1 Memory Mapped I/O High (MMIOH)

| Address Region | From | To |
|----------------|-------------|--------------|
| MMIOH | GMMIOH.Base | GMMIOH.Limit |

The high memory mapped I/O range is located above main memory. This region is used to map I/O address requirements above the 4 GB range. IOH in the system is allocated a portion of this system address region and within that portion, each PCI Express port use up a sub-range.

Each IOH has MMIOH address range registers (LMMIOH and GMMIOH) to support local and remote peer-to-peer in the MMIOH address range. Refer to [Section 7.5.1, “Outbound Address Decoding”](#) and [Section 7.5.2, “Inbound Address Decoding”](#) for details of inbound and outbound decoding for accesses to this region.

7.1.5.2 High System Memory

| Address Region | From | To |
|--------------------|------|------|
| High System Memory | 4 GB | TOHM |



This region is used to describe the address range of system memory above the 4 GB boundary. The IOH forwards all inbound accesses to this region to system memory (unless the requested addresses are also marked as protected. See [Chapter 21, “Configuration Register Space”](#)). A portion of the address range within this high system DRAM region could be marked non-coherent (via NcMem.Base/NcMem.Limit register) and the IOH treats them as non-coherent. All other addresses are treated as coherent (unless modified via the NS attributes on PCI Express). The IOH should not receive outbound accesses to this region. However, the IOH does not explicitly check for this error condition but rather subtractively forwards these accesses to the subtractive decode port of the IOH, if one exists downstream (else, error).

Software must set up this address range such that any recovered DRAM hole from below the 4 GB boundary, that might encompass a protected sub-region, is not included in the range.

7.1.5.3 Privileged CSR Memory Space

| Address Region | From | To |
|----------------|-----------------------|------|
| Privileged CSR | TOCM-64 GB (variable) | TOCM |

This region is used to block inbound access to processor CSRs. This region is located at the top of the Intel QuickPath Interconnect physical memory (TOCM) space which can be either 2^{51} , 2^{46} or 2^{41} , depending on the Intel QuickPath Interconnect profile. This range is above the IOH’s TOHM register and should not overlap with the MMIOH range; therefore, IOH should not positively decode this range and will abort any inbound accesses. IOH should not see any outbound accesses to this range. Refer to [Section 7.5.1.2, “FWH Decoding”](#) for more details of IOH decoding of this privileged CSR region.

7.1.5.4 BIOS Notes on Address Allocation Above 4 GB

Since the IOH supports only a single, contiguous address range for accesses to system memory above 4 G, BIOS must make sure that there is enough reserved space gap left between the top of high memory (TOHM) and the bottom of the MMIOH region, if memory hot add is required. This gap can be used to address hot added memory in the system and would fit the constraints imposed by IOH decode mechanism.

7.1.6 Protected System DRAM Regions

The IOH supports an address range for protecting various system DRAM regions that carry protected OS code or other proprietary platform information. The ranges are

- Intel VT-d protected high range
- Intel VT-d protected low range
- Intel Itanium processor 9300 series protected DRAM range/McSeg for Intel Xeon processors

The Intel VT-d protected ranges protect default page tables set up by Intel VT-d aware OS. There is one of these ranges for above and below 4G. These ranges can be disabled. Note that when these ranges are enabled, IOH protects these addresses regardless of whether translation is enabled or not. When translation is enabled, these ranges are specified in the HPA domain.



The IOH provides a 64-bit programmable address window for this purpose. All accesses that hit this address range are completely aborted by the IOH. This address range can be placed anywhere in the system address map and could potentially overlap one of the coarse DRAM decode ranges.

7.2 I/O Address Space

There are four classes of I/O addresses that are specifically decoded by the IOH:

1. I/O addresses used for VGA controllers.
2. I/O addresses used for ISA aliasing
3. I/O addresses used for the PCI Configuration protocol – CFC/CF8
4. I/O addresses used by downstream PCI/PCIe I/O devices, typically legacy devices. The range can be divided by various downstream ports in the IOH. Each downstream port in the IOH contains a BAR to decode its I/O range. Addresses that fall within this range are forwarded to its respective IOH, then subsequently to the downstream port.

7.2.1 VGA I/O Addresses

Legacy VGA device uses up the addresses 3B0h-3BBh, 3C0h-3DFh. Any PCI Express or ESI port in the IOH can be a valid target of these address ranges if the VGAEN bit in the peer-to-peer bridge control register corresponding to that port is set (besides the condition where these regions are positively decoded within the peer-to-peer I/O address range). In the outbound direction, by default, the IOH decodes only the bottom 10 bits of the 16 bit I/O address when decoding this VGA address range with the VGAEN bit set in the peer-to-peer bridge control register. When the VGA16DECEN bit is set in addition to VGAEN being set, the IOH performs a full 16 bit decode for that port when decoding the VGA address range outbound. In general, on outbound accesses to this space, IOH positively decodes the address ranges of all PCIe ports per the peer-to-peer bridge decoding rules (refer to the *PCI-PCI Bridge 1.2 Specification* for details). When no target is positively identified, the IOH sends it to its subtractive decode port, if one exists. Else, error. For inbound accesses to the VGA address range, IOH always performs full 16 bit I/O decode.

7.2.2 ISA Addresses

The IOH supports ISA addressing per the *PCI-PCI Bridge 1.2 Specification*. ISA addressing is enabled for a PCI Express port via the Bridge Control Register (BCR). Note that when the VGA Enable bit is set for a PCI Express port without the VGA 16-bit Decode Enable bit being set, the ISA Enable bit must be set in all the peer PCI Express ports in the *system*.

7.2.3 CFC/CF8 Addresses

The CFC/CF8 addresses are used by legacy operating systems to generate PCI configuration cycles. The IOH does not explicitly decode the CFC/CF8 I/O addresses or take any specific action. These accesses are decoded as part of the normal inbound and outbound I/O transaction flow, and follow the same routing rules. Refer also to [Table 7-3](#) and [Table 7-4](#) for details of I/O address decoding.



7.2.4 PCI Express Device I/O Addresses

These addresses could be anywhere in the 64 KB I/O space and are used to allocate I/O addresses to PCI Express devices. Each IOH is allocated a chunk of I/O address space; there are IOH-specific requirements on how these chunks are distributed to support peer-to-peer. Each IOH has I/O address range registers (LIO and GIO) to support local peer-to-peer in the I/O address range. Refer to [Section 7.5.1](#) and [Section 7.5.2](#) for details.

7.3 Configuration/CSR Space

There are two types of configuration/CSR space in the IOH: PCI Express configuration space and Intel QuickPath Interconnect CSR space. PCI Express configuration space is the standard PCI Express configuration space defined in the PCI Express specification. CSR space is memory mapped space used exclusively for special processor registers.

7.3.1 PCI Express Configuration Space

PCI Express configuration space allows for up to 256 buses, 32 devices per bus and 8 functions per device. There could be multiple groups of these configuration spaces and each is called a *segment*. The IOH can support multiple segments in a system. PCI Express devices are accessed via NcCfgWr/Rd transactions on Intel QuickPath Interconnect. Within each segment, bus 0 is always assigned to the internal bus number of the IOH which has the legacy ICH attached to it. Refer to [Section 7.5.1](#) and [Section 7.5.2](#) for details.

Each IOH is allocated a chunk of PCIe bus numbers and there are IOH-specific requirements on how these chunks are distributed amongst IOHs to support peer-to-peer. Refer to [Section 7.6, "Intel VT-d Address Map Implications"](#) for details of these restrictions. Each IOH has a set of configuration bus range registers (LCFGBUS and GCFGBUS) to support local and remote peer-to-peer. Refer to [Section 7.5.1, "Outbound Address Decoding"](#) and [Section 7.5.2, "Inbound Address Decoding"](#) for details of how these registers are used in the inbound and outbound memory/configuration/message decoding.

7.3.2 Processor CSR Space

The processor CSR space is different from the PCI Express configuration space and is accessed via the NcWrPtl and NcRd transactions on Intel QuickPath Interconnect. These regions are fixed in memory space between FC00_0000 to FFFF_FFFF.

The IOH allocates all its Intel QuickPath Interconnect and core registers to this space. Refer to [Section 7.5.1.2](#) for details.

7.4 IOH Address Map Notes

7.4.1 Memory Recovery

When software recovers an underlying DRAM memory region that resides below the 4 GB address line that is used for system resources like firmware, localAPIC, and IOAPIC, and so on (the gap below 4 GB address line), it needs to make sure that it does not create system memory holes whereby all the system memory cannot be decoded with two contiguous ranges. It is OK to have unpopulated addresses within these contiguous ranges that are not claimed by any system resource. IOH decodes all



inbound accesses to system memory via two contiguous address ranges (0-TOLM, 4 GB-TOHM) and there cannot be holes created inside of those ranges that are allocated to other system resources in the gap below 4 GB address line. The only exception to this is the hole created in the low system DRAM memory range via the VGA memory address. IOH comprehends this and does not forward these VGA memory regions to system memory.

7.4.2 Non-Coherent Address Space

The IOH supports one coarse main memory range which can be treated as non-coherent by the IOH, that is, inbound accesses to this region are treated as non-coherent. This address range has to be a subset of one of the coarse memory ranges that the IOH decodes towards system memory. Inbound accesses to the NC range are not snooped on Intel QuickPath Interconnect.

7.5 IOH Address Decoding

In general, software needs to guarantee that for a given address there can only be a single target in the system. Otherwise, results are undefined. The one exception is that VGA addresses would fall within the inbound coarse decode memory range. The IOH inbound address decoder forwards VGA addresses to the VGA port in the system only (and not system memory).

7.5.1 Outbound Address Decoding

This section covers address decoding that IOH performs on a transaction from Intel QuickPath Interconnect targets one of the downstream ports of the IOH. For the remainder of this section, the term PCI Express generically refers to all I/O ports: standard PCI Express, or ESI, unless noted otherwise.

7.5.1.1 General Overview

- Before any transaction from Intel QuickPath Interconnect is validly decoded by IOH, the NodeID in the incoming transaction must match the NodeIDs assigned to the IOH; otherwise, it is an error.
- All target decoding towards PCI Express, firmware, and internal IOH devices, follow address-based routing. Address-based routing follows the standard PCI tree hierarchy routing.
- NodeID based routing is not supported south of the Intel QuickPath Interconnect port in the IOH, except in MP Route-Through mode (when the local NodeID is not matched from the Intel QuickPath Interconnect port, the transaction is routed to the other Intel QuickPath Interconnect port).
- The subtractive decode port in an IOH is the port that is a) the recipient of all addresses that are not positively decoded towards any of the valid targets in the IOH and b) the recipient of all message/special cycles that are targeted at the legacy ICH.
 - This can be the ESI or the Intel QuickPath Interconnect port. SUBDECEN bit in the IOH Miscellaneous Control Register (IOHMISCCTRL) sets the subtractive port of the IOH. This bit is set by the LEGACYIOH strap.
 - Virtual peer-to-peer bridge decoding related registers with their associated control bits (for example, VGAEN bit) and other miscellaneous address ranges (I/OxAPIC) of a ESI port are NOT valid (and ignored by the IOH decoder) when they are set as the subtractive decoding port.



- Unless specified otherwise, all addresses (no distinction made) are first positively decoded against all target address ranges. Valid targets are PCI Express, ESI, CB DMA and I/OxAPIC devices. A PCI Express or ESI port are invalid targets for positive decode of Memory/IO/Configuration/Message cycles, if the subtractive decoding has been enabled for that port. Besides the standard peer-to-peer decode ranges for PCI Express ports (refer to the *PCI-PCI Bridge 1.2 Specification* for details), the target addresses for these ports also include the I/OxAPIC address ranges. Software has the responsibility to make sure that only one target can ultimately be the target of a given address and IOH will forward the transaction towards that target.
 - For outbound transactions, when no target is positively decoded, the transactions are sent to the downstream ESI port if it is indicated as the subtractive decode port. If ESI port is not the subtractive decode port, the transaction is master aborted.
 - For inbound transactions, when no target is positively decoded, the transactions are sent to the subtractive decode port which is either Intel QuickPath Interconnect or ESI port.
- For positive decoding, the memory decode to each PCI Express target is governed by Memory Space Enable (MSE) bit in the device PCI configuration space and I/O decode is covered by the I/O Space Enable bit in the device PCI configuration space. The exceptions to this rule are the per port (external) I/OxAPIC address range and the internal I/OxAPIC ABAR address range which are decoded irrespective of the setting of the memory space enable bit. There is no decode enable bit for configuration cycle decoding towards either a PCI Express port or the internal configuration space of the IOH.
- The target decoding for internal VTdCSR space is based on whether the incoming CSR address is within the VTdCSR range.
- Each PCI Express/ESI port in the IOH has one special address range – I/OxAPIC.
- No loopback supported; that is, a transaction originating from a port is never sent back to the same port and the decode ranges of originating port are ignored in address decode calculations.

7.5.1.2 FWH Decoding

This section describes access to flash memory that is resident below the IOH.

7.5.1.2.1 Overview

- FWH accesses are allowed only from Intel QuickPath Interconnect. Accesses from JTAG, SMBus, and PCI Express are not permitted.
- The IOH does not allow boot from an ICH FWH that is not the legacy ICH FWH.
- The IOH indicates presence of bootable FWH to CPU if it is the IOH with a FWH that contains the boot code below the legacy ICH connected to it.
- All FWH addresses (4 GB:4 GB-16 MB) and 1 MB:1 MB-128K that do not positively decode to the IOH's PCI Express ports, are subtractively forwarded to its legacy decode port, if one exists (else, error).
- When the IOH receives a transaction from an Intel QuickPath Interconnect port within 4 GB:4 GB-16 MB or 1 MB:1 MB-128 K and there is no positive decode hit against any of the other valid targets (if there is a positive decode hit to any of the other valid targets, the transaction is sent to that target), then the transaction is forwarded to ESI if it is the subtractive decode port; otherwise it is aborted.



7.5.1.3 I/OxAPIC Decoding

I/OxAPIC accesses are allowed only from the Intel QuickPath Interconnect ports. The IOH provides an I/OxAPIC base/limit register per PCI Express port for decoding to I/OxAPIC in downstream components such as the PXH. The IOH's integrated I/OxAPIC decodes two separate base address registers, both targeting the same I/OxAPIC memory mapped registers. Decoding flow for transactions targeting I/OxAPIC addresses is the same as for any other memory-mapped I/O registers on PCI Express.

7.5.1.4 Other Outbound Target Decoding

Other address ranges that need to be decoded for each PCI Express and ESI port include the standard peer-to-peer bridge decode ranges (MMIOL, MMIOH, I/O, VGA config). Refer to *PCI-PCI Bridge 1.2 Specification* and *PCI Express Base Specification, Revision 2.0* for details.

7.5.1.5 Summary of Outbound Target Decoder Entries

Table 7-1, "Outbound Target Decoder Entries" provides a list of all the target decoder entries required by the outbound target decoder to positively decode towards a target.

Table 7-1. Outbound Target Decoder Entries

| Address Region | Target Decoder Entry | Comments |
|-------------------|----------------------|---|
| VGA (A0000-BFFFF) | 10 ¹ | Fixed |
| MMIOL | 10 | Variable. From P2P Bridge Configuration Register Space |
| I/OxAPIC | 10 | Variable. From P2P Bridge Configuration Register Space |
| MMIOH | 10 | Variable. From P2P Bridge Configuration Register Space |
| CFGBUS | 1 | IOH internal bus is fixed as bus 0 |
| | 11 | Variable. From P2P Bridge Configuration Register Space for PCIe bus number decode. |
| VTBAR | 1 | Variable: Decodes the Intel VT-d chipset registers. |
| ABAR | 1 | Variable. Decodes the sub-region within FEC address range for the integrated I/OxAPIC in IOH. |
| MBAR | 1 | Variable. Decodes any 32-bit base address for the integrated I/OxAPIC in IOH. |
| IO | 11 | Variable. From P2P Bridge Configuration Register Space of the PCIe port. |

Notes:

1. This is listed as 10 entries because each of the 10 P2P bridges have their own VGA decode enable bit and IOH has to comprehend this bit individually for each port.

7.5.1.6 Summary of Outbound Memory/IO decoding

Throughout the tables in this section, a reference to a PCIe port generically refers to a standard PCIe port or an ESI port.



Table 7-2. Decoding of Outbound Memory Requests from Intel QuickPath Interconnect (from CPU or Remote Peer-to-Peer)

| Address Range | Conditions | IOH Behavior |
|---------------------------|---|------------------------|
| I/OxAPIC BAR, ABAR, VTBAR | ABAR, MBAR, VTBAR and remote p2p access | Completer Abort |
| | ABAR, MBAR, VTBAR and not remote p2p access | Forward to that target |
| All memory accesses | (ABAR, MBAR) and one of the downstream ports positively claimed the address | Forward to that port |
| | (ABAR, MBAR) and none of the downstream ports positively claimed the address and ESI is the subtractive decode port | Forward to ESI |
| | (ABAR, MBAR) and none of the downstream ports positively claimed the address and ESI is not the subtractive decode port | Master Abort |

Table 7-3, “Subtractive Decoding of Outbound I/O Requests from Common System Interface” details IOH behavior when no target has been positively decoded for an incoming I/O transaction from Intel QuickPath Interconnect.

Table 7-3. Subtractive Decoding of Outbound I/O Requests from Common System Interface

| Address Range | Conditions | IOH Behavior |
|--|---|---|
| Any I/O address not positively decoded | No valid target decoded and one of the downstream ports is the subtractive decode port | Forward to downstream subtractive decode port |
| | No valid target decoded and none of the downstream ports is the subtractive decode port | Master Abort |

7.5.2 Inbound Address Decoding

This section covers the decoding that is done on any transaction that is received on a PCI Express or ESI port

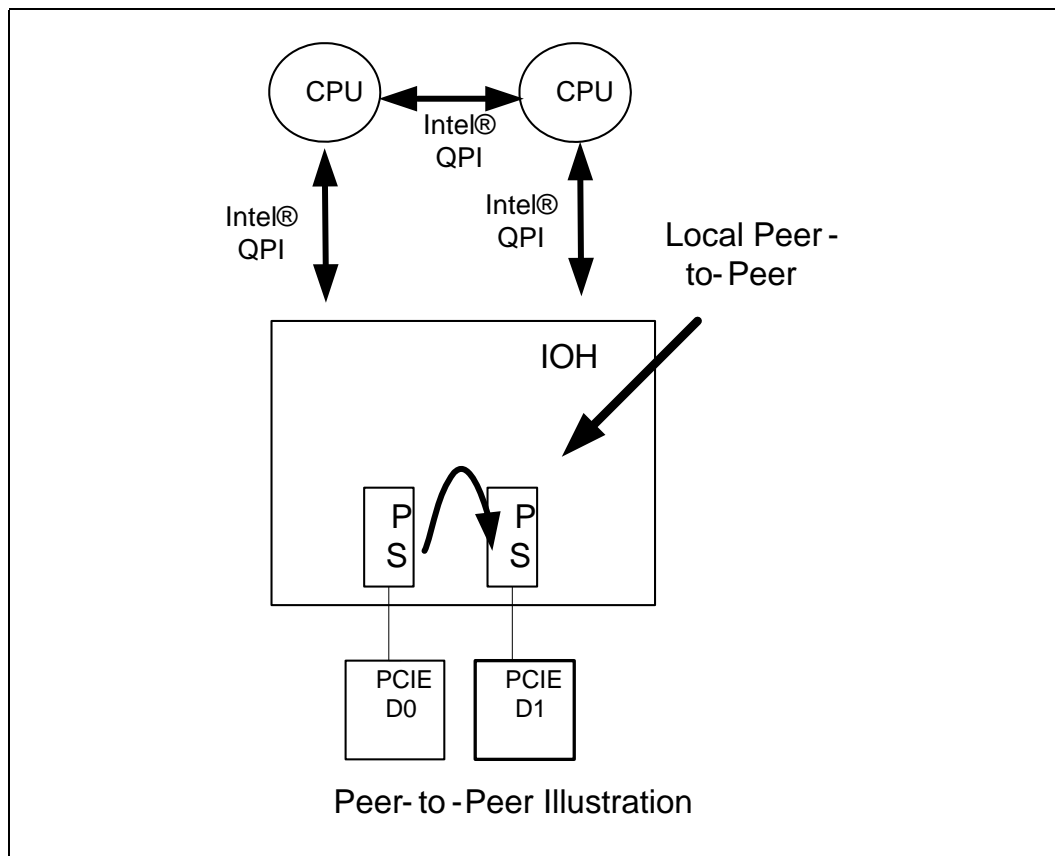
7.5.2.1 Overview

- All inbound addresses that fall above the top of Intel QuickPath Interconnect physical address limit are flagged as errors by the IOH.
- Inbound decoding towards main memory happens in two steps. The first step involves a ‘coarse decode’ towards main memory using two separate system memory window ranges (0-TOLM, 4 GB-TOHM) that can be setup by software. These ranges are non-overlapping. The second step is the fine source decode towards an individual processor socket using the Intel QuickPath Interconnect memory source address decoders.
 - A sub-region within one of the two coarse regions can be marked as non-coherent.
 - VGA memory address would overlap one of the two main memory ranges and the IOH decodes and forwards these addresses to the VGA device of the system.
- Inbound peer-to-peer decoding also happens in two steps. The first step involves decoding peer-to-peer crossing Intel QuickPath Interconnect (remote peer-to-peer) and peer-to-peer not crossing Intel QuickPath Interconnect (local peer-to-peer). The second step involves actual target decoding for local peer-to-peer (if transaction targets another device downstream from the IOH) and also involves

source decoding using Intel QuickPath Interconnect source address decoders for remote peer-to-peer.

- A pair of base/limit registers are provided to positively decode local peer-to-peer transactions. Another pair of base/limit registers are provided that covers the global peer-to-peer address range (that is, peer-to-peer address range of the entire system). Any inbound address that falls outside of the local peer-to-peer address range but that falls within the global peer-to-peer address range is considered as a remote peer-to-peer address.
- Fixed VGA memory addresses (A0000-BFFFF) are always peer-to-peer addresses and would reside outside of the global peer-to-peer memory address ranges mentioned above.
- Subtractively decoded inbound addresses are forwarded to the subtractive decode port of the IOH.

Figure 7-3. Peer-to-Peer Illustration



7.5.2.2 Summary of Inbound Address Decoding

Table 7-4 summarizes IOH behavior on inbound memory transactions from any PCI Express port. Note that this table is only intended to show the routing of transactions based on the address and is not intended to show the details of several control bits that govern forwarding of memory requests from a given PCI Express port. Refer to the *PCI Express Base Specification, Revision 2.0* and the registers chapter for details of these control bits.



Table 7-4. Inbound Memory Address Decoding

| Address Range | Conditions | IOH Behavior |
|--|---|--|
| DRAM | Address within 0: TOLM or 4GB: TOHM | Forward to Intel® QuickPath Interconnect port |
| Interrupts | Address within FEE00000-FEEFFFFFFF and write | Forward to Intel® QuickPath Interconnect port |
| | Address within FEE00000-FEEFFFFFFF and Read | Completer Abort |
| I/OxAPIC, CPUCSR, CPULocalCSR, privileged CSR, INTA/Rsvd, TSeg, Relocated CSeg, On-die ROM, FWH, VTBAR ¹ (when enabled), Protected VT-d range Low and High, Generic Protected dram range and I/OxAPIC BARs ² | FC00000-FEDFFFFFFF or FEF00000-FFFFFFF TOCM >= Address >= TOCM-64GB VTBAR VT-d_Prot_High VT-d_Prot_Low Generic_Prot_DRAM I/OxAPIC MBAR | Completer Abort |
| VGA | Address within 0A0000h-0BFFFFh and main switch SAD is programmed to forward VGA | Forward to Intel® QuickPath Interconnect port |
| | Address within 0A0000h-0BFFFFh and main switch SAD is NOT programmed to forward VGA and one of the PCIe has VGAEN bit set | Forward to the PCIe port |
| | Address within 0A0000h-0BFFFFh and main switch SAD is NOT programmed to forward VGA and none of the PCIe has VGAEN bit set and ESI port is the subtractive decoding port | Forward to ESI port |
| | Address within 0A0000h-0BFFFFh and main switch SAD is NOT programmed to forward VGA and none of the PCIe ports have VGAEN bit set and ESI is not the subtractive decode por | Forward to legacy IOH via Intel QuickPath using enabled SUBSAD; Master abort if SUBSAD is not enabled. |
| Other Peer-to-peer | Address within LMMIOL.BASE/LMMIOL.LIMIT or LMMIOH.BASE/LMMIOH.LIMIT and a PCIe port decoded as target | Forward to the PCI Express port |
| | Address within LMMIOL.BASE/LMMIOL.LIMIT or LMMIOH.BASE/LMMIOH.LIMIT and no PCIe port positively decoded as target ESI is the subtractive decoding port | Forward to ESI |
| | Address within LMMIOL.BASE/LMMIOL.LIMIT or LMMIOH.BASE/LMMIOH.LIMIT and no PCIe port decoded as target and ESI is not the subtractive decoding port | Master Abort |
| | Address NOT within LMMIOL.BASE/LMMIOL.LIMIT or LMMIOH.BASE/LIOH.LIMIT, but is within GMMIOL.BASE/GMMIOL.LIMIT or GMMIOH.BASE/GMMIOH.LIMIT | Forward to Intel QuickPath Interconnect |
| DRAM Memory holes and other non-existent regions | <ul style="list-style-type: none"> { 4G <= Address <= TOHM (OR) 0 <= Address <= TOLM } AND address does not decode to any socket in Intel QuickPath Interconnect source decoder Address > TOCM | Master Abort |
| All Else | | Forward to subtractive decode port, if enabled via CSRMISCCTRL[1], else Master Abort |

Notes:

- Note that VTBAR range would be within the MMIOL range of that IOH. And by that token, VTBAR range can never overlap with any dram ranges.
- The I/OxAPIC MBAR regions of an IOH overlap with MMIOL/MMIOH ranges of that IOH.

Table 7-5 summarizes IOH behavior on inbound memory transactions from any PCI Express port.



Table 7-5. Inbound I/O Address Decoding

| Address Range | Conditions | IOH Behavior |
|------------------------|---|--|
| Any | After disabling Inbound I/O ¹ | Master Abort |
| VGA | Address within 3B0h-3BBh, 3C0h-3DFh, inbound I/O is enabled and RVGAEN is set | Forward to Intel QuickPath Interconnect |
| | Address within 3B0h-3BBh, 3C0h-3DFh, inbound I/O is enabled and RVGAEN is NOT set and one of the PCIe has VGAEN bit set | Forward to that PCIe port |
| | Address within 3B0h-3BBh, 3C0h-3DFh, inbound I/O is enabled and RVGAEN is NOT set and none of the PCIe has VGAEN bit set but IS within the I/O base/limit range of one of the PCIe ports | Forward to that PCIe port |
| | Address within 3B0h-3BBh, 3C0h-3DFh, inbound I/O is enabled and RVGAEN is NOT set and none of the PCIe has VGAEN bit set and is NOT within the I/O base/limit range of any PCIe ports and ESI port is the subtractive decode port | Forward to ESI port |
| | Address within 03B0h-3BBh, 3C0h-3DFh, inbound I/O is enabled and RVGAEN is NOT set, none of the PCIe has VGAEN bit set and is NOT within base/limit range of any PCIe port, and ESI is not the subtractive decode port | Master abort |
| Other Peer-to-peer | Address within LIO.BASE/LIO.LIMIT, inbound I/O is enabled and a PCIe port positively decoded as target | Forward to the PCI Express port |
| | Address within LIO.BASE/LIO.LIMIT, inbound I/O is enabled and no PCIe port positively decoded as target and ESI is the subtractive decode port | Forward to ESI |
| | Address within LIO.BASE/LIO.LIMIT, inbound I/O is enabled and no PCIe port decoded as target and ESI is NOT the subtractive decode port | Master Abort |
| | Inbound I/O is enabled and address NOT within LIO.BASE/LIO.LIMIT, inbound I/O is enabled but is within GIO.BASE/GIO.LIMIT | Forward to the Intel® QuickPath Interconnect |
| Non-existent Addresses | Address => 64 KB | Master Abort |
| All Else | | Forward to subtractive decode port, if enabled via CSRmiscCTRL[1], else Master Abort |

Notes:

1. Inbound I/O is enabled or disabled via CSRmiscCTRLSTS[30].



7.6 Intel VT-d Address Map Implications

Intel VT-d applies only to inbound memory transactions. Inbound I/O and configuration transactions are not affected by Intel VT-d. Inbound I/O, configuration and message decode and forwarding happens the same whether Intel VT-d is enabled or not. For memory transaction decode, the host address map in Intel VT-d corresponds to the address map discussed earlier in the chapter and all addresses after translation are subject to the same address map rule checking (and error reporting) as in the non Intel VT-d mode. There is not a fixed guest address map that IOH Intel VT-d hardware can rely upon (except that the guest domain addresses cannot go beyond the guest address width specified via the GPA_LIMIT register) that is, it is OS dependent. IOH converts all incoming memory guest addresses to host addresses and then applies the same set of memory address decoding rules as described earlier. In addition to the address map and decoding rules discussed earlier, IOH also supports an additional memory range called the VTBAR range and this range is used to handle accesses to Intel VT-d related chipset registers. Only aligned DWORD/QWORD accesses are allowed to this region. Only outbound and SMBus/JTAG accesses are allowed to this range and also these can only be accesses outbound from Intel QuickPath Interconnect. *Inbound accesses to this address range are completely aborted by the IOH.*

§





8 Interrupts

8.1 Overview

The IOH supports both MSI and legacy PCI interrupts from its PCI Express ports. MSI interrupts received from PCI Express are forwarded directly to the processor socket. Legacy interrupt messages received from PCI Express are either converted to MSI interrupts via the integrated I/OxAPIC in the IOH or forwarded to the ESI. When the legacy interrupts are forwarded to ESI, the compatibility bridge either converts the legacy interrupts to MSI writes via its integrated I/OxAPIC or handles them via the legacy 8259 controller. All root port interrupt sources within the IOH (that is, Error and Power management) support MSI mode interrupt delivery. Where noted, these interrupt sources (except the error source) also support the ACPI-based mechanism (via GPE messages) for system driver notification. The IOH does not support legacy PCI INTx mechanism for internal sources of interrupt. In addition to MSI and ACPI messages, the IOH also supports generation of SMI/PMI/MCA/NMI interrupts directly from the IOH to the processor (bypassing ICH), in support of IOH error reporting. For Intel QuickPath Interconnect-defined legacy virtual message Virtual Legacy Wires (VLW) signaling, the IOH provides a sideband interface to the legacy bridge and an inband interface on Intel QuickPath Interconnect. The IOH logic handles conversion between the two.

8.2 Legacy PCI Interrupt Handling

On PCI Express, interrupts are represented with either MSI or inbound interrupt messages (Assert_INTx/Deassert_INTx). The integrated I/OxAPIC in the IOH converts the legacy interrupt messages received from PCI Express into MSI interrupts. If the I/OxAPIC is disabled (via the mask bits in the I/OxAPIC table entries), the messages are routed to the legacy ICH. The subsequent paragraphs describe how the IOH handles the INTx message flow, from its PCI Express ports and internal devices.

The IOH tracks the assert/deassert messages for the four interrupts INTA, INTB, INTC, and INTD from each PCI Express port and INTx for CB DMA and ME (Intel Xeon processor 7500 series-based platform only). Each of these interrupts from each PCI Express root port is routed to a specific I/OxAPIC table entry (see [Table 8-2](#) for the mapping) in that IOH. If the I/OxAPIC entry is masked (via the 'mask' bit in the corresponding Redirection Table Entry), then the corresponding PCI Express interrupt(s) is forwarded to the legacy ICH, provided the 'Disable PCI INTx Routing to ICH' bit is clear, [Section 21.11.2.28, "QPIPINTRC: Intel QuickPath Interconnect Protocol Interrupt Control"](#) .

There is a 1:1 correspondence between message type received from PCI Express and the message type forwarded to the legacy ICH. For example, if the PCI Express Port 0 INTA message is masked in the integrated I/OxAPIC, it is forwarded to the legacy ICH as INTA message (if the 'Disable Interrupt Routing to ICH' bit is cleared). Each IOH combines legacy interrupts (to be forwarded to the legacy ICH) from all PCI Express ports and CB DMA (Intel Xeon processor 7500 series based platform only) and presents a consolidated set of four virtual wire messages. If the I/OxAPIC entry is unmasked, an MSI interrupt message is generated on the Intel QuickPath Interconnect.

The IOH does not provide a capability to route inband PCI INTx virtual wire messages to any component other than the legacy ICH.

When a standard downstream PCI Express root port receives an Assert_INTx message, subsequent Assert_INTx messages of the same type (A/B/C/D) will simply keep the virtual wire asserted until the associated Deassert message is received. The first Deassert message received for a given interrupt type will deassert the internal virtual wire of the root port for the interrupt type. Also the internal virtual wire of the root port is de-asserted automatically in hardware if the link goes down when the internal virtual wire is asserted. Deassert messages received for a given interrupt when no corresponding Assert message was received previously for that interrupt, or Deassert messages received when no virtual wire for that interrupt is asserted, will be discarded with no side effect. On an Intel QuickPath Interconnect link port, the IOH can receive multiple Assert_INTx messages of the same type before it receives any Deassert_INTx message of that type. In normal operation, it is always guaranteed that the IOH will receive a Deassert_INTA message for every Assert_INTA message it receives from the Intel QuickPath Interconnect.

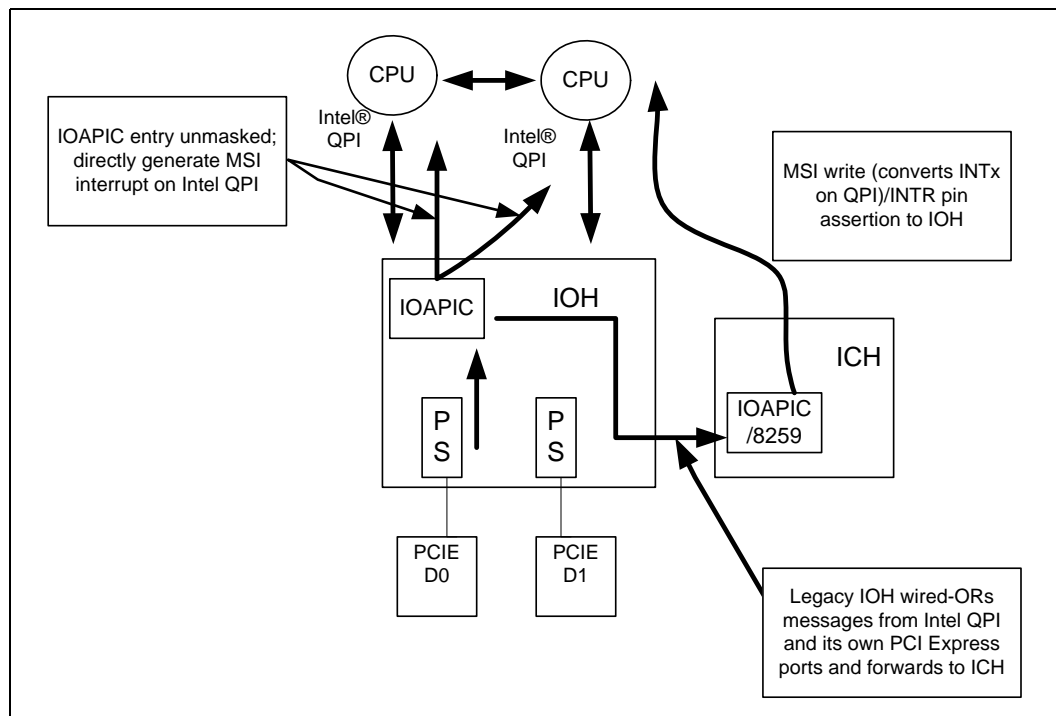
For consolidating interrupts from the PCI Express/ESI ports in non-legacy IOHs, the consolidated set of interrupt messages are routed to the Intel QuickPath Interconnect instead.

8.2.1 Summary of PCI Express INTx Message Routing

An IOH is not always guaranteed to have its ESI port enabled for legacy. When an IOH's ESI port is disabled, it has to route the INTx messages it receives from its downstream PCI Express ports to its the Intel QuickPath Interconnect interface, provided they are not serviced via the integrated I/OxAPIC.

Figure 8-1 illustrates how legacy interrupt messages are routed to the legacy ICH.

Figure 8-1. Legacy Interrupt Routing Illustration (INTA Example)





8.2.2 Integrated I/OxAPIC

The integrated I/OxAPIC converts legacy PCI Express interrupt messages into MSI interrupts. The I/OxAPIC appears as a PCI Express endpoint device in the IOH configuration space. The I/OxAPIC provides 24 unique MSI interrupts. This table is programmed via the MBAR memory region or ABAR memory region (Refer to [Chapter 21](#)).

In Dual IOH Proxy mode, the local IOH uses the Dual NonLegacyIOH ABAR range as defined by “DUAL.NL.ABAR.BASE: Dual NonLegacyIOH ABAR Range Base” and “DUAL.NL.ABAR.LIMIT: Dual NonLegacyIOH ABAR Range Limit” to send transaction targeting at non-legacy IOH’s integrated IOAPIC controller to the non-legacy IOH.

These registers in both legacy and non-legacy IOH need to be programmed with the range claimed by the non-legacy IOH.

In the non-Legacy IOH with 11 PCI Express ports (instead of ESI in the legacy IOH), there are potentially 53 unique legacy interrupts possible, 11 PCIe ports * 4 (sources #1 - #11) + 4 for CB DMA (source #12) + 4 for ME (sources #13 - #16) + 1 IOH RootPorts/core (source #17) as shown in [Table 8-1, “Interrupt Sources in I/OxAPIC Table Mapping”](#), and these are mapped to the 24 entries in the I/OxAPIC as shown in [Table 8-2, “I/OxAPIC Table Mapping to PCI Express Interrupts”](#).

In the legacy IOH, there are 49 unique legacy interrupts possible which are mapped to the 24 entries in the I/OxAPIC, as shown in [Table 8-2](#). The distribution is based on guaranteeing that there is at least one unshared interrupt line (INTA) for each possible source of interrupt. When a legacy interrupt asserts, an MSI interrupt is generated (if the corresponding I/OxAPIC entry is unmasked) based on the information programmed in the corresponding I/OxAPIC table entry.

Table 8-1. Interrupt Sources in I/OxAPIC Table Mapping

| Interrupt Source # | PCI Express Port/Device | INT[A-D] Used / Comment |
|--------------------|---|-------------------------|
| 1 | PCIe port 3 | A,B,C,D / x16, x8, x4 |
| 2 | PCIe port 4 | A,B,C,D / x4 |
| 3 | PCIe port 5 | A,B,C,D / x8, x4 |
| 4 | PCIe port 6 | A,B,C,D / x4 |
| 5 | PCIe port 1 | A,B,C,D / x4, x2 |
| 6 | PCIe port 2 | A,B,C,D / x2 |
| 7 | PCIe port 7 | A,B,C,D / x16, x8, x4 |
| 8 | PCIe port 8 | A,B,C,D / x4 |
| 9 | PCIe port 9 | A,B,C,D / x8, x4 |
| 10 | PCIe port 10 | A,B,C,D / x4 |
| 11 | PCIe port 11 (ESI used as PCIe in non-legacy IOH) | A,B,C,D / x4 |
| 12 | CB3 DMA | A,B,C,D |
| 13 | Intel ME HEC1 | A |
| 14 | Intel ME HEC12 | D |
| 15 | Intel ME IDEr | C |
| 16 | Intel ME KT | B |
| 17 | Root Ports/Core | A |



Table 8-2. I/OxAPIC Table Mapping to PCI Express Interrupts¹

| I/OxAPIC Table Entry# | PCI Express Port/CB DMA Device# | PCI Express Virtual Wire Type |
|-----------------------|---------------------------------|-------------------------------|
| 0 | 1 | INTA |
| 1 | 2, <14>, 3 | INTA, <INTA>, [INTB] |
| 2 | 3, <15> | INTA, <INTC> |
| 3 | 4, <16>, {3} | INTA, <INTB>, {INTC} |
| 4 | 5 | INTA |
| 5 | 6, <17>, [3] | INTA, <INTA>, [INTD] |
| 6 | 7 | INTA |
| 7 | 8, <10> | INTA, <INTB> |
| 8 | 9 | INTA |
| 9 | 10 | INTA |
| 10 | 1, <2>, [3] | INTB, <INTD>, [INTC] |
| 11 | 1, <2>, [3] | INTC, <INTB>, [INTD] |
| 12 | 1, <2>, [3] | INTD, <INTC>, [INTB] |
| 13 | 7, <8>, [4] | INTB, <INTD>, [INTC] |
| 14 | 7, <8>, [4] | INTD, <INTC>, [INTB] |
| 15 | 7, <8>, [4] | INTC, <INTB>, [INTD] |
| 16 | 5, <10>, [6], {9} | INTB, <INTD>, [INTC], {INTC} |
| 17 | 5, <10>, [6], {9} | INTC, <INTB>, [INTB], {INTD} |
| 18 | 5, <10>, [6], {9} | INTD, <INTC>, [INTD], {INTB} |
| 19 | 12 | INTA |
| 20 | 12 | INTB |
| 21 | 12, <15>, [17], {10} | INTC, <INTC>, [INTA], {INTD} |
| 22 | 12, <14>, [16], {10} | INTD, <INTA>, [INTB], {INTC} |
| 23 | 13, <5>, [9], {6} | INTA, <INTD>, [INTC], {INTB} |

Notes:

1. < >, [], and { } associate interrupt from a given device number (as shown in the 'PCI Express Port/CB DMA Device#' column) that is marked thus to the corresponding interrupt wire type (shown in this column) also marked such. For example, I/OxAPIC entry 12 corresponds to the wired-OR of INTD message from source #1 (PCIe port #3), INTC message from source #2 (PCIe port #4), and INTB message from source #3 (PCIe port #5).

Table 8-3. Programmable IOxAPIC Entry Target for Certain Interrupt Sources (Sheet 1 of 2)

| Target Table Entries Numbers | Interrupt Source in Table 8-1 | INT[A-D] | Default Table Entry (3x8, 3x4) |
|------------------------------|-------------------------------|----------|--------------------------------|
| 1, 12 | 3 | INTB | 1 |
| 3, 10 | 3 | INTC | 3 |
| 5, 11 | 3 | INTD | 5 |
| 18, 23 | 5 | INTD | 23 |
| 17, 23 | 6 | INTB | 17 |
| 16, 23 | 9 | INTC | 23 |
| 7, 17 | 10 | INTB | 7 |



Table 8-3. Programmable IOxAPIC Entry Target for Certain Interrupt Sources (Sheet 2 of 2)

| Target Table Entries Numbers | Interrupt Source in Table 8-1 | INT[A-D] | Default Table Entry (3x8, 3x4) |
|------------------------------|-------------------------------|----------|--------------------------------|
| 18, 22 | 10 | INTC | 22 |
| 16, 21 | 10 | INTD | 21 |
| 1, 22 | 14 | INTD | 22 |
| 2, 21 | 13 | INTA | 21 |
| 3, 22 | 16 | INTB | 22 |
| 5, 21 | 17 | INTA | 21 |

8.2.2.1 Integrated I/OxAPIC MSI Interrupt Ordering

As with MSI interrupts generated from PCI Express endpoints, MSI interrupts generated from the integrated I/OxAPIC follow the RdC push memory write ordering rule. For example read completions on reads (config or memory) to I/OxAPIC registers must push previously posted MSI writes from the I/OxAPIC.

8.2.2.2 Integrated I/OxAPIC EOI Flow

Each I/OxAPIC entry can be setup by software to treat the interrupt inputs as either level or edge triggered. For level triggered interrupts, the I/OxAPIC generates an interrupt when the interrupt input asserts, and stops generating further interrupts until software clears the RIRR bit in the corresponding redirection table entry with a directed write to the EOI register; or until software generates an EOI message to the I/OxAPIC with the appropriate vector number in the message. When the RIRR bit is cleared, the I/OxAPIC resamples the level interrupt input corresponding to the entry; if it is still asserted, the I/OxAPIC generates a new MSI message.

The EOI message is broadcast to all I/OxAPICs in the system; the integrated I/OxAPIC is also a target for the EOI message. The I/OxAPIC looks at the vector number in the message, and the RIRR bit is cleared in all the I/OxAPIC entries which have a matching vector number.

IOH has capability to NOT broadcast/multicast EOI message to any of the PCI Express/ ESI ports/ integrated IOxAPIC and this is controlled via bit 0 in the EOI_CTRL register. When this bit is set, IOH simply drops the EOI message received from Intel QuickPath Interconnect and not send it to any south agent. But IOH does send a normal cmp for the message on Intel QuickPath Interconnect. This is required in some virtualization usages.

8.2.3 PCI Express INTx Message Ordering

INTx messages on PCI Express are posted transactions and follow the posted ordering rules. For example, if an INTx message is preceded by a memory write A, the INTx message pushes the memory write to a global ordering point before the INTx message is delivered to its destination (which could be the I/OxAPIC, which decides further action). This guarantees that any MSI generated from the integrated I/OxAPIC (or from the I/OxAPIC in ICH, if the integrated I/OxAPIC is disabled) will be ordered behind the memory write A, guaranteeing producer/consumer sanity.



8.2.4 INTR_Ack/INTR_Ack_Reply Messages

INTR_Ack and INTR_Ack_Reply messages on ESI and IntAck on Intel QuickPath Interconnect support legacy 8259-style interrupts required for system boot operations. These messages are routed from the processor socket to the legacy IOH via the IntAck cycle on Intel QuickPath Interconnect. The IntAck transaction issued by the processor socket behaves as an I/O Read cycle in that the Completion for the IntAck message contains the Interrupt vector. The IOH converts this cycle to a posted message on the ESI port (no completions).

- IntAck – The IOH forwards the IntAck received on the Intel QuickPath Interconnect interface (as an NCS transaction) as a posted INTR_Ack message to the legacy ICH over ESI. A completion for IntAck is not sent on Intel QuickPath Interconnect just yet.
- INTR_Ack_Reply – The ICH returns the 8-bit interrupt vector from the 8259 controller through this posted vendor defined message (VDM). The INTR_Ack_Reply message pushes upstream writes through virtual channel (VC0) in both the ICH and the IOH. This IOH then uses the data in the INTR_Ack_Reply message to form the completion for the original IntAck message.

Note: There can be only one outstanding IntAck transaction across all processor sockets in a partition at a given instance.

8.3 MSI

MSI interrupts generated from PCI Express ports or from integrated functions within the IOH are memory writes to a specific address range, 0xFEEEx_xxxx. If interrupt remapping is disabled in the IOH, the interrupt write directly provides the information regarding the interrupt destination processor and interrupt vector. The details of these are as shown in [Table 8-4](#) and [Table 8-5](#). If interrupt remapping is enabled in the IOH, interrupt write fields are interpreted as shown in [Table 8-6](#) and [Table 8-7](#).

Note: The term APICID in this chapter refers to the 32-bit field on Intel QuickPath Interconnect interrupt packets, in both the format and meaning.



Table 8-4. MSI Address Format when Remapping is Disabled

| Bits | Description |
|-------|---|
| 31:20 | FEEh |
| 19:12 | <p>Destination ID: This will be the bits [63:56] of the I/O Redirection Table entry for the interrupt associated with this message.</p> <p>This field directly identifies the interrupt target in IA-32 mode. This field along with EID field identifies the interrupt target in Intel Itanium processor 9300 series mode.</p> <p>In IA-32 mode:</p> <p>For physical mode interrupts, this field becomes APICID[7:0] on the Intel QuickPath Interconnect interrupt packet and APICID[31:8] are reserved in the Intel QuickPath Interconnect packet.</p> <p>For logical cluster mode interrupts, [19:16] of this field becomes APICID[19:16] on the Intel QuickPath Interconnect interrupt packet and [15:12] of this field becomes APICID[3:0] on the Intel QuickPath Interconnect interrupt packet.</p> <p>For logical flat mode interrupts, [19:12] of this field becomes APICID[7:0] on the Intel QuickPath Interconnect interrupt packet.</p> <p>In Intel Itanium processor 9300 series mode:</p> <p>This field becomes APICID[15:8] of the Intel QuickPath Interconnect interrupt packet.</p> |
| 11:4 | <p>EID: This will be the bits [55:48] of the I/O Redirection Table entry for the interrupt associated with this message.</p> <p>When interrupt remapping is disabled, this field is applicable only for Intel Itanium processor 9300 series and this field along with Destination ID identifies the interrupt target in Intel Itanium processor 9300 series mode. In Intel Itanium processor 9300 series mode, this field becomes APICID[7:0] on the Intel QuickPath Interconnect interrupt packet and APICID[31:16] are reserved in the Intel QuickPath Interconnect packet.</p> |
| 3 | <p>Redirection Hint: This bit allows the interrupt message to be directed to one among many targets, based on chipset redirection algorithm.</p> <p>0 = The message will be delivered to the agent (CPU) listed in bits [19:4]</p> <p>1 = The message will be delivered to an agent based on the IOH redirection algorithm and the scope the interrupt as specified in the interrupt address.</p> <p>The Redirection Hint bit will be a 1 if bits [10:8] in the Delivery Mode field associated with corresponding interrupt are encoded as 001b (Lowest Priority). Otherwise, the Redirection Hint bit will be 0.</p> |
| 2 | <p>Destination Mode: This is the corresponding bit from the I/O Redirection Table entry. 1=logical mode and 0=physical mode. This bit determines if IntLogical or IntPhysical is used on Intel QuickPath Interconnect.</p> |
| 1:0 | 00 |

Table 8-5. MSI Data Format when Remapping Disabled

| Bits | Description |
|-------|---|
| 31:16 | 0000h |
| 15 | Trigger Mode: 1 = Level, 0 = Edge. Same as the corresponding bit in the I/O Redirection Table for that interrupt. |
| 14 | Delivery Status: Always set to 1, that is, asserted |
| 13:12 | 00 |
| 11 | <p>Destination Mode: This is the corresponding bit from the I/O Redirection Table entry. 1=logical mode and 0=physical mode.</p> <p>Note that this bit is set to 0 before being forwarded to Intel QuickPath Interconnect.</p> |
| 10:8 | Delivery Mode: This is the same as the corresponding bits in the I/O Redirection Table for that interrupt. |
| 7:0 | Vector: This is the same as the corresponding bits in the I/O Redirection Table for that interrupt. |

Table 8-6. MSI Address Format when Remapping is Enabled

| Bits | Description |
|-------|--|
| 31:20 | FEEh |
| 19:4 | Interrupt Handle: IOH looks up an interrupt remapping table in main memory using this field as an offset into the table |
| 3 | Sub Handle Valid: When IOH looks up the interrupt remapping table in main memory, and if this bit is set, IOH adds the bits 15:0 from interrupt data field to interrupt handle value (bit 19:4 above) to obtain the final offset into the remapping table. If this bit is clear, Interrupt Handle field directly becomes the offset into the remapping table. |
| 2 | Reserved: IOH hardware ignores this bit |
| 1:0 | 00 |

Table 8-7. MSI Data Format when Remapping is Enabled

| Bits | Description |
|-------|---|
| 31:16 | Reserved – IOH hardware checks for this field to be 0 (note that this checking is done only when remapping is enabled) |
| 15:0 | Sub Handle |

All PCI Express devices are required to support MSI. The IOH converts memory writes to this address (both PCI Express and internal sources) as an IntLogical or IntPhysical transaction on Intel QuickPath Interconnect. The IOH supports two MSI vectors per root port for hot-plug, power management, and error reporting.

8.3.1 Interrupt Remapping

Interrupt remapping architecture serves two purposes:

- Provide for interrupt filtering for virtualization/security usages so that an arbitrary device cannot interrupt an arbitrary processor in the system
- Provide for IO devices to target greater than 255 processors as part of extended xAPIC architecture

Software can use interrupt remapping for either or both of the reasons above. When interrupt remapping is enabled in the IOH, IOH looks up a table in main memory to obtain the interrupt target processor and vector number. When the IOH receives an MSI interrupt (where MSI interrupt is any memory write interrupt directly generated by an IO device or generated by an I/OxAPIC like the integrated I/OxAPIC in the IOH/ICH/PXH) and the remapping is turned on, IOH picks up the ‘interrupt handle’ field from the MSI (bits [19:4] of the MSI address) and adds it to the Sub Handle field in the MSI data field if Sub Handle Valid field in MSI address is set, to obtain the final interrupt handle value. The final interrupt handle value is then used as an offset into the table in main memory as,

$$\text{Memory Offset} = \text{Final Interrupt Handle} * 16$$

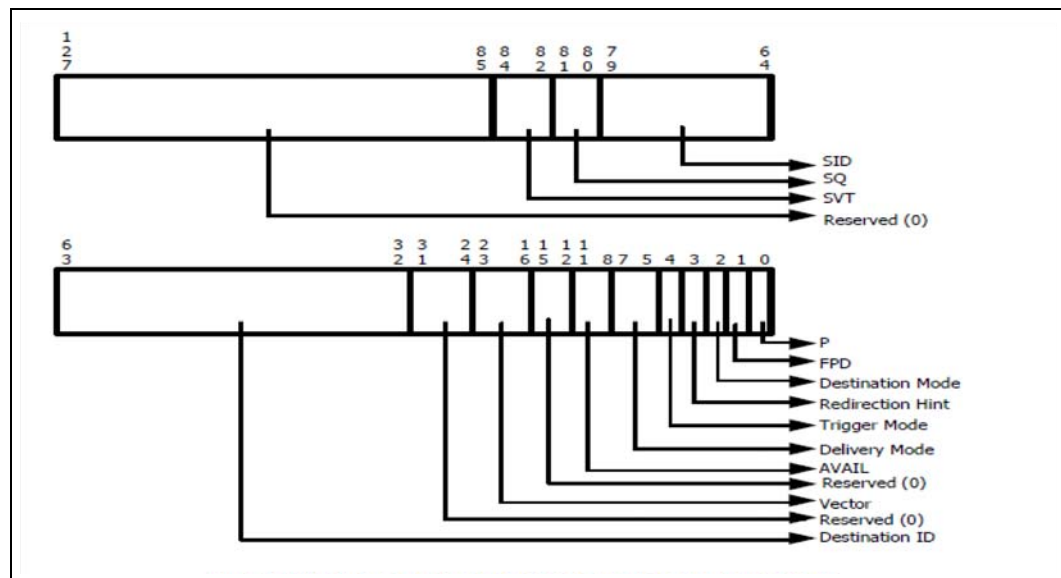
where Final Interrupt Handle = if (Sub Handle Valid = 1) then { Interrupt Handle + Sub Handle} else Interrupt handle.

The data obtained from the memory lookup is called Interrupt Transformation Table Entry (IRTE) and is as follows:



As can be seen, all the information that used to be obtained directly from the MSI address/data fields are now obtained via the IRTE when remapping is turned on. In addition, the IRTE also provides for a way to authenticate an interrupt via the Requester ID, that is, the IOH needs to compare the Requester ID in the original MSI interrupt packet (that triggered the lookup) with the Requester ID indicated in the IRTE. If it matches, the interrupt is further processed, else the interrupt is dropped and error signaled. Subsequent sections in this chapter describe how the various fields in either the IRTE (when remapping enabled) or MSI address/data (when remapping disabled) are used by the chipset to generate IntPhysical/Logical interrupts on the Intel QuickPath Interconnect.

Figure 8-2. Interrupt Transformation Table Entry (IRTE)



The Destination ID shown in the picture above becomes the APICID on the Intel QuickPath Interconnect interrupt packet.

8.3.2 MSI Forwarding: IA-32 Processor-based Platform

IA-32 interrupts have two modes – legacy mode and extended mode (selected via bit). Legacy mode has been supported in all Intel chipsets to date. Extended mode allows for scaling beyond 60/255 threads in logical/physical mode operation. Legacy mode has only 8-bit APICID support. Extended mode supports 32-bit APICID (obtained via the IRTE).

Table 8-8 summarizes interrupt delivery for IA-32 processor based platforms. Table 8-9 summarizes how the IOH derives the processor NodeID at which the interrupt is targeted.

Table 8-8. Interrupt Delivery

| Mode | Sub-mode | Target APIC | NodeID Determined by IOH | IOH Behavior |
|----------|------------|---------------------------------------|--------------------------|--|
| Physical | Directed | APIC identified in APIC ID:EID fields | NodeID per Table 8-9 | Send IntPhysical to selected Intel QuickPath Interconnect NodeID |
| | Redirected | APIC identified in APIC ID:EID | NodeID per Table 8-9 | |



Table 8-9. IA-32 Physical APICID to NodeID Mapping

| Size | EID ¹ (MSI Addr 11:4) | APIC ID ² (MSI Addr 19:12) | NodeID[5:0] |
|--------|-------------------------------------|--|--|
| 1-16S | DC | abnn ³ 00cc ⁴ | yyy ⁵ 10 ⁶ / \langle NCNODEID \rangle ⁷ |
| 17-32S | DC | 0aAbnncc | |
| 33-64S | DC | aaabnncc | |

Notes:

1. These bits must be set to 0 before forwarding to Intel QuickPath Interconnect.
2. Note that IOH hardware would pass these bits untouched from the interrupt source to Intel QuickPath Interconnect.
3. a, b represent the bits that are used to compare against the mask register to identify local versus remote clusters for interrupt routing. b is optionally included in the mask based on whether 4S clusters or 8S clusters are used for scaleup granularity.
4. cc in the table above refers to the core number in TW and IOH does not do anything with that value.
5. yyy is an arbitrary number that is looked up from a table using bits bnn of the APIC ID field. This flexible mapping is provided for CPU migration and also to prevent software in one partition send interrupts to software in another partition.
6. The 10 value in the NodeID field is points to the config agent in the CPU. Note that this value is programmable as well for the table as a whole.
7. When the mask bits match mask register value, the interrupt is considered local and is directed to the socket with NodeID=yyy10. Otherwise the interrupt is remote and is routed to the node controller whose NodeID=NCNODEID, as derived from bits 28:24 of QPIPIASAD register.

Listed below are some basic assumptions around MSI forwarding:

- Processors indicated in the APICID field of the interrupt address (except in the IA-32 broadcast/multicast interrupts) are all valid and enabled for receiving an interrupt. IOH does not maintain a vector of enabled APICs for interrupt redirection purposes.
- Redirected broadcast physical and logical cluster mode interrupts (that is, redirection hint bit being 1b and {legacy_mode \langle APICID=0xFF in physical mode and APICID[7:4]=0xF in logical cluster mode \rangle OR extended_mode \langle APICID=0xFFFFFFFF in physical mode and APICID[31:16]=0xFFFF in logical cluster mode \rangle }) are not supported. IOH reports error if it receives one.
- Redirected broadcast flat mode interrupts (in legacy mode only) are supported
- Physical mode APICID and Extended Logical APICID have a direct correlation with the Intel QuickPath Interconnect NodeID and this relationship is setup by bios. OS never re-assigns the physical mode APICID or the extended mode logical ID.
 - IOH provides an ability to override the default relationship to support RAS features like OS-transparent migration, and so on. The overriding effect can be achieved either via the interrupt SAD entry (QPIPIASAD) that can pick a (limited) arbitrary relationship between Intel QuickPath Interconnect NodeID and Physical APICID (or) via broadcasting physical mode interrupts through BIOS setup (default is to not broadcast).

IOH supports the IntPriUpd message on Intel QuickPath Interconnect in order to know if the system is operating in logical flat or logical cluster mode. Even though this information is relevant for only legacy IA-32 interrupt mode, IOH, if it receives this message, will always extract this information from the message and update bit 1 in QPIPINTRC register. In modes other than IA-32 legacy mode, this register bit goes unused. This additional information outlines how this bit is set by the IOH. The IOH defaults to flat mode. On every IntPriUpd message, IOH samples the flat/cluster mode bit in the message provided the APIC generating the message is indicated as enabled (that is, “disable” bit in the IntPriUpd message data field should be cleared) in the message AND the APICID field in the message is non-zero (the latter check is performed only when QPIPINTRC[0] bit is 1). Once the bit is sampled, IOH updates QPIPINTRC[1] bit that tracks the flat/cluster mode for interrupt redirection purposes.



8.3.2.1 Legacy Logical Mode Interrupts

IA-32 legacy logical interrupts are broadcast by IOH to all processors in the system and it is the responsibility of the CPU to drop interrupts that are not directed to one of its local APICs. IOH supports hardware redirection for IA-32 logical interrupts (see [Section 8.3.2.2](#)) and in IOH-based platforms this is the only hardware redirection that is available in the system since the processor never does any internal redirection of these interrupts. IOH always clears the redirection hint bit on Intel QuickPath Interconnect when forwarding legacy logical mode interrupts. For IA-32 logical interrupts, no fixed mapping is guaranteed between the NodeID and the APICID since APICID is allocated by the OS and it has no notion of Intel QuickPath Interconnect NodeID. Again the assumption is made that APICID field in the MSI address only includes valid/enabled APICs for that interrupt. Refer to [Table 8-10](#) for summary of IA-32 interrupt handling by IOH.

8.3.2.2 Legacy Logical Mode Interrupt Redirection – Vector Number Redirection

In the logical flat mode when redirection is enabled, IOH looks at the bits [6:4] (or 5:3/3:1/2:0 based on bits 4:3 of QPIPINTRC register) of the interrupt vector number and picks the APIC in the bit position (in the APICID field of the MSI address) that corresponds to the vector number. For example, if vector number[6:4] is 010, then the APIC correspond to MSI Address APICID[2] is selected as the target of redirection. If vector number[6:4] is 111, then the APIC correspond to APICID[7] is selected as the target of redirection. If the corresponding bit in the MSI address is clear in the received MSI interrupt, then,

- IOH adds a value of 4 to the selected APIC's address bit location and if the APIC corresponding to modulo eight of that value is also not a valid target (because the bit mask corresponding to that APIC is clear in the MSI address), then,
- IOH adds a value of 2 to the original selected APIC's address bit location and if the APIC corresponding to modulo eight of that value is also not a valid target, then IOH adds a value of 4 to the previous value, takes the modulo 8 of the resulting value and if that corresponding APIC is also not a valid target, then,
- IOH adds a value of 3 to the original selected APIC's address bit location and if the APIC corresponding to modulo eight of that value is also not a valid target, then IOH adds a value of 4 to the previous value, takes the modulo 8 of the resulting value and if that corresponding APIC is also not a valid target, then,
- IOH adds a value of 1 to the original selected APIC's address bit location and if the APIC corresponding to modulo eight of that value is also not a valid target, then IOH adds a value of 4 to the previous value, takes the modulo 8 of the resulting value and if that corresponding APIC is also not a valid target, then it is an error condition

In the logical cluster mode (except when APICID[19:16] != Fh), the redirection algorithm works exactly as described above except that IOH only redirects between 4 APICs instead of 8 in the flat mode. So IOH uses only vector number bits [5:4] by default (selectable to 4:3/2:1/1:0 based on bits 4:3 of QPIPINTRC: [Intel QuickPath Interconnect Protocol Interrupt Control](#) register). The search algorithm to identify a valid APIC for redirection in the cluster mode is to:

- First select the APIC corresponding to the bit position identified with the chosen vector number bits. If the corresponding bit in the MSI address bits A[15:12] is clear, then,
- IOH adds a value of 2 to the original selected APIC's address bit location and if the APIC corresponding to modulo four of that value is also not a valid target, then,



- IOH adds a value of 1 to the original selected APIC's address bit location and if the APIC corresponding to modulo four of that value is also not a valid target, then IOH adds a value of 2 to the previous value, takes the modulo four of the resulting value and if that corresponding APIC is also not a valid target, then it is an error condition.

8.3.2.3 Legacy Logical Mode Interrupt Redirection – Round-Robin Redirection

IOH also supports a mode where the vector-based redirection is disabled and a simple round-robin mode is selected for redirection between the cores/APICs. In the logical flat mode, redirection is done in a round-robin fashion across the cores that are enabled via the corresponding mask bit in the interrupt address (max possible 8 enabled cores). In the logical cluster mode (except when cluster id = Fh), IOH maintains round-robin logic per cluster (max 15 clusters) and within each cluster IOH round-robins amongst the valid APICs.

Round-robin algorithm defaults at power-on to starting from the LSB (in the bit mask) and moving towards the MSB.

8.3.2.4 Physical Mode Interrupts and Extended Logical Cluster Mode Interrupts

By default, IA-32 physical interrupts and IA-32 extended logical cluster mode interrupts are directed to the correct socket by IOH, with the exception of the physical/extended cluster mode broadcast interrupts. Legacy physical mode broadcast interrupts are ones with APICID[7:0]=0xFF. Extended physical mode broadcast interrupts are ones with APICID[31:0]=0xFFFFFFFF. Extended cluster mode broadcast interrupts are ones with APICID[31:16]=0xFFFF. The default non-broadcast behavior can be changed via QPIPINTCR[6] where even these interrupts are broadcast.

IOH does not perform any hardware redirection of IA-32 physical mode interrupts (legacy or extended). IOH simply forwards the RH bit on these interrupts, except when these interrupts are broadcast (via bit QPIPINTCR[6]) as well, when the RH bit is cleared for legacy mode. IOH performs redirection of extended cluster mode interrupts as described in [Section 8.3.2.2](#) and [Section 8.3.2.3](#), with the additional detail that there can be up to 8 processors within a cluster in the extended mode. So the vector-based algorithm and round-robin algorithm described in these sections should be extended to up to 8 targets within a cluster for supporting extended mode. When IOH does the extended cluster mode redirection, IOH clears the RH bit (in addition to setting the mask bit corresponding to only the selected processor) before the interrupt is forwarded to the Intel® QuickPath Interconnect. Redirection of extended cluster mode interrupt can be disabled via bit 2 in QPIPINTCR. When extended cluster mode redirection is disabled, IOH simply forwards the interrupt as is to the Intel® QuickPath Interconnect (regardless of whether the interrupt is routed or broadcast)- including the RH bit.

In IA-32 physical mode (legacy or extended) and extended cluster mode, the mapping between APICID and NodeID is obtained as described in the QPIPAPICSAD register, with one exception when the Physical APICID field has a value of <0xFF (in legacy mode) or 0xFFFFFFFF (in extended mode)> or the extended logical cluster ID had a value of 0xFFFF, which indicate broadcast. Also, in the IA-32 physical mode, the APIC identified in the APICID field is always valid excluding the exception (broadcast) case. Note that QPIPAPICSAD register also provides details about how physical/extended-logical interrupt routing happens in hierarchical systems with node controller. For these systems, interrupts need to be either routed to a processor within the local cluster the interrupt device belongs to or to a remote processor via the node controller. So IOH



needs to check for local/remote routing as described in that register and if local, directly route the interrupt to the processor in the cluster and if remote, route to the node controller, whose NodeID is also identified in the QPIPAPICSAD register.

8.3.2.5 IA-32 Interrupt Delivery Summary

Table 8-10. IA-32 Interrupt Delivery Summary (Sheet 1 of 2)

| Mode ¹ | Sub-Mode ² | APICID | Target APIC | NodeID Determined by IOH | IOH Behavior | |
|-------------------|-----------------------|---------------------------|---------------------------------|--|--|--|
| Legacy Physical | Directed | APICID[7:0]≠0xFF | APIC identified in APICID | NodeID ³ as determined by QPIPAPICSAD register | IntPhysical to selected ³ Intel QuickPath Interconnect NodeID | |
| | | APICID[7:0]=0xFF | All enabled APICs | All CPU NodeIDs | Broadcast IntPhysical to all CPU NodeIDs. | |
| | Redirected | APICID[7:0]≠0xFF | APIC identified in APICID field | NodeID ⁴ as determined by QPIPAPICSAD register | IntPhysical to selected ⁴ Intel QuickPath Interconnect NodeID | |
| | | APICID[7:0]=0xFF | - | - | Report Error | |
| Extended Physical | Directed | APICID[31:0]≠0xFFFF FFFFF | APIC identified in APICID | NodeID ³ as determined by QPIPAPICSAD register | IntPhysical to selected ³ Intel QuickPath Interconnect NodeID | |
| | | APICID[31:0]=0xFFFF FFFFF | All enabled APICs | All CPU NodeIDs | Broadcast IntPhysical to all CPU NodeIDs. | |
| | Redirected | APICID[31:0]≠0xFFFF FFFFF | APIC identified in APICID field | NodeID ⁴ as determined by QPIPAPICSAD register | IntPhysical to selected ⁴ Intel QuickPath Interconnect NodeID | |
| | | APICID[31:0]=0xFFFF FFFFF | - | - | Report Error | |
| Legacy Logical | Flat ⁵ | Directed | DC | APICs specified in APICID[7:0] bit vector | All CPU NodeIDs | IntLogical to all CPU Intel QuickPath Interconnect NodeIDs (IOH can broadcast up to 8S in a cluster) |
| | | Redirected | DC | IOH selects one APIC from 8 possible APICs specified by APICID[7:0], in a simple round-robin (OR) APIC selection based on Vector-Number ⁶ | All CPU NodeIDs | IntLogical to all CPU Intel QuickPath Interconnect NodeIDs (IOH can broadcast up to 8S in a cluster) |



Table 8-10. IA-32 Interrupt Delivery Summary (Sheet 2 of 2)

| Mode ¹ | | Sub-Mode ² | APICID | Target APIC | NodeID Determined by IOH | IOH Behavior |
|-------------------|-----------------------|------------------------------|------------------------|---|---|--|
| Legacy Logical | Cluster | Directed | APICID[19:16]!=0xF | Specified APIC(s) of APICID[3:0] in the specified cluster identified by APICID[19:16] | All CPU NodeIDs | IntLogical to all CPU NodeIDs (IOH can broadcast up to 8S in a cluster) |
| | | | APICID[19:16]=0xF | Specified APIC(s) in all clusters | All CPU NodeIDs | IntLogical to all CPU NodeIDs (IOH can broadcast up to 8S in a cluster) |
| | | Redirected ^{7, 8} | APICID[19:16]!=0xF | IOH selects one APIC from 4 possible APICs specified by bit-vector APICID[3:0] a simple round-robin involving the four APICs in the cluster (note this requires IOH to maintain a round-robin arbiter per cluster, max 15 arbiters) OR APIC selection based on Vector-Number ⁹ | All CPU NodeIDs (IOH can broadcast up to 8S in a cluster) | IntLogical to all CPU NodeIDs |
| | | | APICID[19:16]=0xF | - | - | Report Error |
| Extended Logical | Cluster ¹⁰ | Directed | APICID[31:16]!=0xF FFF | Specified APIC(s) of APICID[15:0] in the specified cluster identified by APICID[31:16] | NodeID ³ as determined by QPIPAPICSAD register | IntLogical to selected ³ Intel QuickPath Interconnect NodeIDs |
| | | | APICID[31:16]=0xF FFF | Specified APIC(s) in all clusters | All CPU NodeIDs | IntLogical to all CPU NodeIDs (IOH can broadcast up to 8S in a cluster) |
| | | Redirected ^{11, 12} | APICID[31:16]!=0xF FFF | IOH selects one APIC from 8 possible APICs specified by bit-vector APICID[7:0] a simple round-robin involving the 8 APICs in the cluster (note this requires IOH to maintain a round-robin arbiter per cluster, max 15 arbiters) | NodeID ³ as determined by QPIPAPICSAD register | IntLogical to selected ³ Intel QuickPath Interconnect NodeIDs |
| | | | APICID[31:16]=0xF FFF | - | - | Report Error |

Notes:

1. Determined by either bit A[2] in the MSI address (when remapping is disabled) or by the combination of EIE bit in the Interrupt Remapping Table Address registers (defined in VT-d2 spec) and RTE[2] (when remapping is enabled)
2. Determined by either bit A[3] in MSI address (when remapping is disabled) or IRTE[3] (when remapping is enabled). Value of 1b is redirected and 0b is directed
3. This default 'routed' behavior can be changed via QPIPINTRC[6], so that these interrupts are broadcast to all processors. Final RH is defined by QPIPINTRC: [Intel QuickPath Interconnect Protocol Interrupt Control](#).
4. This default 'routed' behavior can be changed via QPIPINTRC[6], so that these interrupts are broadcast to all processors. Final RH is defined by QPIPINTRC: [Intel QuickPath Interconnect Protocol Interrupt Control](#).
5. The sub-mode within logical addressing mode is determined by bit 1 in QPIPINTRC: [Intel QuickPath Interconnect Protocol Interrupt Control](#)
6. See [Section 8.3.2.2](#) for details.



7. IOH will always set the redirection hint bit to a 0 when sending IntLogical messages in legacy mode, on Intel QuickPath Interconnect. Note IOH can set/clear RH bit in IntLogical Messages in the extended mode, depending on whether it is performing extended logical mode redirection or not
8. While redirecting interrupts, IOH selects one APIC among many and updates the APICID field in the received interrupt, with only the bit corresponding to the selected APIC set and all other bits clear, before forwarding to Intel QuickPath Interconnect
9. See Section 8.3.2.2 for details
10. Extended Flat mode is not architecturally supported
11. IOH will always set the redirection hint bit to a 0 when sending IntLogical messages in legacy mode, on Intel QuickPath Interconnect. Note IOH can set/clear RH bit in IntLogical Messages in the extended mode, depending on whether it is performing extended logical mode redirection or not
12. While redirecting interrupts, IOH selects one APIC among many and updates the APICID field in the received interrupt, with only the bit corresponding to the selected APIC set and all other bits clear, before forwarding to Intel QuickPath Interconnect.

8.3.3 MSI Forwarding: Intel Itanium Processor 9300 Series Based Platform

Table 8-11 summarizes the interrupt delivery requirements when IOH is in Intel Itanium processor 9300 series mode. The APICID[15:0] field identifies the targeted APIC. In Intel Itanium processor 9300 series, the APIC identified in the APICID field is always valid/enabled and IOH can always direct any incoming MSI to the corresponding APIC.

Table 8-11. Itanium® Processor Family Platform Interrupt Delivery

| Mode | Sub-mode | Target APIC | NodeID Determined by IOH | IOH Behavior |
|----------|------------|--|---|---|
| Physical | Directed | APIC identified in APICID[15:0] ¹ field | NodeID ² as determined by QPIPAPICSAD register | Send IntPhysical to selected ² Intel QuickPath Interconnect NodeID |
| | Redirected | | | |

Notes:

1. APICID[31:16] on Intel® QuickPath Interconnect is reserved in Intel Itanium processor 9300 series mode
2. This default 'routed' behavior can be changed via QPIPINTCR[6], so that these interrupts are broadcast to all processors. When these interrupts are broadcast, the RH bit is always set to 0 in the Intel® QuickPath Interconnect packet, otherwise it is preserved from the original MSI interrupt/IRTE entry

The mapping between APICID and NodeID is obtained as described in the QPIPAPICSAD register. There are some peculiarities associated with APICID to NodeID conversion for the Intel Itanium processor 9300 series which is discussed during the latter part of this section. QPIPAPICSAD register also provides details about how physical interrupt routing happens in hierarchical systems with node controller. For these systems, interrupts need to be either routed to a processor within the local cluster the interrupt device belongs to or to a remote processor via the node controller. So IOH needs to check for local/remote routing as described in that register and if local, directly route the interrupt to the processor in the cluster and if remote, route to the node controller, whose NodeID is also identified in the QPIPAPICSAD register.

Regarding APICID to Intel® QuickPath Interconnect NodeID conversion, APICID[11:8] identify the core within the CPU socket that the targeted APIC resides in. APICID[7:0]&APICID[15: 12] identify the socket. This is how the Intel Itanium processor 9300 series does the APICID assignment and the IOH has to follow suit. This is illustrated in Table 8-12. In general, in the Intel Itanium processor 9300 series, the NodeID for an interrupt can always be derived statically from the APICID field since this mapping is under BIOS/firmware control. But IOH supports OS transparently migrating a CPU from a running socket to a spare back-up socket and also the ability to mask interrupts from reaching certain sockets/cores for hard-partition management. These require the flexibility to map any APIC ID to any NodeID and IOH supports a table (as described in QPIPAPICSAD register) for this purpose.

IOH does not support any priority update messages on Intel QuickPath Interconnect (IOH will complete any IntPrioUpd message received normally on Intel QuickPath Interconnect but it has no effect on the interrupt forwarding behavior of IOH) for Intel Itanium processor 9300 series mode interrupt delivery and IOH does not support any interrupt redirection in Intel Itanium processor 9300 series mode. Expectation is that OS distributes the interrupts evenly across all the CPUs. Also, Intel Itanium processor 9300 series CPUs provide redirection amongst the cores in the socket and this is sufficient optimization. IOH, when setup for 'routed' mode, passes along the redirection hint bit it received in the MSI address/IRTE to the IntPhysical message. When IOH is setup for IntPhysical messages to be broadcast, it always clears the RH bit in the interrupt packet on Intel QuickPath Interconnect.

Table 8-12. Intel Itanium Processor 9300 Series APICID to NodeID Mapping Example (Setup through QPIPAPICSAD Register)

| Size | APICID[15:8] | APICID[7:0] | NodeID[5:0] |
|----------|-------------------------------------|-------------|---|
| 1-16S | ssss ¹ cccc ² | 00000000 | LocalNodeID ³ / <NCNODEID> ⁴ |
| 17-32S | sssscccc | 0000s00 | |
| 33-64S | sssscccc | 0000ss00 | |
| 65-128S | sssscccc | 000sss00 | |
| 129-256S | sssscccc | 00ssss00 | |

Notes:

- s represents the bits that are used to identify socket
- c in the table above refers to the bits used to identify core number in Intel Itanium processor 9300 series.
- LocalNodeID is an arbitrary number that is looked up from a table using bits a subset of 's' bits from the APIC ID field. This flexible mapping is provided for CPU migration and also to prevent software in one partition send interrupts to software in another partition.
- When interrupt is determined to be local (see QPIPAPICSAD register), NodeID=yyyy10. Otherwise the interrupt is remote and is routed to the node controller whose NodeID=NCNODEID, as derived from bits 13:8 of QPIPAPICSAD register.

8.3.4 External I/OxAPIC Support

I/OxAPICs can also be present in external devices such as the PCI Express-to-PCI-X/PCI bridge (PXH) and ICH. For example, the PXH has two integrated I/OxAPICs, one per PCI bus, that are used to convert the INTx wire interrupts from PCI slots to local APIC memory writes. These devices require special decoding of a fixed address range FECx_xxxx in the IOH. The IOH provides these decoding ranges which are outside the normal prefetchable and non-prefetchable windows supported in each root port. Refer to Chapter 7, "System Address Map" for address decoding details.

8.4 Virtual Legacy Wires

In IA-32, IOH can generate VLW messages on Intel QuickPath Interconnect. The conditions are:

- Receiving NMI/SMI#/INTR/INIT# signals from the legacy bridge and forwarding to Intel QuickPath Interconnect as inband VLW messages. Similarly, the IOH receives the FERR# message from Intel QuickPath Interconnect and converts it to a pin output in the legacy IOH.
- Generating SMI/NMI VLW messages for error events the IOH reports directly to the processor.

The rest of this section describes generating VLW messages from the legacy pins only.



The IOH also supports converting the NMI/SMI#/INIT# signals to IntPhysical messages on Intel QuickPath Interconnect for Intel Xeon processor 7500 series and Intel Itanium processor 9300 series based platforms. Refer to [Section 8.5.2](#) for details. SMI#, NMI and INIT# are treated as edge-sensitive signals and INTR is treated as level-sensitive. The IOH generates a message on Intel QuickPath Interconnect for SMI#, NMI and INIT# whenever there is an asserting edge on these signals. The IOH creates a message on Intel QuickPath Interconnect for INTR whenever there is an asserting or a deasserting edge on these signals.

The IOH receives the FERR message from Intel QuickPath Interconnect and pulses the FERR# pin output to the legacy ICH. The IOH guarantees that any subsequent transactions to ESI (that is, transactions ordered behind FERR message) are not delivered to ESI till the FERR# pin asserts.

Note: Design should provide as much timing delay as possible between assertion of FERR# pin and delivering subsequent transactions to ESI, to keep the legacy FERR# emulation in Intel QuickPath Interconnect platforms, as close as possible to FSB platforms.

All the VLW messages (inbound over Intel QuickPath Interconnect) are considered 'synchronous'. These messages are inserted on Intel QuickPath Interconnect ahead of any completions from the ESI port. That is, as soon as the IOH sees an edge on the legacy signals from the IOH and a VLW message is to be scheduled, that VLW message is pushed ahead of any pending completion transactions from the ESI port.

The IOH broadcasts *all* VLW messages to all processors within the partition. The IOH does not support outbound VLW messages.

8.5 Platform Interrupts

8.5.1 GPE Events

The IOH generates GPE events for PCI Express Hot-Plug (Assert/Deassert_HPGPE) and PCI Express power management (Assert/Deassert_PMEGPE). PXH components below the IOH could generate Assert/Deassert_GPE messages for PCI-X slot hot-plug events. These GPE events are sent as level-triggered virtual wire messages to the legacy ICH. Processors generate Intel QuickPath Interconnect GPE messages for internal socket events. The Intel QuickPath Interconnect GPE events are routed as DO_SCI messages, which are edge triggered, to the legacy ICH.

The same rules that govern the collection and routing of legacy PCI INTx messages (refer to [Section 8.2](#)) through an IOH, also govern the collection and routing of all level-sensitive GPE messages.

[Figure 8-3](#) illustrates how hot-plug and Power Management GPE messages are routed to the legacy ICH.

Figure 8-3. Assert/Deassert_(HP, PME) GPE Messages

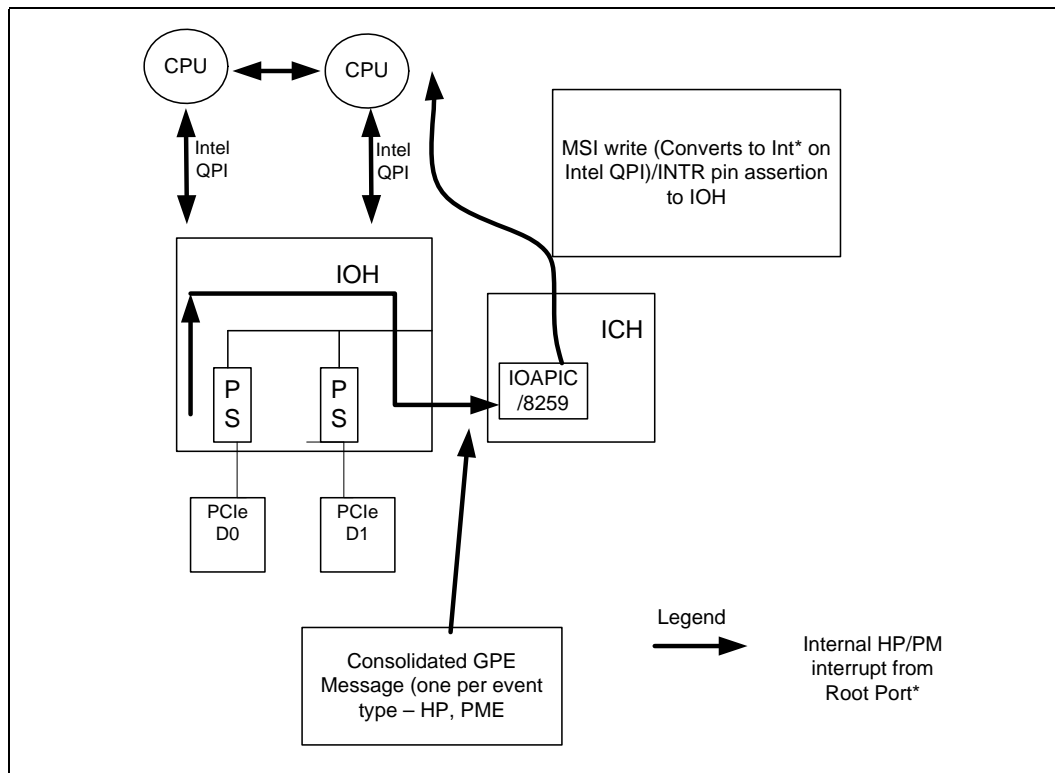
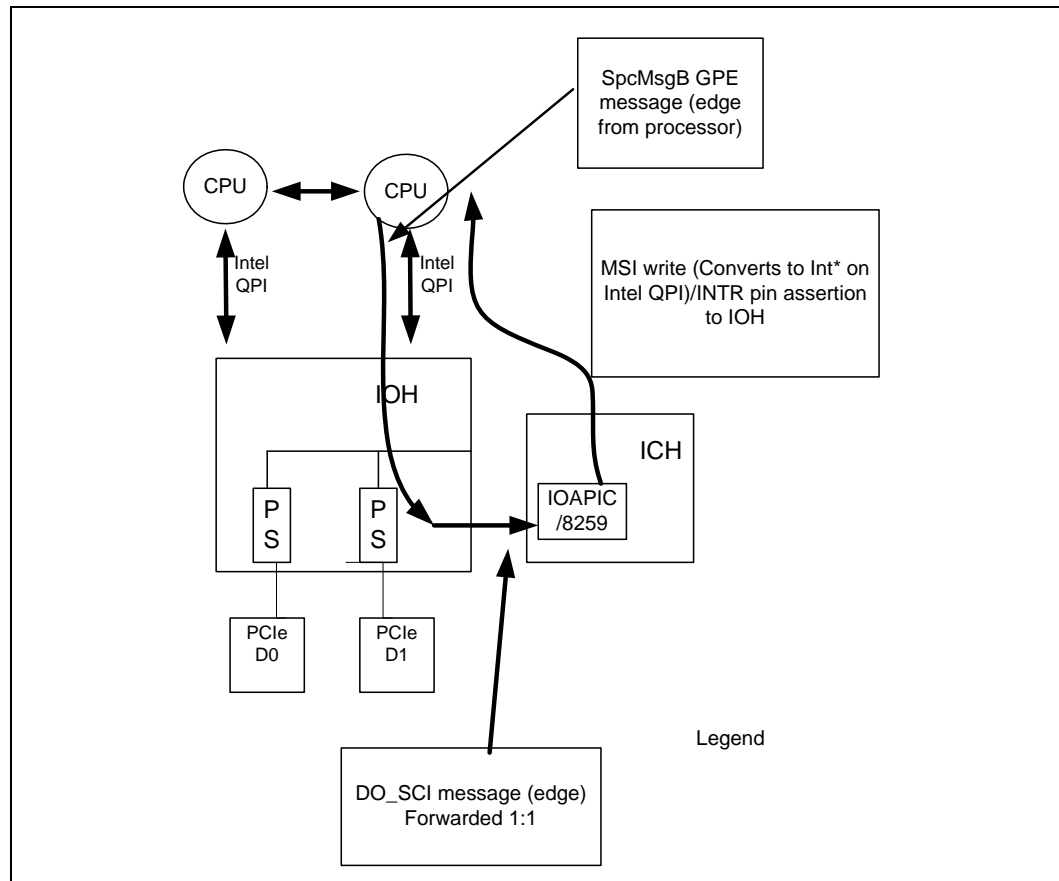


Figure 8-4 illustrates how GPE messages from the processor are routed to the legacy ICH. Processors generate GPE for a variety of events. Refer to the appropriate processor specification for details. Since the GPEX messages from the processor are edge-triggered and the DO_QPI message on ESI is also edge-triggered, the IOH transparently converts the Intel QuickPath Interconnect GPE message to the DO_SCI message and does not maintain any status bits.

Figure 8-4. Intel QuickPath Interconnect GPE Messages from Processor and DO_SCI Messages from IOH



8.5.2 PMI / SMI / NMI / MCA / INIT

The IOH can directly generate the IntPhysical (PMI/SMI/NMI/MCA) messages on Intel QuickPath Interconnect (ICH is bypassed) for RAS events such as errors and Intel® QuickPath Interconnect hot-plug. Also, refer to [Chapter 16, “IOH Error Handling Summary”](#) for error event causes of these interrupts. IOH generates an IntPhysical message on Intel QuickPath Interconnect for generating these interrupts. Note that the NMI pin input can be controlled to generate either a IntPhysical(NMI) or IntPhysical(MCA) message via the Interrupt Control Register (INTRCTRL). See the [Chapter 21, “Configuration Register Space”](#) for details.

Note: Software is responsible for programming the IOH error interrupt registers with the appropriate interrupt address and data when generating any of the interrupts above. Any broadcast requirements (for example, SMI interrupt) is indicated by programming the APICID field of the interrupt address with a value of 0xFF. The IOH does not make the determination that an interrupt should be broadcast based on interrupt encoding.

8.5.2.1 INIT#

IOH supports converting the INIT# pin into the corresponding IntPhysical message on Intel QuickPath Interconnect. An IntPhysical (INIT) message is sent on Intel QuickPath Interconnect whenever there is an asserting edge on INIT signal.



8.5.2.2 Global SMI

Normally, the IOH generates SMI based on some internal event or when it receives an SMI from one of its downstream ports. These SMI events are only sent to the sockets within the specific partition. Global SMI is used during quiescence flows on Intel QuickPath Interconnect where the Intel QuickPath Interconnect link configuration changes, for quickly bringing the system to a quiesced state. The IOH uses the quiescence broadcast list to send this global SMI. Note that in systems with larger than 8S, firmware must write to multiple IOHs to broadcast SMI to all processors in the system.

8.5.2.3 Software Initiated MCA

The IOH provides the capability to programmatically generate an MCA. Software can write to the 'Trigger MCA' bit in the MCA Register (MCA) to trigger an MCA to an indicated processor core within the partition.

8.5.3 CPEI

All non-legacy IOHs route hardware corrected errors they detect as ERR_COR message to the legacy IOH. This includes the PCI Express corrected errors also, provided native handling of these errors by the OS is disabled. If legacy IOH is to be reached over Intel QuickPath Interconnect, the transaction is tunneled on Intel QuickPath Interconnect via the NcP2PB packet. All CPEI events from processors are also sent to the legacy IOH via the Intel QuickPath Interconnect CPEI message. Legacy IOH combines the corrected error messages received from processors and IOHs.

ERR_N[0] will not be asserted from the message block by decoding CPEI message. A CPEI message is decoded by the Intel QuickPath Interconnect block where the corresponding error logging status bit is set. The Intel QuickPath Interconnect block also drives the appropriate error severity signal to the Global Error Escalation block to assert the error pin ERR_N[0].

When legacy IOH converts an Intel QuickPath Interconnect CPEI message (which is edge-triggered) or an ERR_COR message to the ERR[0] pin (which is level-sensitive), it maintains a status bit, bit 0 in [Section 21.6.7.10](#). When this status bit is set, further Intel QuickPath Interconnect CPEI or ERR_COR messages are simply dropped. When this bit is cleared, a new Intel QuickPath Interconnect CPEI or ERR_COR message will set the status bit and assert the ERR[0] pin. Software must clear this status bit before it polls all sources of corrected errors.

§



9 System Manageability

9.1 Introduction

This section combines many different features into one category that aids in platform or system management. Features such as SMBus and JTAG Test Access Port provide the protocol interfaces for access to the configuration registers. These registers are the program interface between the logical feature implementation and the software interface producing or consuming the data. System management uses this data for error diagnosis, system integrity, or work load analysis to optimize the performance of the platform.

Several miscellaneous features that aid in system manageability are presented here.

9.2 Error Status and Logging

System manageability requires that errors and their logs are captured in registers and accessible through the SMBus interface. Error status and logging is defined in the IOH RAS section of this specification for further information. Error counters and a "Stop on Error" feature are provided to support system management functions. An error freeze mechanism, with programmable error severity, is also provided, to halt traffic on the interfaces when an error occurs. Details are described in [Section 16.4.4.3, "Stop on Error" on page 227](#).

9.3 Component Stepping Information

Component stepping information is provided for PCI Express RID assignments. This information is also used in the JTAG ID code field. BIOS can override this value so that old code can execute on a newer stepping of the IOH.

9.4 Intel® Interconnect Built-In Self Test

Intel® Interconnect Built-In Self Test (Intel® IBIST) has features for the IOH's Intel QuickPath Interconnect and PCI Express interfaces. Intel IBIST is only available through the JTAG port using an externally enabled third party vendor. Contact your Intel Sales Representative for vendor information.

9.5 Hot-Plug Status Access

System management has full access to the status and control registers for hot-plug events. PCI Express hot-plug events are controlled through configuration register access.

9.6 Link Status Indication

Each Intel QuickPath Interconnect and PCI Express interface contains status bits to indicate if it is currently active and the frequency of operation. See [Table 9-1](#).



Table 9-1. Status Register Location Table

| Interface | Register Reference | Comments |
|---|--|---|
| Intel QuickPath Interconnect – Frequency Indication | Section 21.6.7.19, “CAPTIM: Cap Timer” | The register contains the frequency of each port. |
| PCI Express – Active | Section 21.12.4.20, “LNKSTS: PCI Express Link Status Register” | Bits [9:4] indicates the negotiated width. |
| PCI Express – Frequency Indication | Section 21.12.4.20, “LNKSTS: PCI Express Link Status Register” | Bits [3:0] will indicate the link speed. |

9.7 Thermal Sensor

The IOH integrates a thermal sensor that allows system management software to monitor and regulate the thermal activity levels in the die.

§

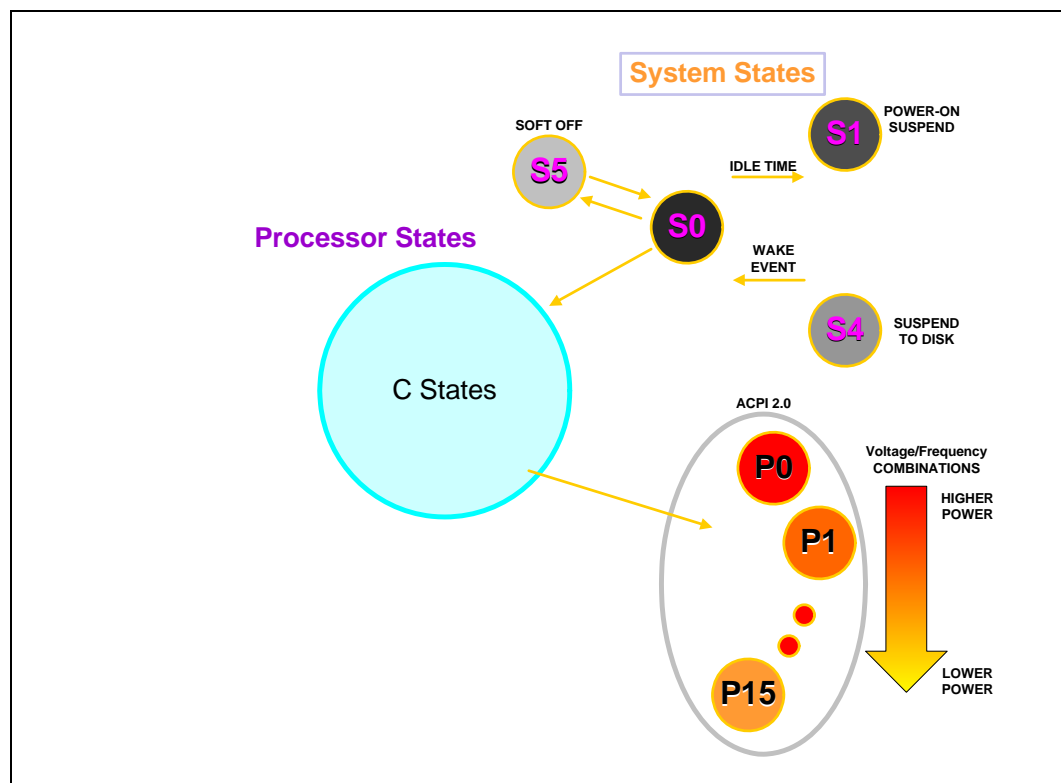


10 Power Management

10.1 Introduction

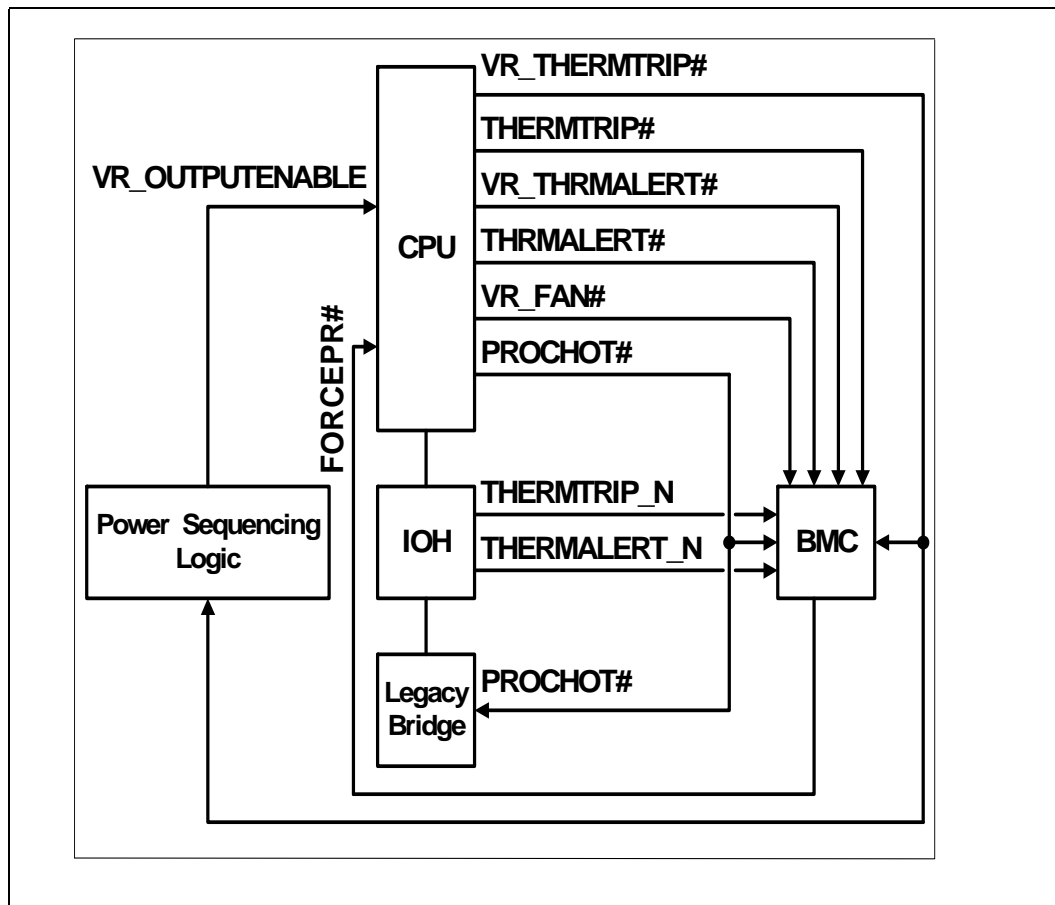
IOH power management is compatible with the *PCI Bus Power Management Interface Specification, Revision 1.1* (referenced as PCI-PM). It is also compatible with the *Advanced Configuration and Power Interface (ACPI) Specification, Revision 2.0b*. The IOH is designed to operate seamlessly with operating systems employing these specifications.

Figure 10-1. ACPI Power States in G0 and G1 States



Platforms are expected to incorporate a system management controller, such as the BMC. Numbers of "P"-states cited in Figure 10-1 are examples ONLY. P-states supported by the platform should not be inferred from these examples.

Figure 10-2. Typical Platform Showing Power Saving Signals to BMC



10.2 Supported Processor Power States

Refer to [Table 10-1](#) for examples. Since no Intel QuickPath Interconnect messages are exchanged upon C-state transitions within the processor sockets, the IOH is not involved.

Refer to [Table 10-2](#) for further details connected with System (S) states.

10.3 Supported System Power States

The supported IOH system power states are enumerated in [Table 10-1](#). Note that no device power states are explicitly defined for the IOH. In general, the IOH power state may be directly inferred from the system power state.



Table 10-1. IOH Platform Supported System States

| System State | Description |
|--------------|--|
| S0 | Full On Normal operation. |
| S1 | Stop-Grant <ul style="list-style-type: none"> • No reset or re-enumeration required. • Context preserved in caches and memory. • All processor threads go to the C1 state. • After leaving only one “monarch” thread alive among all threads in all sockets, system software initiates an I/O write to the SLP_EN bit in the ICH's power management control register (PMBase + 04h) and then halts the “monarch”. This will cause the ICH to send the GO_C2 ESI message to the IOH. The IOH responds with an ACK_C2 ESI message to the ICH. (The “monarch” is the thread that executes the S-state entry sequence.) |
| S4 | Suspend to Disk (STD) [Supported] CPU, PCI and Memory reset. The S4 state is similar to S3 except that the system context is saved to disk rather than main memory. This state is commonly known as “Hibernate”. The IOH uses the same sequence as S3. |
| S5 | Soft off [Supported] Power removed. The IOH supports this mode by following the S3 sequence. |

The IOH platform supports the S0 (fully active) state since this is required for full operation. The IOH also supports a system level S1 (power-on suspend) state but the S2 is not supported. The IOH supports S4/S5 powered-down idle sleep states. In the S4/S5 states, platform power and clocking are disabled, leaving only one or more auxiliary power domains functional. Exit from the S4 and S5 states requires a full system reset and initialization sequence.

A request to enter the S4/S5 power states are communicated to the IOH by the ICH via the “Go_S3” vendor defined message on the ESI interface. In response, the IOH will return an “Ack_S3” vendor defined message to the ICH. Upon completion of this sequence, the IOH will tolerate the removal of all reference clocks and power sources. A full system initialization and configuration sequence is required upon system exit from the S4/S5 states, as non-AUX internal configuration information is lost throughout the platform. AUX power sources must remain “up.”

10.3.1 Supported Performance States

The IOH platform does not coordinate P-state transitions between processor sockets with Intel QuickPath Interconnect messages. As such, the IOH supports, but is uninvolved with, P-state transitions.

10.3.2 Supported Device Power States

The IOH supports all PCI-PMI and PCI Express messaging required to place any subordinate device on any of its PCI Express ports into any of the defined device low power states. Peripherals attached to the PCI segments provided via the ICH/PXH components may be placed in any of their supported low power states via messaging directed from the IOH through the intervening PCI Express hierarchy.

In addition, the Crystal Beach integrated device (DMA) in the IOH can be placed in D0 or D3hot states by programming the PMCSR register of its power management structure. Any of the standard PCI Express ports¹ and ESI can be placed in D0 or D3hot states by programming the respective PMCSR registers.



Directly attached native PCI Express devices are not limited in their available low power states, although not all available states support the downstream device “wake-up” semantic.

10.3.3 Supported ESI Power States

Transitions to and from the following Power Management states are supported on the ESI Link:

Table 10-2. System and ESI Link Power States

| System State | CPU State | Description | Link State | Comments |
|--------------|-----------|--|------------|--|
| S0 | C0 | Fully operational / Opportunistic Link Active-State. | L0/L0s | Active-State Power Management |
| S0 | C1 | CPU Auto-Halt | L0/L0s | Active-State Power Management |
| S1 | C2 | (S1 same as C1/C2) | L0/L0s | Active-State Power Management |
| S3/S4/S5 | N/A | STR/STD/Off | L3 | Requires Reset. System context not maintained in S5. |

10.4 Intel® QuickData Technology Power Management

Chipset devices that implement Intel® QuickData Technology (formerly code named Crystal Beach) as well as client I/O devices that use Intel QuickData Technology may support different device power states. The IOH Intel QuickData Technology device supports the D0 device power state that corresponds to the “fully-on” state and a pseudo D3hot state. Intermediate device power states D1 and D2 are not supported. Since there can be multiple permutations with Intel QuickData Technology and/or its client I/O devices supporting the same or different device power states, care must be taken to ensure that a power management capable operating system does not put the Intel QuickData Technology device into a lower device power (for example, D3) state while its client I/O device is fully powered on (that is, D0 state) and actively using Intel QuickData Technology. Depending on how Intel QuickData Technology is used under an OS environment, this imposes different requirements on the device and platform implementation.

10.4.1 Power Management with Assistance from OS Level Software

In this model, there is a Intel QuickData Technology device driver and the host OS can power-manage the Intel QuickData Technology device through this driver. The software implementation must make sure that the appropriate power management dependencies between the Intel QuickData Technology device and its client I/O devices are captured and reported to the operating system. This is to ensure that the operating system does not send the Intel QuickData Technology device to a low power (for example, D3) state while any of its client I/O devices are fully powered on (D0) and actively using Intel QuickData Technology. For example, the operating system might attempt to transition the Intel QuickData Technology device to D3 while placing the system into the S4 (hibernate) system power state. In that process, it must not

1. This is a Pseudo state. The IOH will not physically power down the logic associated with the PCI Express/ESI ports though the D3hot is supported. From Software point of view, it appears as if the device goes into D3hot. Future proliferations could consider real D3hot implementations on the PCI Express ports.



transition the Intel QuickData Technology device to D3 before transitioning all its client I/O devices to D3. In the same way, when the system resumes to S0 from S4, the operating system must transition the Intel QuickData Technology device from D3 to D0 before transitioning its client I/O devices from D3 to D0.

10.5 Device and Slot Power Limits

All add-in devices must power-on to a state in which they limit their total power dissipation to a default maximum according to their form-factor (10W for add-in edge-connected cards). When BIOS updates the slot power limit register of the root ports within the IOH, the IOH automatically transmits a Set_Slot_Power_Limit message with corresponding information to the attached device. It is the responsibility of platform BIOS to properly configure the slot power limit registers in the IOH. Failure to do so may result in attached endpoints remaining completely disabled in order to comply with the default power limitations associated with their form-factors.

10.5.1 ESI Power Management

1. The IOH sends the ACK-Sx for Go-C0, Go-C2, Go-S3 messages.
2. The IOH never sends an ACK-Sx unless it has received a Go-Sx.

10.5.1.1 S0 -> S1 Transition

Note: Steps referring to SAL are for Intel Itanium Processors based systems only

1. The processor "monarch" thread spins on a barrier
2. The OSPM "monarch" performs the following functions:
 - Disables interrupts
 - Raises TPR to high
 - Sets up the ACPI registers in the ICH
 - Sets the fake SLP_EN which triggers a SAL_PMI
 - Spins on WAK_STS
3. The SAL PMI handler writes the Sleep Enable (SLP_EN) register in the Legacy Bridge. After this, the last remaining "monarch" thread halts itself.
4. The ICH responds to the SLP_EN write by sending the Go_C2 Vendor-Defined message to the IOH
5. The IOH responds to Go_C2 by multicasting a NcMsgB-PMReq(S1) message to the CPUs
6. The CPUs respond by acknowledging the NcMsgB-PMReq(S1) message
7. The IOH responds to the NcMsgB-PMReq(S1)-Ack from the CPUs by sending the Ack_C2 Vendor_Defined message to the Legacy Bridge
8. The IOH and/or ICH may transition the ESI link to L0s autonomously from this sequence when their respective active-state L0s entry timers expire

10.5.1.2 S1 -> S0 Transition

1. The ICH detects a break event, for example, Interrupt, PBE and so on.
2. The ICH generates the Go_S0 Vendor_Defined message to the IOH
3. In response to reception of Go_S0, the IOH multicasts a NcMsgB-PMReq(S0) message to the CPUs

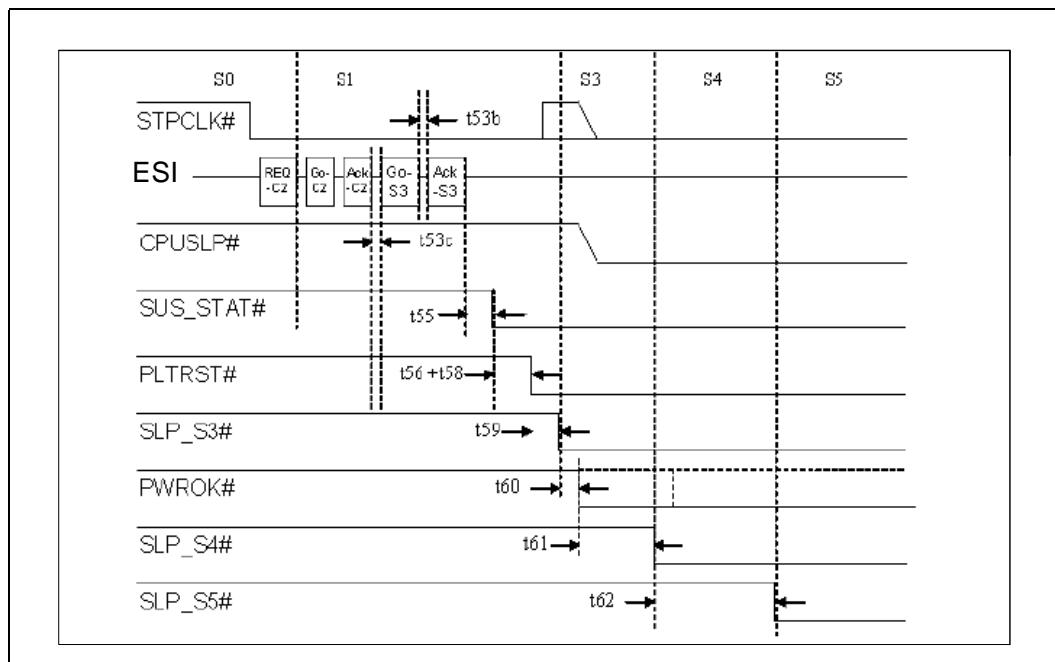
4. After receiving responses to the NcMsgB-PMReqs, the IOH sends the Ack_CO Vendor_Defined message to the ICH

10.5.1.3 S0 -> S4/S5 Transition

In the S3 sleep state, system context is maintained in memory. In S3-Hot, the power may remain enabled except for the processor cores and Intel QuickPath Interconnect (note that the PWRGOOD signal stays active). The IOH/ICH ESI link and all standard PCI Express links will transition to L3 Ready prior to power being removed, which then places the link in L3. In S3-Hot, the link will transition to L3 Ready and remain in this state until reset asserts and then de-asserts, since power is not removed.

Refer to [Figure 10-3, “ICH Timing Diagram for S4,S5 Transition”](#) on page 162 for the general interaction between the IOH Chipset and the ICH.

Figure 10-3. ICH Timing Diagram for S4,S5 Transition



Refer to the [Chapter 14](#) for the normal hard reset sequence that places the system back into S0.

10.6 PCI Express Interface Power Management Support

The IOH supports the following link states:

- L0s as receiver and transmitter
- L1 link state
- L3 link state
- MSI or GPE event on power manage events internally generated (on a PCI Express port hot-plug event) or received from PCI Express
- D0 and D3 hot states on a PCI Express port



- Wake from D3hot on a hot-plug event at a PCI Express port

The IOH does *not* support the following link states:

- ASPM L1 link state
- L1a link state (L1a is supported on ESI)
- L2 link state (that is, no auxiliary power to the IOH)
- Inband beacon message on PCI Express

10.6.1 Power Management Messages

When the IOH receives PM_PME messages on its PCI Express ports (including any internally generated PM_PME messages on a hot-plug event at a root port), it either propagates it to the ICH over the ESI link as an Assert/Deassert_PMEGPE message or generates an MSI interrupt. If 'Enable ACPI mode for PM' in the Miscellaneous Control and Status Register (MISCCTRLSTS) is set, GPE messages are used for conveying PM events on PCI Express, else MSI mode is selected.

The rules for GPE messages are the similar to the standard PCI Express rules for Assert_INTx and Deassert_INTx:

- Conceptually, the Assert_PMEGPE and Deassert_PMEGPE message pair constitutes a "virtual wire" conveying the logical state of a PME signal.
- When the logical state of the PME virtual wire changes on a PCI Express port, the IOH communicates this change to the ICH using the appropriate Assert_PMEGPE or Deassert_PMEGPE messages. Note: Duplicate Assert_PMEGPE and Deassert_PMEGPE messages have no affect, but are not errors.
- The IOH tracks the state of the virtual wire on each port independently and present a "collapsed" version (Wire-OR'ed) of the virtual wires to the ICH.

Note: Refer to [Chapter 8, "Interrupts"](#) for details on how these messages are routed to the ICH over Intel QuickPath Interconnect.

10.7 Other Power Management Features

10.7.1 Fine-Grained Dynamic Clock Gating

The IOH employs a traditional leaf-level clock-enable to clock-gate re-synthesis scheme.

10.7.2 Core Power Domains

- Intel ME: always "up"
- x16 PCIe link and protocol layers
- Everything else: shut down in standby.

10.7.3 L0s on Intel QuickPath Interconnect and PCIe

- Initiated by sender occasionally.
- Expecting very short exit latency target.



10.7.4 L1 on Intel QuickPath Interconnect and PCIe

L1 can also be used to save power during S1. On Intel QuickPath Interconnect, IOH needs to “wake-up-and-train” the link before propagating the S1-exit-to-S0 power-control message.

10.7.5 Static Clock Gating

Turn off clocks to subsystems that are disabled.

§



11 Partitioning

11.1 Partitioning Overview

Partitioning allows a system to be divided into multiple virtual machines capable of running their own operating systems and applications. Multiple partitions can exist within a system, each of which is logically isolated from the other, providing different degrees of reliability and security depending on the particular type of partitioning. In all partitioning schemes, partitions are isolated from the OS and applications perspective. Partitioning can be static or dynamic. With static partitioning, the system is partitioned at boot time and re-partitioning requires reboot of the affected partitions. The partitioning is dynamic if resources can be added or removed from a partition without the need to reboot the system or the unaffected partitions. The IOH supports dynamic partitioning. Partitioning in the following sections is always assumed to be dynamic, unless otherwise stated. Dynamic partitioning requires additional OS support to add or remove resources from a virtual machine.

The granularity at which resources can be added to or deleted from a partition is referred to as a “module”. The modules supported are dependent on the platform and partitioning technology. The partitioning could be coarse, where the module may be a field replaceable unit (FRU); or partitioning could be fine, at the level of individual I/O devices, CPU sockets, or CPU cores. On-line addition and deletion of a module from a running partition requires OS support. A module may be comprised of CPUs only, memory (+ memory controller), IO Hub, or some combination of the preceding, depending on the particular implementation and platform configuration. Partitioning control is done through system service processor(s) (SSPs), baseboard management controllers (BMCs) and/or protected firmware running on the processor(s).

Benefits of partitioning include:

- **Consolidation:** A single powerful system running different operating systems and applications, capable of managing multiple independent physical computers in the system.
- **RAS:** Improved availability as faults are isolated to individual partitions. Resources can be re-assigned to different partitions without rebooting the platform and potential minimization of system down time.
- **Development/validation environment:** New hardware and software updates can be tested in an isolated partition prior the migration to the system.
- **Gradual rebalance of resources:** Dynamic partitioning with hot-add and hot-remove will allow hardware resources to be moved between partitions to optimize for the computer workloads. Dynamic partitioning also allows creation of more partitions or deletion of existing partitions without re-booting. This helps optimize the resources for the expected load.

11.1.1 Types of Partitioning

Two types of partitioning supported by the IOH are described in this section: Hard, and Virtual partitioning. Generally there are two opposing requirements for partitioning: Isolation and Flexibility.

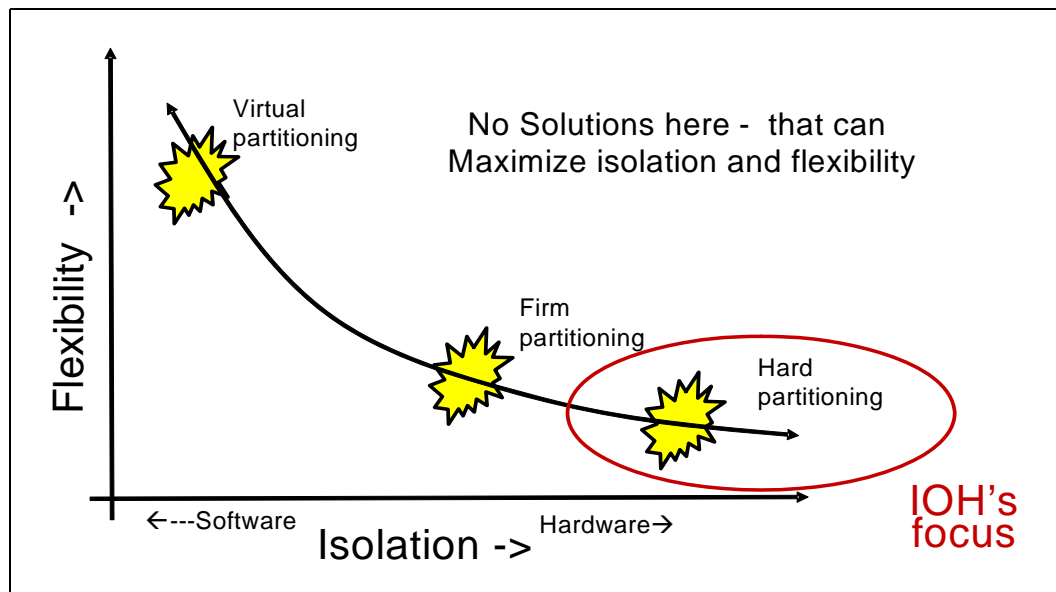
- **Isolation:** Refers to how well a partition is isolated from others. Greater isolation means better RAS – A failed partition cannot affect the operation of

another partition. Better isolation requires fewer shared components/hardware between partitions, and minimal software dependence.

- **Flexibility:** Refers to the granularity of hardware resources that can be assigned to or removed from a partition, and flexibility refers to how freely one can allocate and share hardware resources between different partitions. Better granularity and flexibility requires more (finer) partitions and shared hardware/software across partitions, and greater dependence on software to configure the partitions. Techniques that can get more flexibility tend to reduce isolation.

In general, the more flexible of the partitioning, the less the isolation. The relationship between partition type and the isolation/flexibility is illustrated in Figure 11-1.

Figure 11-1. Partition Type and Isolation/Flexibility Chart



The characteristics for each type of partition are described below:

- **Hard partitioning**

- Minimum granularity is at component level, for example, hardware components in the platform (CPU or chipset) are not shared across two or more partitions and any component is allocated to just one partition.
- Only buses and links between components cross partition boundaries, and those will be disabled or inactive. Transactions are isolated to components assigned to the partition by disabling the links crossing partition boundaries.
- Hardware errors in components and links are isolated to a partition at worst, and will not affect multiple partitions.
- Even firmware cannot distinguish hard partition from full system once the links that cross the partition are disabled and partition is re-booted.

Hard partitioning provides the highest level of partition isolation, but limited granularity and flexibility. Maximum number of hard partitions supported in the IOH is the minimum of (Number of processors, Number of IOH, Number of legacy devices). The granularity may be even lower and at the FRU level (multiple components on a single board) and fixed by the system/platform architecture. Partitioning configurations may be further restricted (for example, it may not be possible to combine any IOH or IO devices with any processors to form a partition). Hard partitioning requires more components and may increase the hardware cost of the platform. Hard partitioning is more suitable for larger platforms that support the highest RAS features.



- **Virtual partitioning**

- Most of hardware is shared across partitions, with hardware firewalls used when hardware is switched from one partition to the other.
- Partition boundaries may not physically exist at any given instant and may depend more on the context switches so that hardware is assigned to different partitions at different times. The granularity may be individual processor threads and IO devices OR even those may be shared across partitions.
- Virtual partitioning requires a software layer called Virtual Machine Monitor (VMM) below the operating system. VMM is aware of partitions and resources shared across partitions. VMM manages the hardware and uses the mechanisms provided by hardware to assign/protect OS/applications of one partition from the other.

Virtual partitioning provides the least isolation as it requires most of the hardware and even VMM software to be shared by different partitions, but provides the greatest flexibility in configuring partitions. Virtual partitioning can create more partitions than the number of processors and IO devices in the system. Unlike hard partitioning, virtual partitioning does not require any specific hardware support and can be entirely supported by the software. However, isolation and performance can be improved by appropriate hardware mechanisms in processor and chipsets. Virtual partitioning is suited primarily for consolidation and balancing of resources across partitions to improve platform efficiency.

Both processor and IOH provide hardware mechanisms to support virtual partitioning. The IOH includes an address translation table which allows virtualization of the IO devices.

11.1.2 Configuration of Partition – Static & Dynamic

Another main attribute of partitioning is the impact of changing partitioning configuration. The partitioning can be either static or dynamic.

Static Partitioning requires an operator to stop all CPU activities and shut down the system first, then take the steps to reconfigure the system. Reconfiguration is achieved either by manually changing the strap pins, or by outband agent configuring the partition CSRs of the chipset. Once the strap pins and CSRs are configured properly, a system wide reboot is issued. Upon reboot, the new configuration takes effect, and the system operates either in a single partition, or in multiple independent partitions. The benefit of static partitioning is its simplicity, but the drawback is the requirement to shut down the system.

- Benefit of Static Partitioning is its simplicity:
 - Requires no OS support
 - Requires little or no firmware support
 - Requires little BMC support
- Disadvantages of Static Partitioning are:
 - Requires halting all CPU activities and perform shutdown sequence
 - May require operator to manually change switches/straps for the new configuration
 - Requires system-wide reset to establish the new configuration

Dynamic partitioning differs from static partitioning in that reconfiguration of the system does not require the system to reboot. Consequently, dynamic partitioning allows the system to continue to operate while the system reconfiguration is in



progress. This is achieved through the hot add/remove (also termed as hot-plug/remove) of modules. With hot add/remove support, components can be taken off-line from one partition by the hot remove operation, and later inserted to a different partition by the hot add operation. No physical addition or removal of the component actually takes place. In general, the out-of-band system manager (BMC) coordinates with the CPUs to perform these operations. BMC configures the CSRs in various components, and generates hot-plug interrupts to notify the system of the hot add/remove requests. Upon receiving the hot-plug interrupt, the CPU performs the standard hot-plug operation and incorporates the new resources for merging, or removing the resources from its partition. The benefit of dynamic partitioning is the absence of rebooting the system. CPU in one partition can continue to operate, while the another partition is being created or merged. However, dynamic partitioning flow is significantly more sophisticated and requires substantial software and BMC support.

- Benefits of Dynamic Partitioning is partition/merge without shutting down the system:
 - Support through the hot-plug/remove flows of PCIe and Intel QuickPath Interconnect
- Disadvantage of Dynamic Partitioning is:
 - Dynamic flow is significantly more complex than static
 - Hot add/remove of CPU or IOH node requires sophisticated software and BMC support

Dynamic partitioning flow is detailed in the Common Platform Firmware Specification.

11.2 Hard Partitioning

An IOH system can be hard partitioned into independent domains. This implies each partition contains a full set of hardware resources such that an operating system cannot distinguish between a partition and an unpartitioned system. The partitions do not share the interconnect fabric or hardware resources between them. This requires that the individual domain is constructed with separate hardware. Interactions between different partitions are minimized so that hardware or software failures in one partition do not affect other partitions in the system. Partition isolation is enhanced by disabling the links between the partitions. The primary driver for the hard partition model is to enable system consolidation and to avoid single points of failure.

Figure 11-2. Example of Hard Partitioning

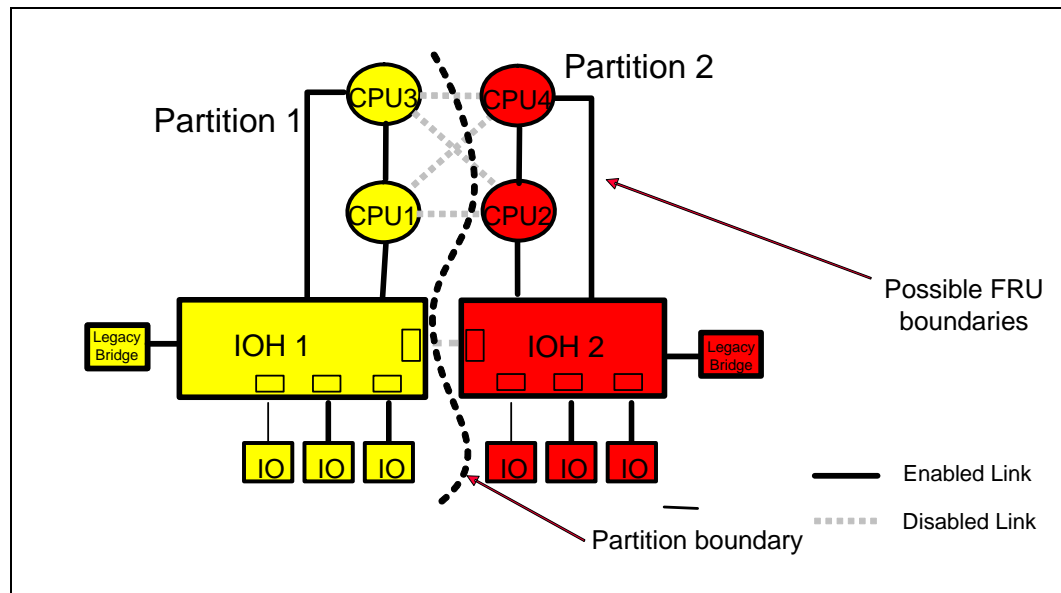


Figure 11-2 shows an example of IOH hard partitioning. In this example of a 4-CPU system, the hardware resources are evenly divided into two partitions at component granularity (for example, no component can belong to two different partitions). The legacy devices and FWH are also replicated for each partition. Hard partitioning has the following features:

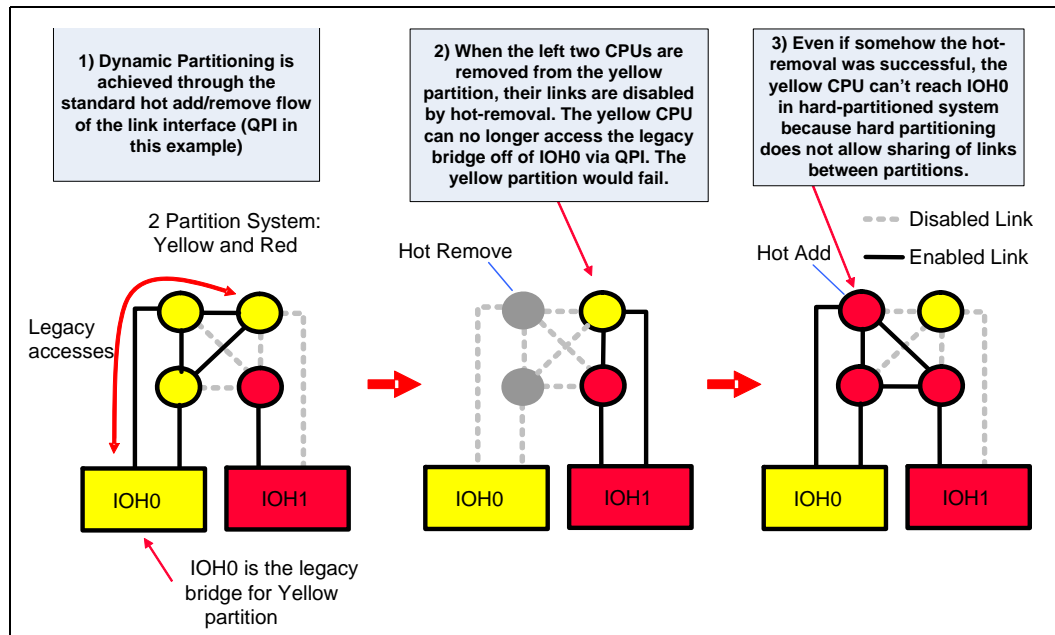
- **Isolation – guaranteed by hardware**
 - Each partition is able to reset and boot without affecting the other
 - Errors/faults are contained within the partition
 - Each partition sees its own address space and system resources. System resources between partitions are not shared.
- **Serviceability – minimal impact to up-time**
 - Partitions can be brought down for service independently (for example, to replace faulty parts)
 - System can hot-add and remove parts within a partition, or hot-add and remove the entire partition
- **Flexibility – dynamically reconfigure partitions**
 - Split/merge partition via firmware support
 - Add/remove resources to a partition via hot add/remove
- **Hard Partition Support by the IOH**
 - Link disabled/isolation to divide system into partitions
 - Partition reset isolation – ability to reset/re-boot components in a given hard partition

11.2.1 Hard Partitioning Topologies

A system can be hard partitioned into independent domains. These domains do not share the interconnect fabric nor hardware components between them. Since the domains do not share the interconnects, some dynamic partitioning rules apply in a

hard partition system. Figure 11-3 illustrates an example of an invalid hard partition reconfiguration. In this example the system decides to re-allocate two yellow CPUs to the red partition. However, when the system hot-removes the left CPUs, it also cuts off the Intel® QuickPath Interconnect path of the right yellow CPU to the legacy bridge causing system failure. As a rule, hot remove of a CPU for hard partitioning must not cut off the Intel® QuickPath Interconnect path to a legacy IOH. This rule together with other rules are stated in the subsequent section.

Figure 11-3. Example of Dynamic Hard Partitioning Violation



11.2.1.1 Hard Partitioning Topology Rules

The following lists the hard partitioning topology rules.

1. Hot removal of a CPU for partitioning must not cut off the Intel QuickPath Interconnect path to the legacy IOH of an active CPU (see Figure 11-3)
2. At least one of the IOH(s) in the partition must be connected to a CPU in the same partition by the Intel QuickPath Interconnect. If none of the IOHs are connected to the Intel QuickPath Interconnect fabric, then there is no way for IO to reach the CPUs as the CPU does not have PCIe links.
3. It is not necessary for all CPUs in a partition to connect to an IOH as long as the rules above are satisfied.

CPUs support routing Intel QuickPath Interconnect traffic through them and have built-in routers. A CPU not connected to an IOH can route its traffic through other CPUs. There must be a path as all CPUs are connected in the Intel QuickPath Interconnect and at least one IOH that is connected to the Intel QuickPath Interconnect fabric.



4. All links between components assigned to different partitions are disabled.
5. There is no memory or I/O sharing between two hard partitions.

Generally a partition may consist of one or more FRUs (Field Replaceable Unit). An FRU is a minimum unit of hardware that is replaceable in the platform. The IOH expects that any FRU will belong to the same partition. Therefore, replacing a faulty FRU unit would have minimal impact to the system and only affects that partition (although the partition may have to be shut down). It is also possible to service a faulty FRU without affecting any partition if the FRU can be hot removed without having to shutdown the partition using that FRU.

The following figures illustrate different hard partitioning examples adhering to the rules above.

11.2.1.2 Hard Partitioning in 2 IOH Configurations

The following figures illustrate different examples of 2 IOH hard partitioned topologies. Figure 11-4 shows an example of how hard partitioning can be applied in a 2-CPU system.

Figure 11-4. Examples of Hard Partitioning in 2-CPU Systems

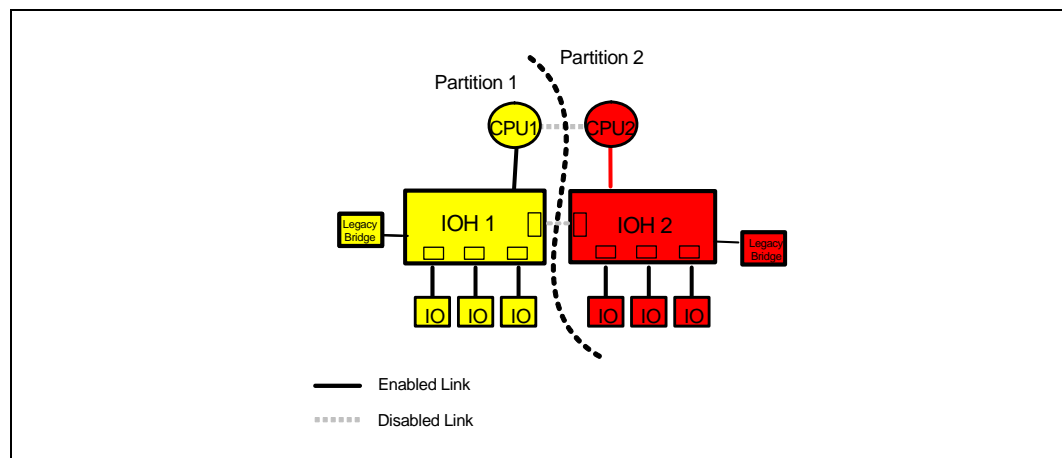


Figure 11-5 shows two examples of hard partitioning in a 4-CPU system. For example, if CPUs are grouped with IOH1 to form partition 1, then the remaining CPUs can be grouped with IOH2 to form partition 2.

Figure 11-5. Examples of Hard Partitioning in Topology 4-2-F (4 CPU, 2 IOH)

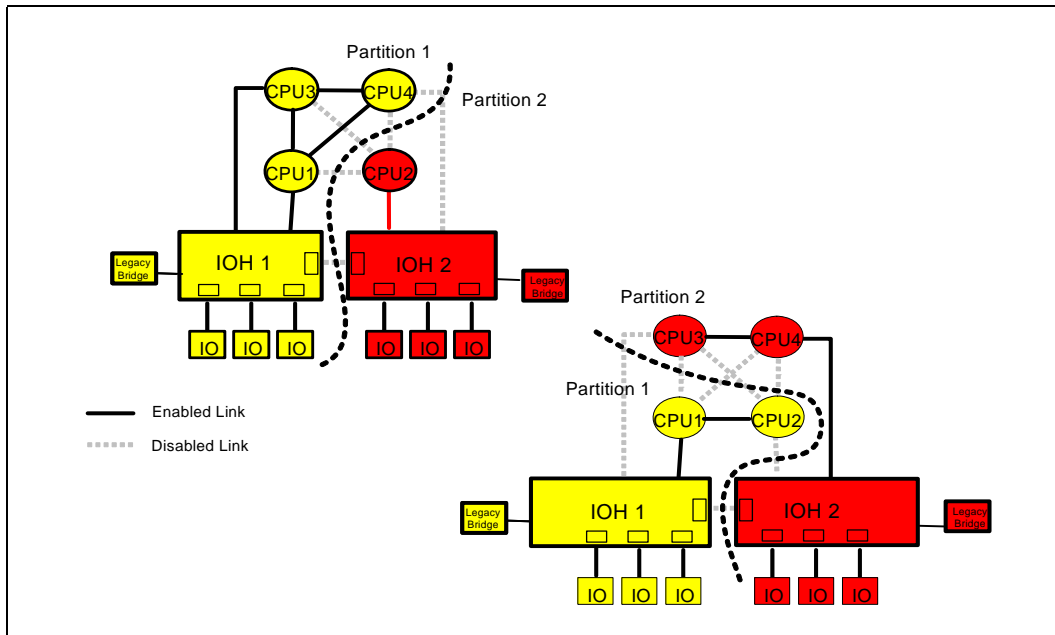
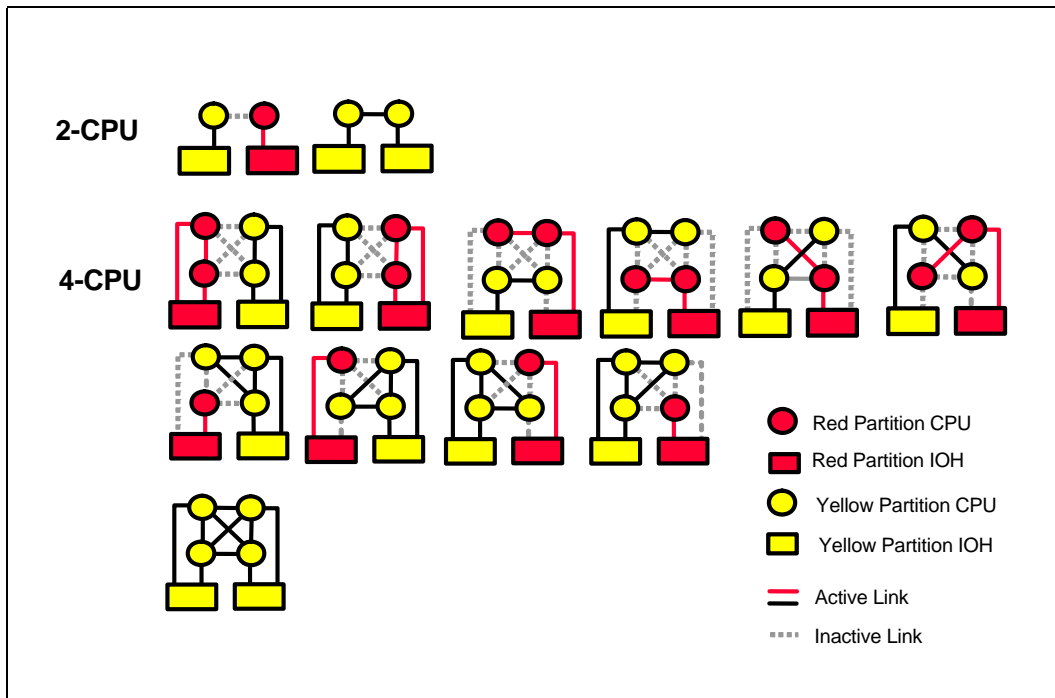


Figure 11-6 shows all legal hard partitioning in a 2-CPU and 4-CPU systems. Note that configurations that creates minimal hops between CPU and IOH is preferred. For example, the first configuration in the 2-CPU, and the first and second configurations in the 4-CPU.

Figure 11-6. Legal Hard Partitioning





11.3 Virtual Partitioning

An IOH component must entirely belong to a single partition. Virtual partitioning is supported through software, however IOH provides Extended Intel VT-d technology that allows I/O device virtualization. Please refer to the virtualization chapter for more details.

11.4 Hard Partition System Address Model

The system address model for the hard partition is identical to a system without partitioning. Each partition sees its own independent address space and system resources. There is no sharing of address space between partitions.

11.5 IOH Partitioning Support

This section describes the hardware features supported by the IOH for hard partitioning. The main requirement for the partitioning support is the ability to recognize transactions from different partitions and isolate them to the resources in their respective partitions. Traffic in one partition should never go to the resources in another partition. For hard partitioning, most of the hardware requirements are platform related and have little impact to the IOH. These mechanisms are detailed in the subsequent sections:

- **Partition Isolation:** IOH partition isolation is achieved through the Intel QuickPath Interconnect NodeID-based routing, and disabling Intel QuickPath Interconnect links at the partition boundary. A transaction targeting a specific node (belonging to a specific partition) is routed to the target based on the NodeID. The target NodeID is generated by the requesting component through the lookup of the source address decoder. By providing the proper programming of the source decoders, the address referenced by a CPU or IOH can only be directed to a component within its partition. In addition, the Intel QuickPath Interconnect links at the partition boundary can be electrically disabled to further isolate hard partitions. Therefore, partition isolation is achieved through the standard Intel QuickPath Interconnect functionality of the source address decoder and disabling of Intel QuickPath Interconnect ports. IOH provides no special hardware support for partitioning management.
- **Partition management:** IOH partitioning is managed by the BMC through the console interface. An operator can convey the desired partition topology to the BMC. The BMC conducts the programming of the NodeIDs of the components and the disabling/enabling of the Intel QuickPath Interconnect interface. Firmware then coordinates with BMC to provide proper programming of the source address decoders. The programmability of the NodeIDs and the Intel QuickPath Interconnect interface are parts of the standard Intel QuickPath Interconnect support. IOH provides no special hardware support for partitioning management.
- **Partition Reset:** Hard partition reset initializes all components within the partition to their reset default state. Since hard partitioning guarantees hardware isolation by disabling the links between the partition boundaries, a hard reset can be applied to a partition as if it were an independent machine. The partition reset is isolated within a partition. Resetting one partition does not affect functions of another partition. Partition reset could be generated by software with CF9 write to the active ICH in the partition, or any other reset source to the active ICH. These events will eventually cause the assertion of the PLTRST# reset output in the ICH. When this occurs, ICH puts itself in the reset state, and the assertion of PLTRST# can be used to drive partition reset. Platform hardware is responsible for driving



partition specific resets and in particular must account for the change in reset routing during partition transitions.

11.5.1 Considerations for Multiple ICHs in a Partition

Domain partitions that contain multiple ICHs must only have one active ICH within that partition. Each partition must have one designated legacy IOH where the active ICH must be connected via the ESI port. Any ICH connected to a non-legacy IOH must be at a minimum disabled by holding it in a reset state by inhibiting the VRMPWRGOOD signal while the POWROK signal performs its normal sequence. The IOH Strapping configuration for non-legacy IOH with a disabled ICH connected to the ESI port should be as follows: LEGACYIOH = 0, FWAGNT_ESIMODE = 0.

§



12 Scalable Systems

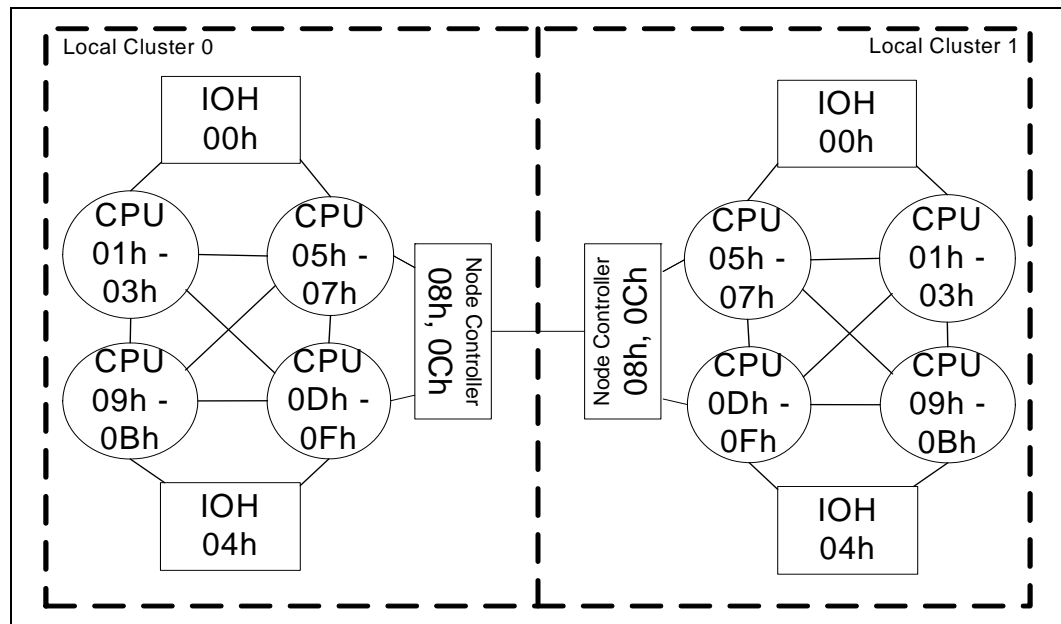
12.1 Introduction

This chapter will describe the requirements and limitations of systems above four sockets.

12.2 Intel QuickPath Interconnect Standard and Extended Headers

The IOH will support a maximum 5-bit NodeID with standard headers, and 6-bit NodeID with extended headers, which limits it to 32 or 64 NodeIDs. When more than this limit of NodeIDs is required in a system, a node controller must be used as a proxy for the additional agents. Figure 12-1 shows an example of an 8 socket system configuration with node controllers.

Figure 12-1. Example NodeID Assignment for 2-Cluster System with 4S Clusters



12.3 Broadcast from IOH

12.3.1 Coherent

Source broadcasting snoops use either a 32-entry or 64-entry broadcast vector, corresponding to a 5-bit or 6-bit NodeID limit for standard or extended headers, respectively. A hierarchical bridge (OEM component) within the local cluster can be used to manage coherency of non-local clusters.

The other option for dealing with coherency is to switch to a home node based scheme. In this scheme, source broadcast is turned off and the home agent will ensure coherency. This could be done by home node broadcast or by a directory based flow.

12.3.2 Non-Coherent (IA-32 Only)

IA-32 interrupts may broadcast to other caching agents. The broadcast vector within the IOH only targets 32 or 64 NodeIDs. Any broadcast outside the local cluster requires a hierarchical controller to send the requests to other agents.

Note: Physical APICID mode targets one NodeID such that broadcast is not required. All interrupts will be directed to either a local socket or the node controller.

12.3.3 Lock Arbiter (IA-32 Only)

IOH supports the outbound locks in IA-32. This requires a “Lock Arbiter” to arbitrate in the case of multiple agents. IOH has implemented a “Lock Arbiter” with an arbitration queue of 8 deep, which limits its scaling to 8 lock sources. This will equate to 8 sockets for the Intel Xeon processor 7500 series.

When scaling beyond 8 lock sources there are two options: (1) use a OEM “Lock Arbiter” that can support more than 8 sources, or (2) use a hierarchical controller to queue requests from remote clusters to ensure the IOH “Lock Arbiter” does not receive more the 8 simultaneous requests.

The “Lock Arbiter” in IOH can only support broadcast to the local cluster via a 32-entry or 64-entry broadcast vector. Beyond the 6-bit NodeID targets, a hierarchical controller must act as a proxy to remote clusters.

12.3.4 Quiesce Master

Quiesce Master functionality exists in the IOH which will quiesce all agents in the quiesce list. This list can only target 64 agents with a 6-bit NodeID. Software controls the initiation and completion of this flow via configuration registers within the IOH. In some systems the target lists will be both CPU and IOH. In others it may include only the IOH, and each CPU will be quiesced via its own configuration register.

In a hierarchical system there are two options for handling this flow: (1) the Node Controllers could forward the broadcast of StopReq*/StartReq* to each cluster, and ensure that completion is only given to the Quiesce Master after each of the remote cluster has completed each phase, or (2) software could control quiesce independently in each of the clusters, thus removing the need for node controller involvement.

12.4 Source Address Decoder

The source address decoder in the IOH will support 16 decoder entries for memory space (DRAM and MMIO/H). Each entry has the ability to do a low-order interleave across 8 NodeIDs.

The source address decode methodology leads to flat scaling limitations. The more home agents that exist, the more regularity required in the memory population to fit them within the 16 entry limit. There is also a strict limit on the number of targets the I/O decoder can target. For LocalxAPIC and Legacy I/O, only 8 NodeIDs can be targeted directly, which limits the flat system to 8 IOHs and 8 sockets. Scaling beyond this requires a hierarchical Node controller to decode outside of the local cluster.

Scaling with a hierarchical controller is limited by the I/O decoder’s ability to divide its address space because the range is handled by a single decoder. MMIO/H and LocalxAPIC have ranges separated into local and remote ranges, where the local range



is interleaved across the target list, and the remote range targets a single NodeID corresponding to the node controller. This method in IOH assumes a maximum of 64 clusters.

Note: The Intel Xeon processor 7500 series Source Address Decoder (SAD) model has similar alignment and interleave constraints to IOH. The Intel Itanium processor 9300 series has a SAD with more restrictive alignment and a 16-way interleave.

12.5 Performance

The IOH queues are sized for 4 socket system latencies achieving peak bandwidth on Intel QuickPath Interconnect and PCI Express for inbound reads and writes to DRAM. Therefore, the larger latencies seen in larger platforms with greater than 4 sockets will result in reduced performance. In hierarchical systems, this problem may be addressed by NUMA software optimizations that direct I/O traffic to local sockets.

12.6 Time-Out

Requirements grow for large system time-outs. IOH provides the ability to program the time-out on requests generated on the Intel QuickPath Interconnect to 43 seconds to meet large system requirements. Special consideration must be taken with PCI Express I/O cards. The current PCI Express specification requires a maximum time-out value of 50 ms, which is below the value necessary for large systems, although extensions have been added to allow growth to 64 sockets. Systems with larger time-out requirements may need non-standard cards to workaroud this issue.

12.7 PCI Segments

PCI bus numbering is limited to only 256 buses. The method to extend this is called "PCI Segments", which is a method to create multiple 256 bus segments in one system.

Requirements for segment support are:

- Peer-to-peer traffic allowed only within a segment
- Segments will be supported on an IOH granularity
- Processor address decode must have the ability to support multiple segments

12.8 Flat System Configurations

Flat system configurations are limited to 32 NodeIDs in standard header mode (5-bit) or 64 NodeIDs in extended header mode (6-bit). In standard header mode this inherently limits their scaling to support a maximum of 8 sockets, where the processor uses 3 NodeIDs each, and the remaining NodeIDs are allocated to the IOH.

The Intel Itanium processor 9300 series uses 5 nodeIDs per socket (4-caching agents, 1 configuration agent). This results in limiting Intel Itanium processor 9300 series-based platforms flat scaling with IOH to only 4S with standard headers or 8S with extended headers.

The home agent limits the number of the requests that each source may send to it. The limit is based on the ability to divide the home tracker among all the sources in the system. Some processors may limit the IOH to only 16 requests per home. If this occurs, then IOH will only be able to utilize 128 of the 256 ORB entries. This will result in a reduction in peak bandwidth from the IOH.

Figure 12-2. 8 Processor Topology Example

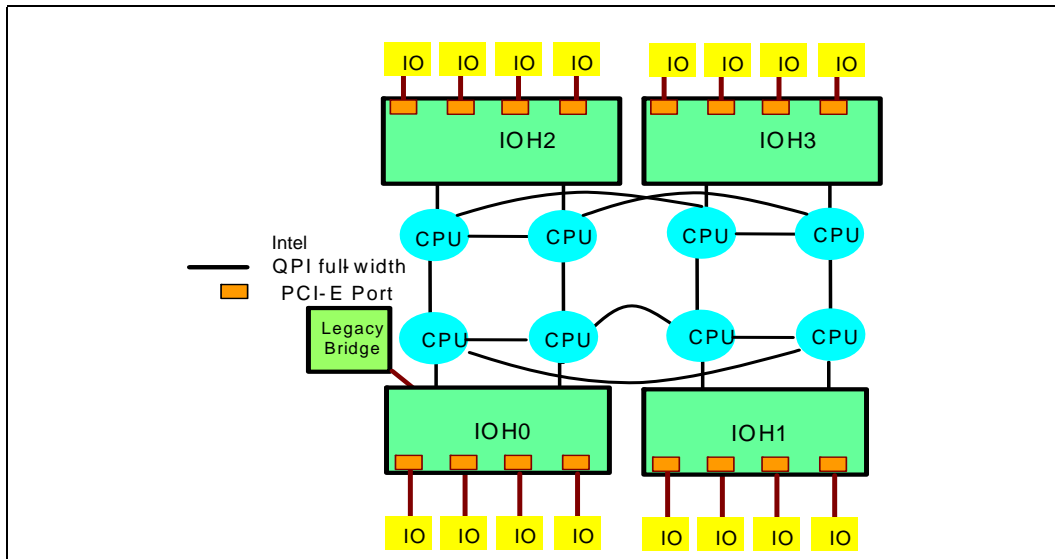


Figure 12-2 is an example configuration that is a “glueless” eight processor platform with four IOH components. Four hard partitions are possible.

Figure 12-3. 8 Processor Topology Example (Intel Itanium Processor 9300 Series-based Platform Only)

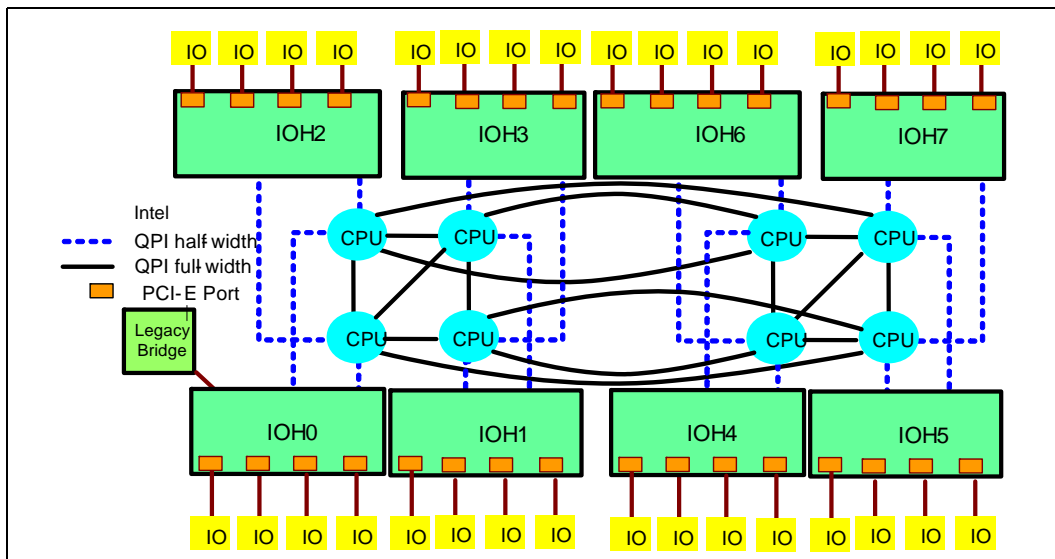


Figure 12-2 is an example configuration that is a “glueless” eight processor platform with eight IOH components. Eight hard partitions are possible. This example is only possible by using the extra Intel QuickPath Interconnect link provided by the Intel Itanium processor 9300 series.



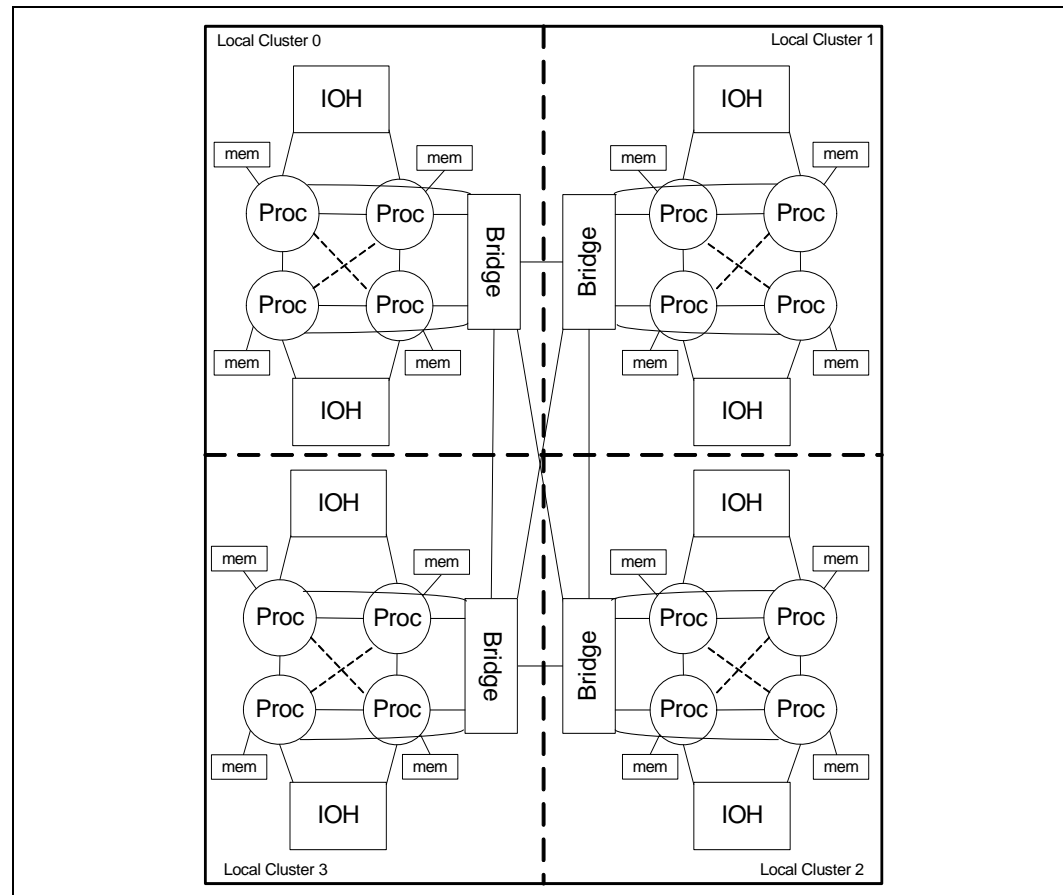
12.9 Hierarchical System Configurations

Hierarchical systems use the basic 2 or 4 socket system building blocks and a “hierarchical controller” that manages coherency outside of that block. The controller could broadcast snoops to other clusters or include a full or partial directory that filters snoops outside of the local cluster. See the example in [Figure 12-4](#). In this system type, the memory controller on the processor can be used without any special directory in the home agent. The CPU only supports 4 Intel QuickPath Interconnect links which limits its configurability in hierarchical systems. In this example, the cross-CPU Intel QuickPath Interconnect link, shown as a dashed line, could be removed.

Hierarchical system options/requirements:

- Hierarchical controller must do address decode to locate remote cluster and target NodeID within the remote cluster
- Source broadcast snooping may occur within the local cluster, but the hierarchical controller is responsible for all coherency outside of the local cluster
- Non-coherent broadcast requests may be done within the local cluster, but the hierarchical controller is responsible for broadcast in all remote clusters
- When there are more than 8 Lock sources in a system, there are two options: (1) Node Controller implements a lock arbiter, or (2) Node Controller queues locks to ensure the IOH with the Lock arbiter never receives more than 8 simultaneous locks.

Figure 12-4. Hierarchical System Example





§



13 Intel® Management Engine

This section includes details of the management interfaces for the platform, including the Reduced Media Independent Interface (RMII), and the Control Link (CLink).

13.1 Intel Management Engine Overview

The Intel Xeon processor 7500 series-based platform implements an Intel® Management Engine (Intel® ME) subsystem to provide Intel Server Platform Services (SPS) functionality. One hardware implementation is an optional discrete Baseboard Management Controller (BMC) component which can be used for manageability functions. The BMC allows the user to monitor and log the state of the system by utilizing out-of-band operations to determine the last operating state of the system.

The SPS on the Intel ME functions as satellite controller services to a typical system BMC that OEMs integrate. It does not serve as a standalone complete management solution. Also, at this time, the RMII interface is not planned to be used and as such does not need to be connected to anything. The BMC will handle access to LAN devices, and the ME acts only as satellite functions behind the BMC. Also, it is currently planned to make these services work without requiring discrete DRAM for Intel ME.

13.2 Management Engine External Interaction

This section describes the Intel ME interaction with the other entities in the platform. Although the interface to the other units within the IOH are beyond the scope of this document, this section highlights the software visible elements.

13.2.1 Receive

Intel ME can receive transactions from the CPU (host) and the Controller Link interface (CLINK).

The host accesses the Intel ME through a variety of software interfaces.

13.2.2 Transmit

Intel ME accesses to system memory are subject to Intel VT-d translation accesses.

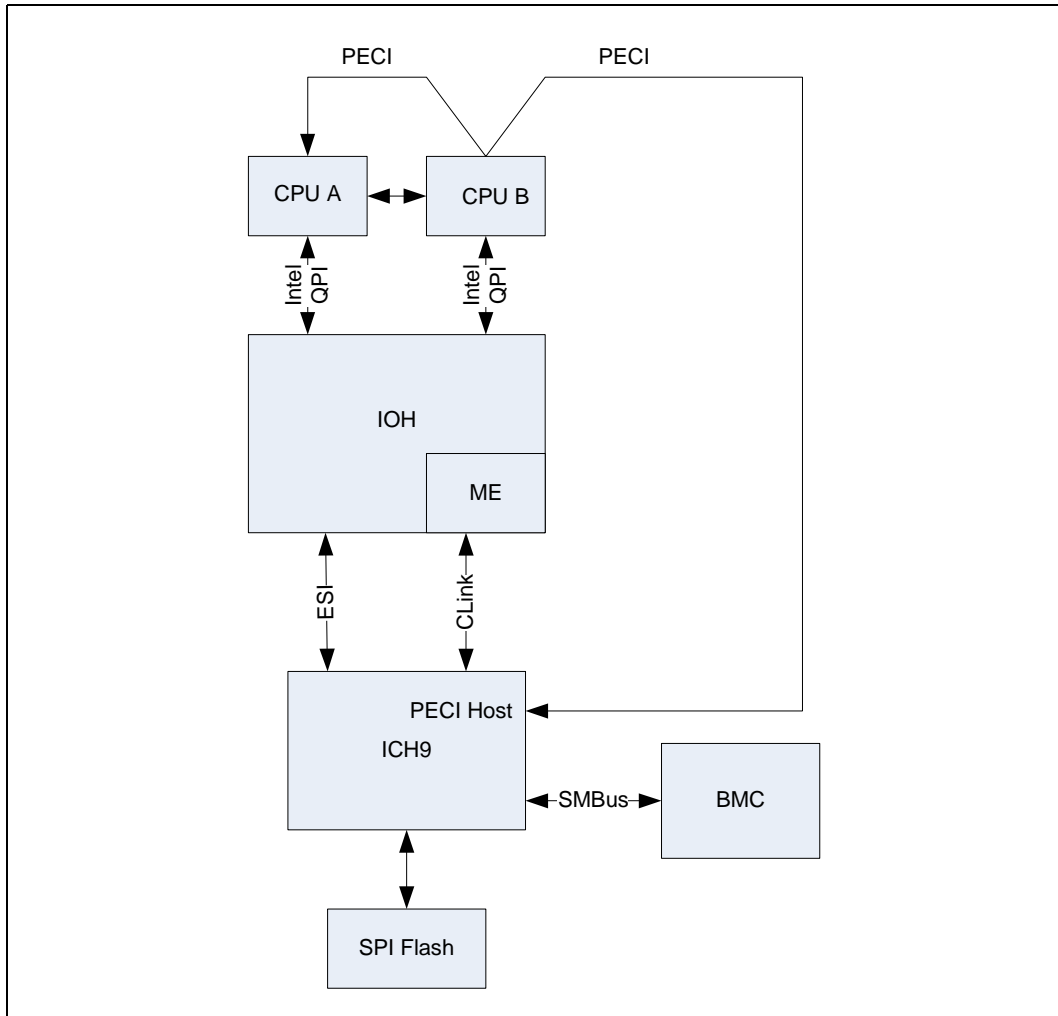
Table 13-1. Signal Type Definition (Sheet 1 of 2)

| Signal | Description |
|---------|--|
| In (I) | Input is a standard input-only signal |
| Out (O) | Totem Pole Output is a standard active driver |
| T/S | Tri-State is a bi-directional, tri-state input/output pin |
| S/T/S | Sustained Tri-State is an active low tri-state signal owned and driven by one and only one agent at a time. The agent that drives an s/t/s pin low must drive it high for at least one clock before letting it float. A new agent cannot start driving an s/t/s signal any sooner than one clock after the previous owner tri-states it. |
| O/D | Open Drain allows multiple devices to share as a wired-OR |

Table 13-1. Signal Type Definition (Sheet 2 of 2)

| Signal | Description |
|--------|-----------------------|
| A-in | Analog input signals |
| A-out | Analog output signals |
| B | Input bias |

Figure 13-1. Example of Intel ME Configuration with Intel SPS Implementation





13.3 Controller Link (CLINK)

The Controller Link (CLINK) is the private low pin count, low power communication interface.

Table 13-2 describes the CLINK signal names connected between the IOH and Intel ME.

Table 13-2. Controller Link Interface

| Signal Name | Type | Description |
|-------------|------|----------------------------|
| CLCLK | I/O | CLINK bi-directional clock |
| CLDATA | I/O | CLINK bi-directional data |
| CLRST_N | I | Active-low CLINK reset |

13.4 MESW Register

Note: The MESW CSR can only be accessed in the S0 power state. If firmware tries to read or write this CSR when the IOH is in any other S-state, it may cause a hang.

13.4.1 MESW_CBM_OVR_CTRL: Config Busmaster Override Control

| Device: 0 Function: 0 Offset: 400800h | | | |
|--|------------|---------|---|
| Bits | Attributes | Default | Description |
| 31 | RW | 0h | Override Enable (OVREN): set to 1'b1 to enable override of upstream transaction from Config Bus interface with CSR fields. This bit controls bits[23:16] and bits[6:0] of this register. |
| 30:25 | RO | 0h | Reserved (RSVD): Reserved |
| 24 | RW | 0h | MSI Enable (MSI_EN): set to generate an upstream MSI transaction via the CBM override mechanism. Note that {FMT, TYPE} should be set to {2'b10, 5'b0_0000} (Memory Write) for MSI transactions. |
| 23:16 | RW | 0h | Message Code (MC): Specifies MsgCode[7:0] for message request. |
| 15:8 | RW | 0h | Target Bus Number (TGTBUSNO): Specifies target bus number. This field will always be included in upstream config requests. |
| 7 | RW | 0h | Config Type (CT): Indicates config request as config type 0 (1'b0) or config type 1 (1'b1). |



| Device: 0 Function: 0 Offset: 400800h | | | |
|--|------------|---------|--|
| Bits | Attributes | Default | Description |
| 6:5 | RW | 0h | <p>Format (FMT): Specifies transaction format together with TYPE[4:0]. The encodings for valid {FMT[1:0], TYPE[4:0]} are:</p> <p>{2b'00, 5b'0_0000}: Memory Read {2b'01, 5b'0_0000}: Memory Read with 64 bit extended address {2b'00, 5b'0_0001}: Reserved {2b'01, 5b'0_0001}: Reserved {2b'10, 5b'0_0000}: Memory Write {2b'11, 5b'0_0000}: Memory Write with 64 bit extended address {2b'00, 5b'0_0010}: IO Read {2b'10, 5b'0_0010}: IO Write {2b'00, 5b'0_0100}: Reserved {2b'10, 5b'0_0100}: Reserved {2b'00, 5b'0_0101}: Reserved {2b'10, 5b'0_0101}: Reserved {2b'00, 5b'0_1010}: Reserved {2b'10, 5b'0_1010}: Reserved {2b'00, 5b'0_1011}: Reserved {2b'10, 5b'0_1011}: Reserved {2b'01, 5b'1_OR[2:0]}: Message (R[2:0] are as defined in PCIe spec) {2b'11, 5b'1_OR[2:0]}: Message with Data. The data will come from the CCDW write.</p> |
| 4:0 | RW | 0h | <p>Type (TYPE): This field combined with Format field specifies the transaction type. See the encodings in FMT[1:0] field above.</p> |

§



14 Reset

14.1 Introduction

This chapter describes IOH-specific aspects of hardware reset. Subsequent I/O initialization procedures requiring configuration register dialogue are not described in this chapter.

14.1.1 Reset Types

- **Power-Up Reset**

Power-up reset is a special case of Power Good reset. It consists of energizing the power rails and involves the assertion of the COREPLLWRDET signal to the IOH.

- **Power Good Reset**

Power Good reset involves the deassertion of the COREPWRGOOD and AUXPWRGOOD signals, and is part of Power-up reset. Deassertion of COREPWRGOOD and AUXPWRGOOD can happen at any time, and is not necessarily associated with Power-up reset. The Power-Good reset spawns Intel QuickPath Interconnect, ESI, PCI Express, SMBus, and JTAG resets. “Surprise” Power-good reset clears program state, can corrupt memory contents, clears error logs, and so on, and therefore, should only be used as a last resort in extreme lock-up situations.

- **Hard Reset**

Hard reset involves the assertion of the CORERST_N signal, and is part of both Power-up and Power-Good reset. Hard reset is the “normal” component reset, with relatively shorter latency than either Power-up or Power-Good, particularly in its preservation of “sticky bits” (for example, error logs and power-on configuration (that is, Intel QuickPath Interconnect link initialization packet “4”s). Hard reset preserves hard partitions, and sets the phase on PLL post-dividers. The hard reset spawns Intel QuickPath Interconnect, ESI, PCI Express, and SMBus resets. “Surprise” hard reset clears program state, can corrupt memory contents, etc., and therefore, should only be used as a means to un-hang a system while preserving error logs which might provide a “trail” back to the “fault” that caused the “hang”. When a hard reset is issued via an external tool warm reset and calibration is bypassed via PHCTR, Intel QPI links do not train. This is due to ICOMP and RCOMP not being restarted after an external tool warm reset with calibration bypassed. When Intel QPI calibration is bypassed, a hard reset is not associated with Power-up or Core Powergood reset is not permitted. This may result in the Intel QPI links not training.

- **Intel QuickPath Interconnect PHY Layer Hard and Soft Reset**

There are two resets in the Intel QuickPath Interconnect PHY layer: hard and soft. Both resets only reset the PHY Layer of the Intel QuickPath Interconnect port. There are individual PHY hard and soft resets for each Intel QuickPath Interconnect port. PHY layer resets are completely orthogonal to Link layer resets. CSR bits with attribute type “P” and “PP” get reset on hard reset. CSR bits with attribute type “PP” get reset on soft reset. Refer to the Intel QuickPath Interconnect Specification for details on the differences and how soft/ hard resets are initiated.

If an Intel QuickPath Interconnect PHY hard or soft reset occurs when the Link Layer is active, the Link Layer will initiate a Link Layer Retry (LLR) to resend any Flits that were dropped during the PHY Layer reset. The Link and higher layer protocols will resume normal operation when the LLR is complete.



- **Intel QuickPath Interconnect Link Layer Reset**

There are two resets in the Intel QuickPath Interconnect Link Layer: hard and soft. Both resets only reset the Link Layer of the Intel QuickPath Interconnect port. There are individual link hard and soft resets for each Intel QuickPath Interconnect port. Link Layer resets are completely orthogonal to PHY Layer resets (except under special circumstances defined in the Intel QuickPath Interconnect specification section covering Link Layer Initialization). In the event that a Intel QuickPath Interconnect Link Layer reset occurs while a protocol layer packet is being processed by the Link Layer, the Intel QuickPath Interconnect provides no method for the protocol layer to recover. Therefore, in order to avoid data corruption, a Link Layer reset may only be asserted when the Intel QuickPath Interconnect port is idle. The difference between the Link Layer hard reset and Link Layer soft reset is the following: 1) Link Layer hard reset is initiated by a write to the Intel QuickPath Interconnect specification defined register bit (QPILCL[1]), which takes effect immediately, resulting in link layer re-init which clears all link layer state and resets all CSR bits with attribute type "N"; 2) Link Layer soft reset is initiated by a write to the Intel QuickPath Interconnect specification defined register bit (QPILCL[0]), which takes effect after 512 Intel QuickPath Interconnect core clock ("16UI") cycles, resulting in link layer re-init, which clears all link layer state and resets all CSR bits with attribute type "NN".

- **PCI Express Reset**

PCI Express reset combines a physical-layer reset and a link-layer reset for a PCI Express port. There are individual PCI Express resets for each PCI Express port. It resets the PCI Express port, for first initialization after power-up, exit from a power-off system-management sleep state, or such as a fault that requires an individual reset to un-hang a particular PCI Express port.

- **JTAG Reset**

JTAG reset resets only the JTAG port. JTAG reset does not reset any state that is observable through any other interface into the component (for example, CSRs, and so on).

- **SMBus Reset**

SMBus Reset resets only the slave SMBus controller. SMBus reset does not reset any state that is observable through any other interface into the component (for example, CSRs, and so on).

- **Intel® Trusted Execution Technology (MLINK Bus Reset)**

An MLINK reset resets only the MLINK controller.

- **RMII Bus Reset**

An RMII reset resets only the RMII controller.

- **Intel® Trusted Execution Technology (Intel® TXT) Reset**

Intel® Trusted Execution Technology (Intel® TXT) reset is a mechanism that stops the platform due to a security violation. It is a trigger for a HARD reset.

- **CPU Warm Reset (Supported by Intel Xeon processor 7500 series-based platform Only)**

A CPU can reset the CPUs and only the CPUs by setting the IOH.SYRE.CPURESET bit. System validation uses this feature to quickly initiate tests, averting the aeons



of test time required to navigate through an entire HARD reset. Setting the IOH.SYRE.CPURESET has no effect after LT.SENTER or while LT.SENTER.STS is set.

14.1.2 Reset Triggers

Possible triggers for reset:

- Energize power supplies
- COREPWRGOOD deassertion
- CORERST_N assertion with IOH.SYRE.RSTMSK = 0
- IOH.QPILCL[0]
- IOH.QPILCL[1]
QPILCL is a standard Intel QuickPath Interconnect control register
- Loss of Received Clock
- Receipt of Link Initialization Packet
- IOH.BCR.SRESET
BCR is a standard PCI Express control register
- TRST_N assertion or TCK/TMS protocol
- SMBus protocol
- MLRST_N assertion
- AUXPWRGOOD de-assertion
- RMII protocol
- IOH.SYRE.CPURESET

14.1.3 Trigger and Reset Type Association

Table 14-1 indicates Reset Triggers initiate each Reset Type.

Table 14-1. Trigger and Reset Type Association

| Reset Trigger | Reset Type |
|---------------------------------------|-----------------------------------|
| Energize Power Supplies | Power-Up |
| COREPWRGOOD signal deassertion | Power Good |
| CORERST_N assertion & IOH.SYRE.HARDEN | Hard |
| IOH.QPILCL[0] (Link Layer Hard Reset) | Link Intel QuickPath Interconnect |
| IOH.QPILCL[1] (Link Layer Soft Reset) | |
| Receipt of Link Initialization Packet | |
| IOH.BCR.SRESET | PCI Express |
| TRST_N assertion | JTAG |
| TCK/TMS protocol | |
| SMBus protocol | SMBus |

Note: Auxiliary power-up, power good, or Hard reset without an equivalent core reset is not allowed.



14.1.4 Domain Behavior

This is how each of the domains is treated during reset:

- Unaffected by reset:
 - PLLs
- Indirectly affected by reset:
 - Strap flip-flops:
Hold last value sampled before COREPWRGOOD assertion
 - Analog I/O compensation:
Only triggered by link power-up
- JTAG:
 - Asynchronous COREPWRGOOD deassertion, COREPWRGOOD deasserted, asynchronous TRST_N assertion, TRST_N asserted, or synchronous TCK/TMS protocol navigation to reset state: reset.
- SMBus:
 - Asynchronous COREPWRGOOD deassertion, COREPWRGOOD deasserted, hard reset assertion, or synchronous SMBus reset protocol: reset.
 - Signals are deasserted after hard reset assertion, signals are observable after hard reset deassertion
- Sticky configuration bits:

Per port, Intel QuickPath Interconnect Link layer bits except QPILCL.1 are sticky when the QPILCL.1 configuration bit is set.

Per port, Intel QuickPath Interconnect Physical-layer bits except QPIPHCL.1 are sticky with the QPIPHCL.1 configuration bit is set.

 - Asynchronous COREPWRGOOD deassertion, COREPWRGOOD deasserted: defaults.
 - (synchronized CORERST_N assertion or synchronized CORERST_N asserted) while COREPWRGOOD asserted: no-change.
- Tri-state-able outputs:
 - Asynchronous COREPWRGOOD deassertion, COREPWRGOOD deasserted: tri-state.
Place outputs in tri-state or electrical-disable when COREPWRGOOD is deasserted.
- PCI Express:
 - Asynchronous COREPWRGOOD deassertion, COREPWRGOOD deasserted: Electrical idle and reset
 - COREPWRGOOD asserted, one cycle after synchronized CORERST_N assertion, BCR.SRESET set: link down (one per port)
 - CORERST_N deassertion, BCR.SRESET cleared: initialize, train, link up
- Intel QuickPath Interconnect:
 - See [Section 14.1.6](#).
- ESI:
 - Asynchronous COREPWRGOOD deassertion, COREPWRGOOD deasserted: tri-state and reset.
 - COREPWRGOOD asserted, one cycle after synchronized CORERST_N assertion: link down



- CORERST_N deassertion: initialize, train, link up
- Processor CORERST_N cycle counter extinguished: send ESI.CPU_Reset_Done

14.1.5 Reset Sequences

Reset sequences are specified by a lexical “grammar”:

- “Trigger = trigger_list:”: one or more items from [Section 14.1.2, “Reset Triggers” on page 187](#).
- “{A, B,...}”: Logically “OR”-ed list
- Following the trigger list:
 - “Synchronous” indicates synchronous edge-sensitive trigger synchronized to a clock.
 - “Synchronized” indicates asynchronous edge-sensitive trigger synchronized to a clock.
 - “Asynchronous” indicates level-sensitive trigger asynchronous to any clock.
- “Condition” indicates trigger qualifier.
Condition must be valid for trigger to be recognized.

A bulleted list of “actions” follow the trigger and condition specifications:

- “->” within bullet: sequential execution.

14.1.5.1 Power-Up

Trigger = energize power supplies and stabilize master clock inputs with COREPWRGOOD deasserted ->

- -> Toggle clock trees.

14.1.5.2 COREPWRGOOD Deasserted

Trigger = COREPWRGOOD deassertion: asynchronous ->.

Condition = Power and master clocks remain stable.

- -> Tri-state other I/O.
- -> Toggle PLL outputs.
- -> Reset I/O Ports.

14.1.5.3 COREPLLWRDET Assertion

Trigger = COREPLLWRDET assertion: asynchronous ->

- -> PLLs acquire lock ->.
- -> PCI Express, ESI, and Intel QuickPath Interconnect interfaces complete calibration.
Calibration is initiated as soon as internal clocks are stable.

14.1.5.4 COREPWRGOOD Assertion

Trigger = COREPWRGOOD assertion: synchronized ->.

Condition = Voltages are within specifications. Master clocks are stable. The TCK signal may be in any state.

- -> Sample straps.



- -> Un-tri-state I/O.
- -> Hold Intel QuickPath Interconnect, PCI Express, ESI links down.

14.1.5.5 Hard Reset Asserted

Trigger = CORERST_N assertion: synchronized ->.

Condition = COREPWRGOOD is asserted.

Consequence = All buffered writes may be dropped.

- -> Protect sticky configuration bits.
- -> Synchronously assert internal asynchronous flip-flop initialization inputs.
- -> Private JTAG chains may be reset.
- -> Reset Intel QuickPath Interconnect, PCI Express and ESI protocol -> Take PCI Express and ESI links down.

14.1.5.6 Hard Reset Deassertion

Trigger = CORERST_N deassertion: synchronized ->.

Condition = COREPWRGOOD is asserted.

Consequence = Inputs from buses tri-stated during reset are masked until it is guaranteed that bus values are electrically and logically valid.

- -> Allow normal operation of sticky configuration bits.
- -> Initialize Intel QuickPath Interconnect, PCI Express and are links -> Engage Intel QuickPath Interconnect, PCI Express and are link training -> Bring Intel QuickPath Interconnect, PCI Express and ESI links up.

14.1.5.7 IOH PCI Express* Reset Asserted

Trigger = BCR.SRESET set: synchronous ->.

- -> Initialize PCI Express protocol -> Take PCI Express link down.

14.1.5.8 IOH PCI Express Reset Deasserted

Trigger = BCR.SRESET cleared: synchronous ->

- -> Initialize PCI Express link -> Engage PCI Express link training -> Bring PCI Express link up.

14.1.5.9 Intel QuickPath Interconnect Link Reset Assertion

Triggers = {IOH.QPILCL[0] (Link Layer Hard Reset) set to '1'; IOH.QPILCL[1] (Link Layer Soft Reset) set to '1'; Receive Link Training Packet}: synchronous ->

->Take Intel QuickPath Interconnect link to link initialization

14.1.5.10 Intel QuickPath Interconnect Link Reset De-Assertion

There are no persistent Intel QuickPath Interconnect Link Reset sources. The Intel QuickPath Interconnect Link Reset proceeds through link initialization to full Intel QuickPath Interconnect protocol operation upon detection of Intel QuickPath Interconnect Link Reset Assertion.



14.1.5.11 JTAG Reset Assertion

Triggers = {TRST_N assertion: asynchronous; TMS asserted for five TCK rising edges: synchronous}

->.

- -> Initialize JTAG.

14.1.5.12 JTAG Reset Deassertion

Trigger = TRST_N deassertion: asynchronous ->.

- -> Release JTAG port and to operate normally.

14.1.5.13 CLINK Reset Assertion (Intel Xeon Processor 7500 Series based Platform Only)

Triggers = CLRST_N assertion: asynchronous ->

- -> Initialize CLINK port.

14.1.5.14 CLINK Reset De-Assertion (Intel Xeon Processor 7500 Series based Platform Only)

Trigger = CLRST_N de-assertion: asynchronous ->

- -> Release CLINK port and to operate normally.

14.1.5.15 SMBus Reset Sequence

Trigger = SMBus: synchronous ->.

- -> Reset SMBus interface.

14.1.5.16 RMI Reset Sequence (Intel Xeon Processor 7500 Series based Platform Only)

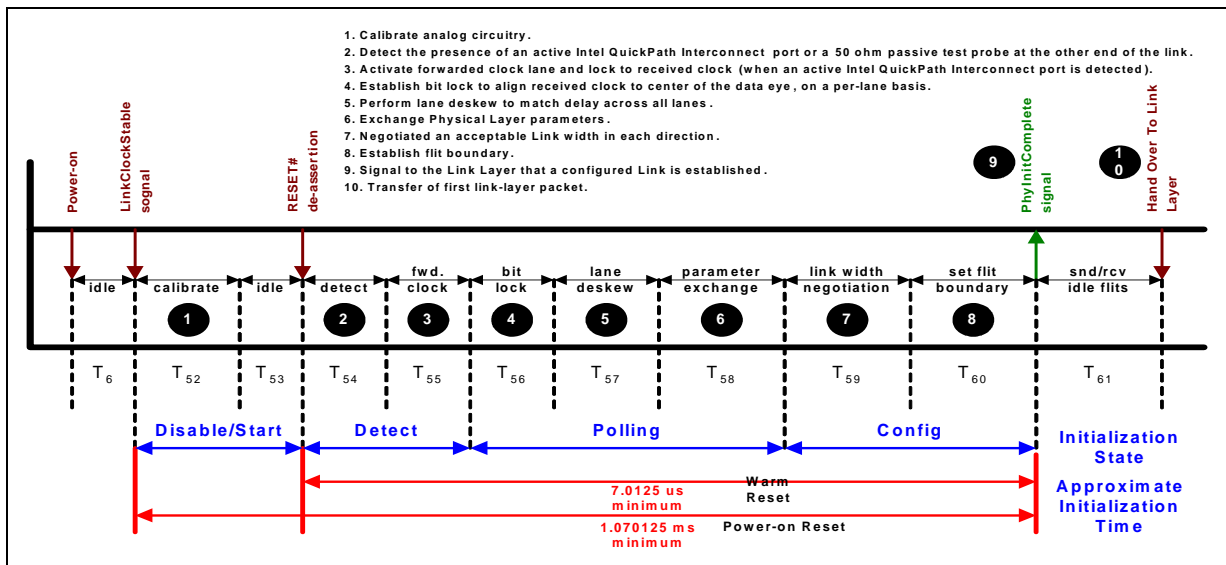
Trigger = RMI protocol: synchronous ->

- -> Reset RMI interface

14.1.6 Intel QuickPath Interconnect Reset

The Link Training and Status State Machine (LTSSM) as well as any self test, BIST, or loopback functions in the Intel QuickPath Interconnect link must be built around and compatible with the IOH's reset protocols, which includes core logic, power supply sequencing, and the calibration of other analog circuits such as PCI Express I/O, ESI I/O, and PLLs. Platform determinism is enforced at the CPU socket.

Figure 14-1. Physical Layer Power-Up and Initialization Sequence



14.1.6.1 Inband Reset

An inband reset mechanism is used for Intel QuickPath Interconnect Soft Reset. The inband reset is initiated by stopping a Forwarded Clock, and detected by observing the absence of an expected Received Clock transition, ultimately resulting in a link failure.

An inband reset is not a power-up link reset. An inband reset is only defined for a link that is up and running. Loss of Forwarded Clock prior to the completion of the first detect state after power-up will not result in an inband reset.

Figure 14-2. Inband Reset Sequence Initiated by Port A to Port B

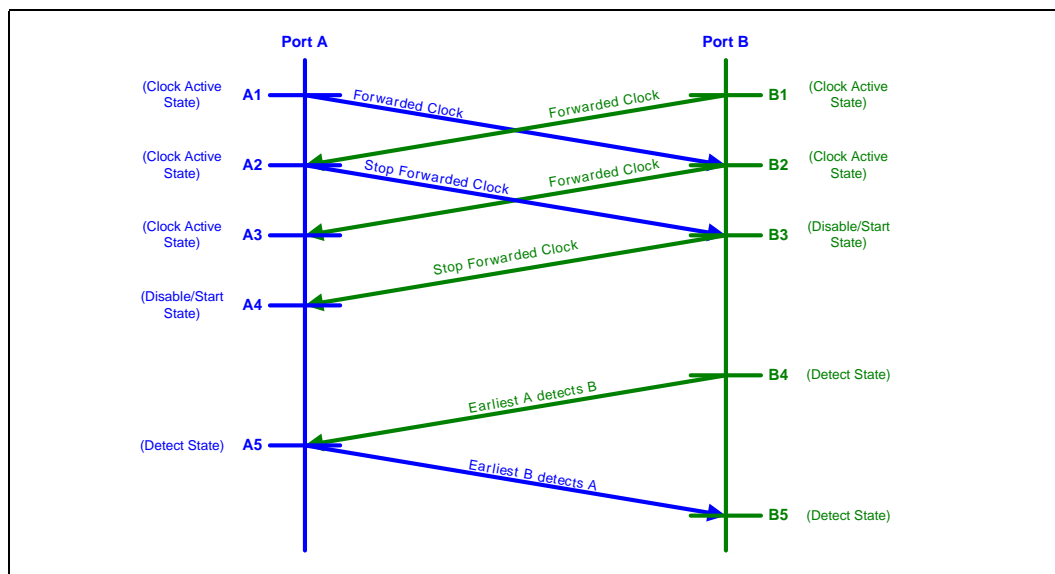




Table 14-2. Intel QuickPath Interconnect Inband Reset Events

| Port A | Port B |
|--|--|
| A1: Port A is in a clock-active state (a state other than Disable/Active or Detect.1). Forwarded Clock is currently being transmitted and/or Received Clock is being received. | B1: Port B is in a state other than Disable/Active or Detect.1. Forwarded Clock is currently being transmitted and/or Received Clock is being received. |
| A2: Port A sends an Inband Reset to Port B by stopping forwarded clock. Simultaneously, Port A stops driving data lanes as well, but Port A receivers continue to observe Received Clock and accepts incoming data. | B2: Port B is still in a clock-active state as Inband Reset is in flight. Port B continues to send forwarded clock (and data) to Port A. |
| A3: This is the same as event A2 , as Port A continues to observe Received Clock. | B3: Port B interprets loss of Received Clock as an Inband Reset. Port B immediately stops driving Forwarded Clock and data lanes. Port B enters the Disable/Start state. |
| A4: Port A loses Received Clock from Port B. Port A interprets this as an acknowledgement to the Inband reset that it initiated in event A2 . Port A enters the Disable/Start state. | <p>B4: If Calibration is bypassed, Port B allows a minimum period of $T_{\text{INBAND_RESET_INIT}}$ to elapse after event B3 before asserting the <i>PhyInitBegin</i> signal, which advances Port B to the Detect.1 state.</p> <p>If Calibration is forced, Port B allows the minimum period of $T_{\text{INBAND_RESET_INIT}}$ to elapse after completion of calibration after event B3 before asserting the <i>PhyInitBegin</i> signal and advancing to the Detect.1 state.</p> <p>Upon entering the Detect.1 state, Port B commences the process of detecting Port A.</p> <p>The $T_{\text{INBAND_RESET_INIT}}$ parameter is defined to be much longer than the time of flight, so Port A is guaranteed to be in the Disable/Start state by the time Port B advances to the Detect.1 state. This time-out avoids any false detection of Port A (by Port B) when Inband Reset is in flight.</p> |
| <p>A5: If Calibration is bypassed, Port A allows a minimum period of $T_{\text{INBAND_RESET_INIT}}$ to elapse after event A4 before asserting the <i>PhyInitBegin</i> signal, which advances Port A to the Detect.1 state.</p> <p>If Calibration is forced, Port A allows the minimum period of $T_{\text{INBAND_RESET_INIT}}$ to elapse after completion of calibration after event A4 before asserting the <i>PhyInitBegin</i> signal and advancing to the Detect.1 state.</p> <p>Upon entering the Detect.1 state, Port A commences the process of detecting Port B, and resumes driving Forwarded Clock after Port B is detected.</p> | B5: This is the earliest event in which Port B can detect Port A. When Port B detects Port A, Port B resumes driving Forwarded Clock. |

14.2 Platform Signal Routing Diagram

Figure 14-3. Basic Reset Distribution

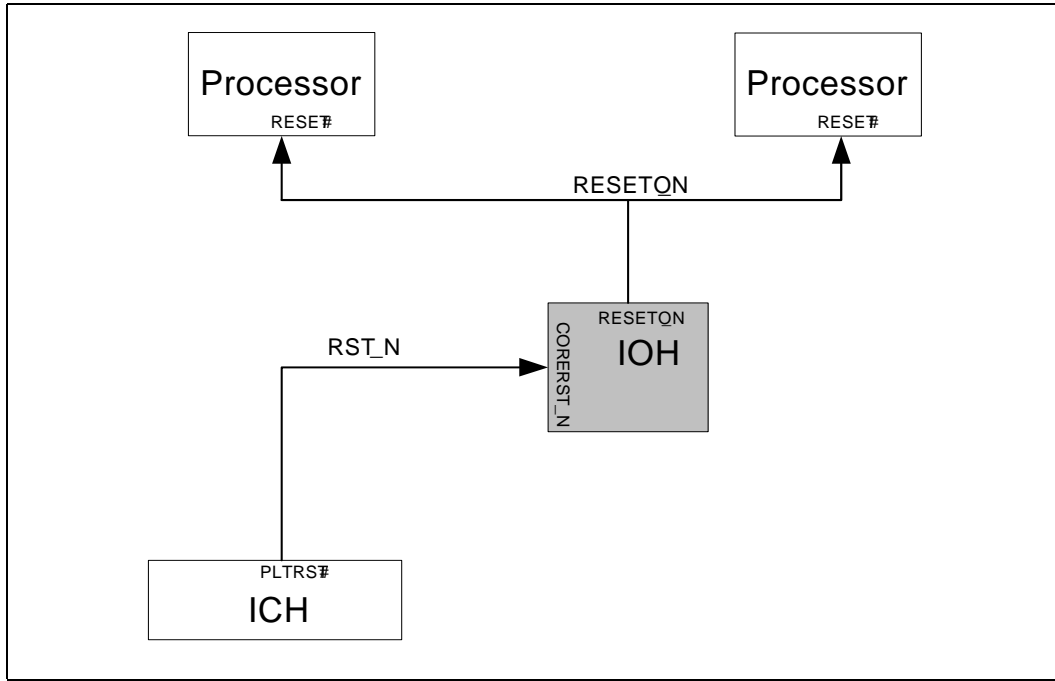
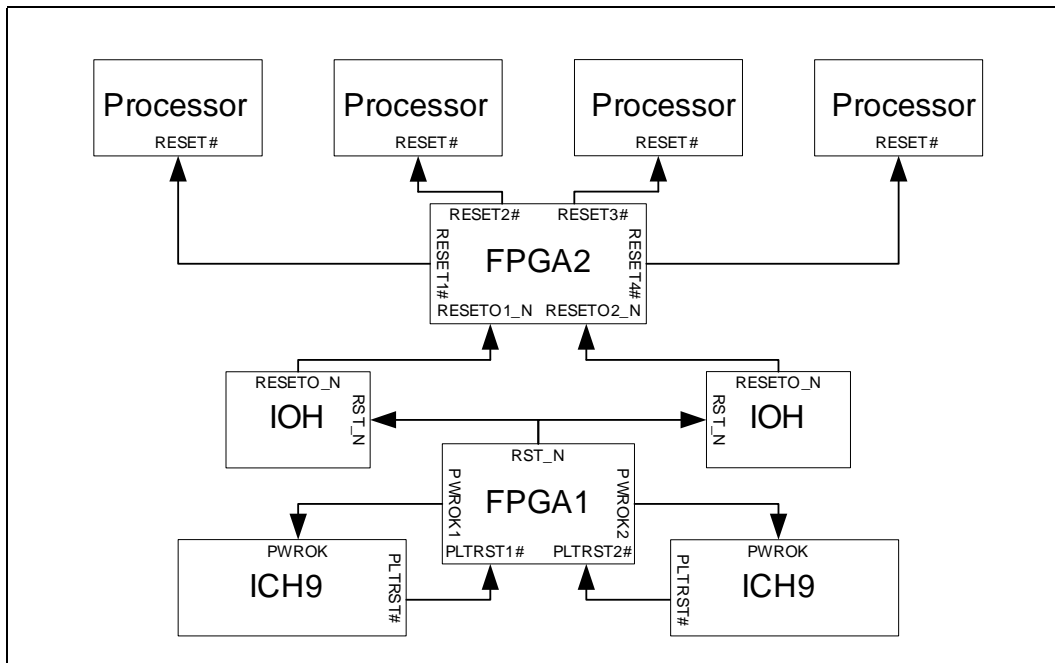


Figure 14-4. Basic MP System Reset Distribution





14.3 Platform Timing Diagrams

The following diagrams represent platform reset timing sequences without BMC presence. With a BMC:

- Intel QuickPath Interconnect link initialization may stall indefinitely between:
 - Completion of physical initialization, and
 - Transfer of the first outbound link initialization packet.Stall is not allowed if the Physical layer was initialized but the Link layer was not.

Figure 14-5. Power-Up (example)

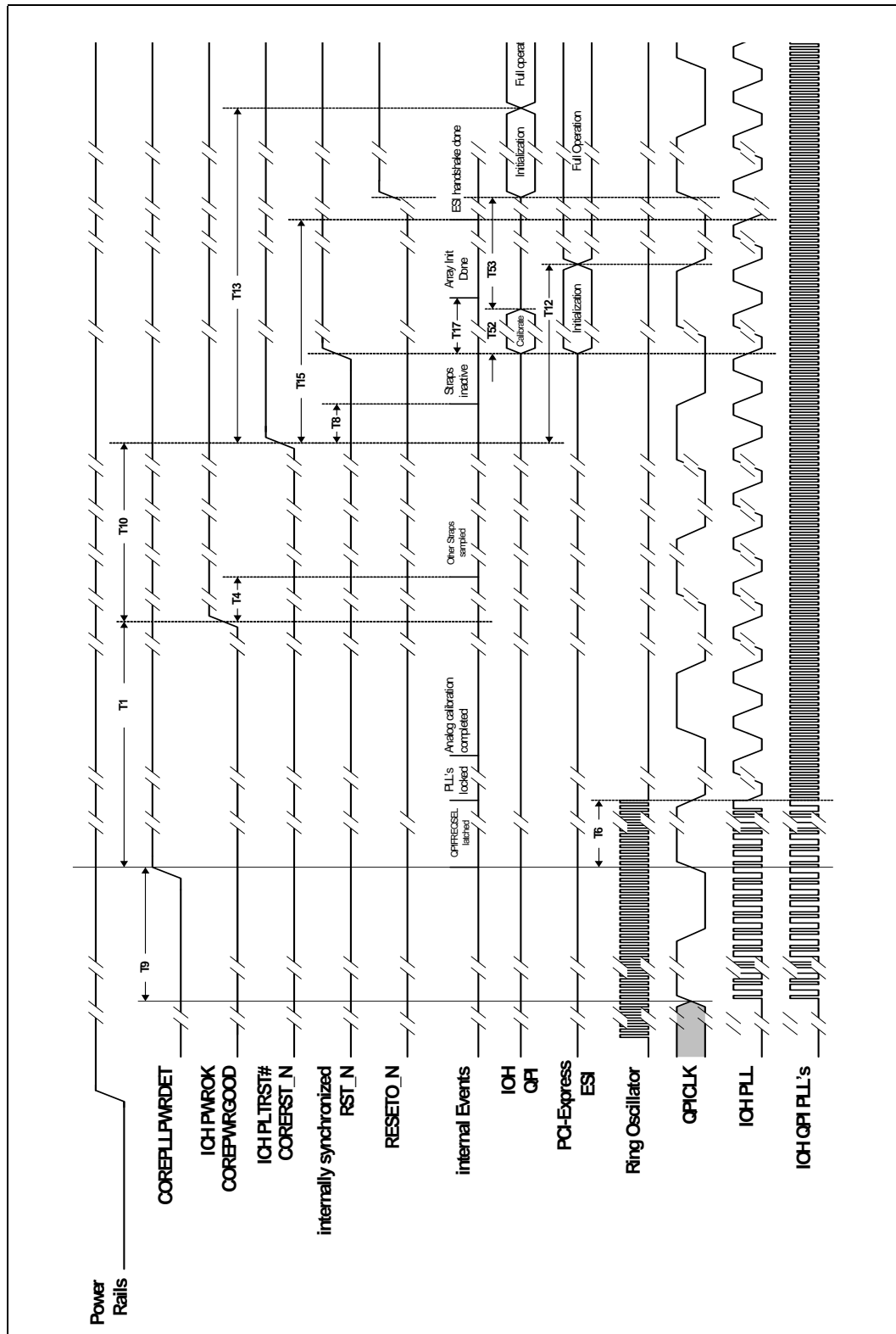




Figure 14-6. COREPWRGOOD Reset (example)

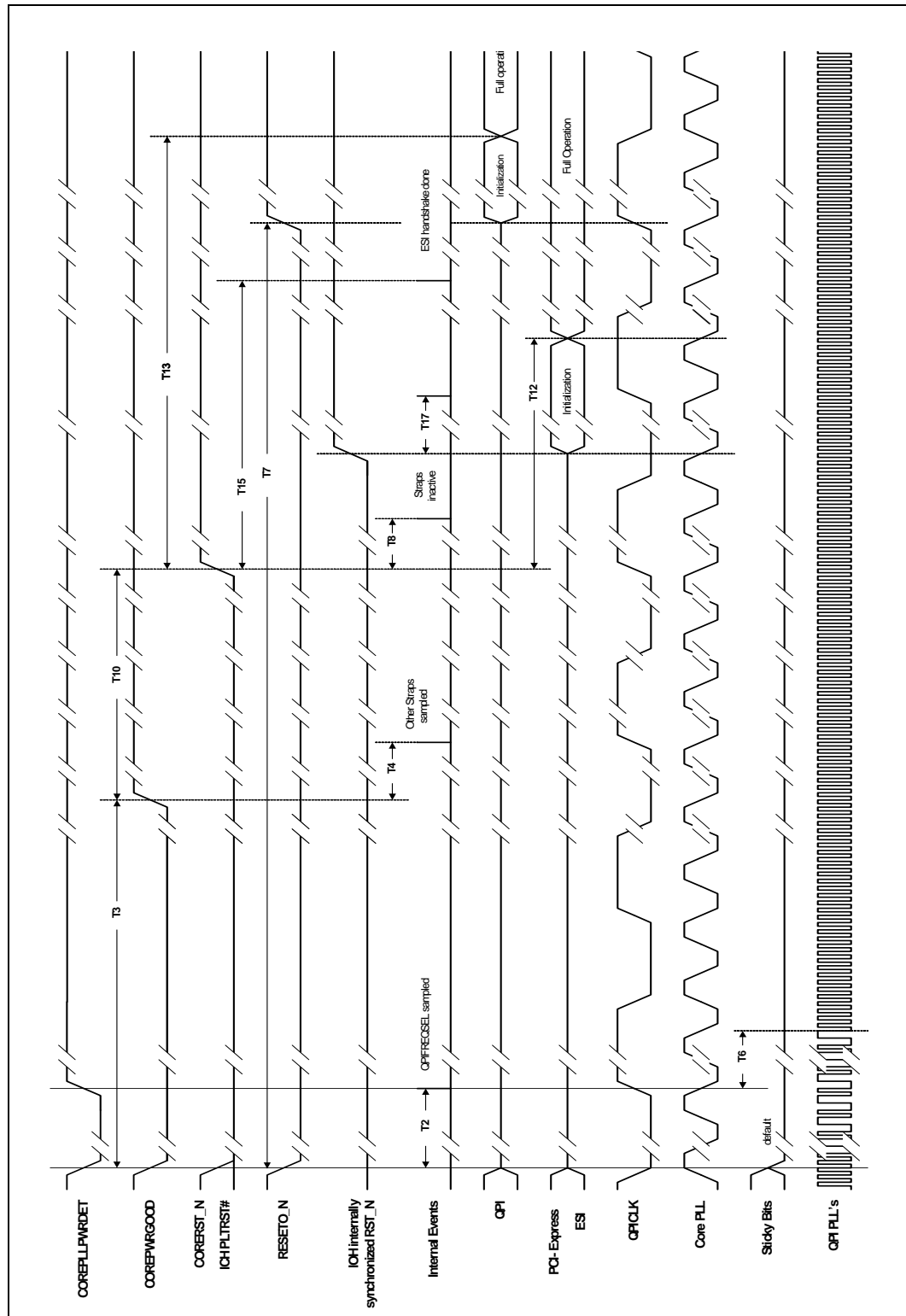


Figure 14-7. Hard Reset (Example)

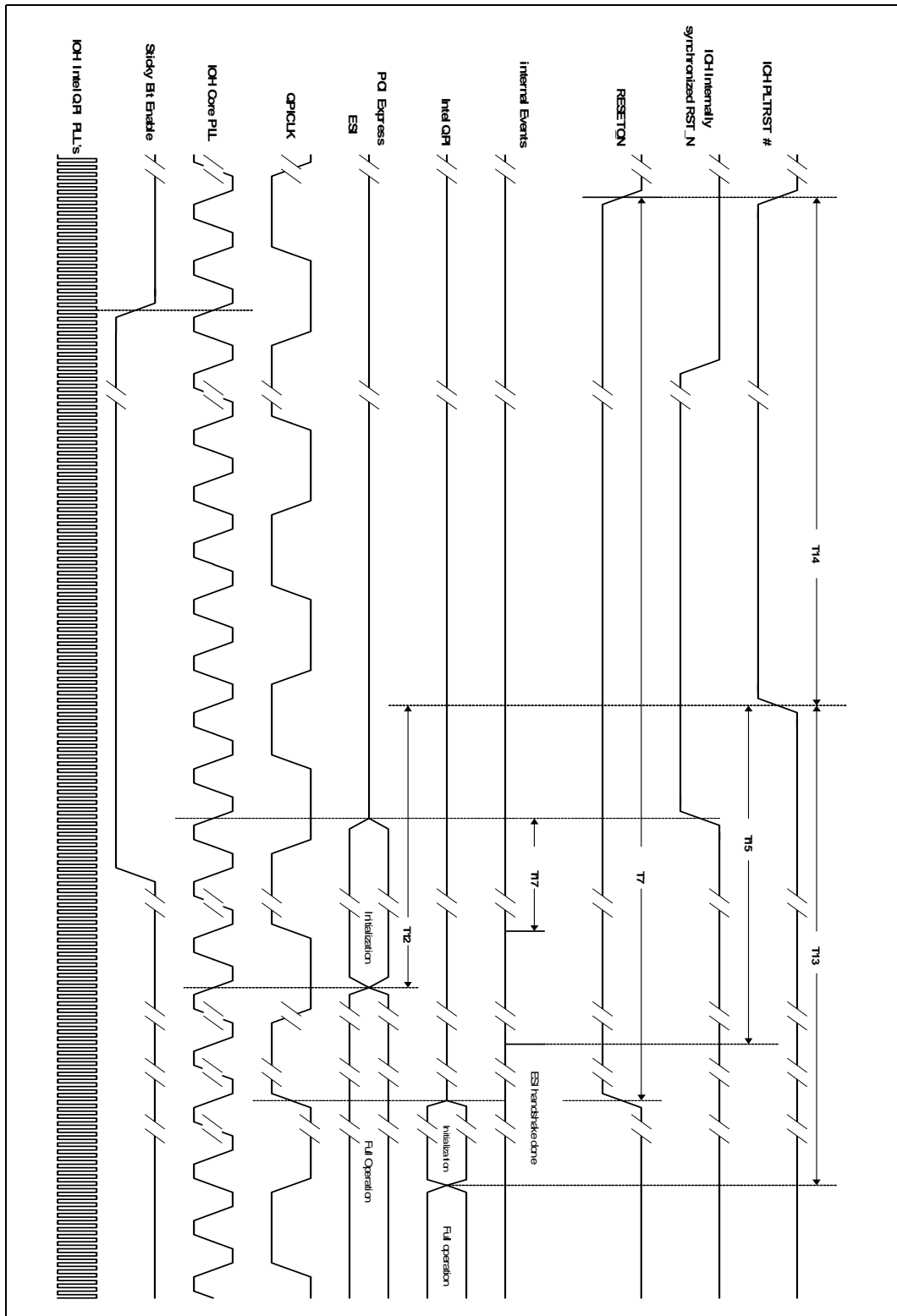




Figure 14-8. IOH CORERST_N Re-Trigging Limitations

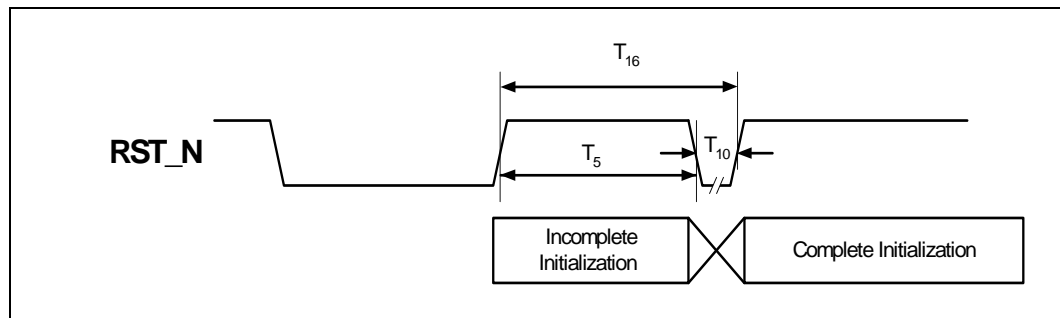


Table 14-3 specifies the timings drawn in Figure 14-5, Figure 14-6, Figure 14-7, and Figure 14-8. Nominal clock frequencies are described. Specifications still hold for de-rated clock frequencies.

Table 14-3. Core Power-Up, Core POWERGOOD, and Core Hard Reset Platform Timings (Sheet 1 of 2)

| Timing | Description | Min | Max | Comments |
|--------|---|------------|---------|--|
| T1 | COREPLLWRDET signal assertion to COREPWRGOOD signal assertion | 100 us | 15 ms | Min/Max timing applies to PLLWRDET signal assertion to AUXPWRGOOD signal assertion when Intel ME AUX power uses separate power source. |
| T2 | COREPWRGOOD de-assertion to straps stable | | 40 ns | |
| T3 | COREPWRGOOD de-assertion | 80 ns | | Minimum COREPWRGOOD de-assertion time while power and platform clocks are stable. |
| T4 | COREPWRGOOD assertion to straps sampled | | 0 ns | |
| T5 | CORERST_N de-assertion to CORERST_N assertion | 50 QPICKLs | | Minimum CORERST_N re-trigger time. |
| T6 | PLLWRDET assertion or CORERST_N assertion or stable strap on COREPWRGOOD de-assertion to PLL lock acquisition | 960 ns | | COREPWRGOOD de-assertion only requires Intel QuickPath Interconnect PLL lock re-acquisition because their frequency is determined by a strap. CORERST_N assertion only requires Intel QuickPath Interconnect PLL lock re-acquisition when the FREQ register changed. |
| T7 | RESET0_N duration | | | N/A |
| T8 | CORERST_N de-assertion to straps inactive | 12 ns | 18 ns | Strap Hold Time |
| T9 | Reference clock stable to PLLWRDET signal assertion | 100 ns | | Min/Max timing applies to PLLWRDET when Intel ME AUX power uses separate power source. |
| T10 | COREPWRGOOD assertion to CORERST_N de-assertion | 1 ms | | During Core Power Cycling |
| T12 | CORERST_N signal de-assertion to completion of PCI Express initialization sequence | | 12.5 ms | |
| T14 | CORERST_N assertion to CORERST_N de-assertion | 2.1 us | | |
| T15 | CORERST_N signal de-assertion to completion of ESI reset sequence | | 100 us | ICH specification |
| T16 | CORERST_N re-trigger delay | T5+T10 | | |
| T17 | SMBus delays w/r/t CORERST_N | 3 QPICKLs | | |

**Table 14-3. Core Power-Up, Core POWERGOOD, and Core Hard Reset Platform Timings (Sheet 2 of 2)**

| Timing | Description | Min | Max | Comments |
|--------|--|---------|-----|--|
| T52 | Intel QuickPath Interconnect Calibration | 1 ms | | |
| T54 | Intel QuickPath Interconnect Detect | 20 ns | | Min = processor is ready before IOH |
| T55 | Intel QuickPath Interconnect Activate Forwarded Clock | 3.9 us | | |
| T56 | Intel QuickPath Interconnect bit lock | 1.25 us | | |
| T57 | Intel QuickPath Interconnect bit-lane deskew | 98 ns | | |
| T58 | Intel QuickPath Interconnect physical parameter exchange | 98 ns | | |
| T59 | Intel QuickPath Interconnect link width negotiation | 98 ns | | |
| T60 | Intel QuickPath Interconnect set flit boundary | 10 ns | | |
| T61 | Intel QuickPath Interconnect wait for link initialization stall to clear | 150 ms | | Min = all stall conditions cleared. Starts at de-assertion of CORERST_N. Quote from TWD was 15us, allowing 10,000x guardband for S/W delays. |

§



15 Component Clocking

15.1 Component Specification

15.1.1 Reference Clocks

The QPI{0/1}CLK reference clock is the core PLL reference clock, operating at 133 MHz; the reference clock frequency is common between all Intel QuickPath Interconnect agents. This mesochronous reference clock requires no phase matching between agents, tolerating zero ppm frequency offset between agents.

The PE{0/1}CLK is the reference clock supplied to the IOH for PCI Express and ESI interfaces, and operates at 100 MHz.

The Intel QuickPath Interconnect interface's phit frequency domain is derived from the QPI{0/1}CLK reference clocks.

When QPIxCLK spectrum spreading is disabled, the PCI Express links can operate mesochronously (zero ppm frequency tolerance between PCI Express agents) or plesiochronously (a few hundred ppm frequency tolerance between PCI Express agents) while the PExCLK and QPIxCLK domains operate mesochronously.

When QPIxCLK spectrum spreading is enabled, a PCI Express link operating plesiochronously (because both ends' master clocks are derived from the different oscillators) prevents ratioed PExCLK:QPIxCLK domains, necessitating "asynchronous" data transfer between domains.

Asynchronous PExCLK reference clocks may be derived from different oscillators.

15.1.2 JTAG

The JTAG clock, TCK, is asynchronous to core clock. For private TAP register accesses, one TCK cycle is a minimum of 12 core cycles. The TCK high time is a minimum of 6 core cycles in duration. The TCK low time is a minimum of 6 core cycles in duration.

For public TAP register accesses, TCK operates independently of the core clock.

15.1.3 SMBus

The SMBus clock is synchronized to the Intel QuickPath Interconnect core clock. Data is driven into the IOH with respect to the serial clock signal. Data received on the data signal with respect to the clock signal will be synchronized to the core. The serial clock can not be active until 10 mS after RST_N de-assertion. When inactive, the serial clock should be de-asserted (High). The serial clock frequency is 100 KHz.

15.1.4 Hot-Plug Serial Buses

The PCI Express hot-plug Virtual Pin Interface clock is a dedicated SMBus interface; the PCI Express Hot-Plug signals reside on this serial interface. The serial clock frequency is 100 KHz. This clock is not available during standby.



15.1.5 RMI I Bus

The RMI I reference clock frequency is 50 MHz, which is available during standby.

If the RMI I bus is not used, the RMI I reference clock input pin can be tied to ground through a resistor.

15.1.6 CLINK Bus

The CLINK reference clock frequency is 66 MHz, which is available during standby.

15.1.7 Intel Management Engine Clock

The Intel Management Engine (Intel ME) can handle a reference clock frequency of 133 MHz or 100 MHz which must be provided for Intel ME operation. The Intel ME's 200 MHz DDRCLK[P/N] clock output is derived from this reference clock.

For a non-Intel ME configuration it is optional to omit the reference clock to the Intel ME with certain restrictions. While this configuration may slightly improve platform BOM for non-Intel ME-based systems, it is not recommended unless absolutely sure that no future desire to operate Intel ME firmware on this platform will arise.

Table 15-1. The Clock Options for a Intel ME and Non-Intel ME Configuration System

| Reference clk Source | Intel ME-USED System | Intel ME-UNUSED System | Notes |
|----------------------|--------------------------------|------------------------|---|
| 133 MHz | Support | Support | DDRFREQ[3:2] tied to '00 |
| 100 MHz | Support | Support | DDRFREQ[3:2] tied to '01 |
| No CLK | Configuration is not supported | Support | This configuration is only for ME-Unused system. ME_CLK_SRC must be tied to 0 if no reference clock is provided. |

15.1.8 Clock Pin Descriptions

Table 16-1. Clock Pins (Sheet 1 of 2)

| Pin Name | Pin Descriptions |
|--------------|---|
| QPIOREFCLKP | Intel QuickPath Interconnect 0-interface reference clock |
| QPIOREFCLKN | Intel QuickPath Interconnect 0-interface reference clock (complement) |
| QPI1CLKP | Intel QuickPath Interconnect 1-interface reference clock |
| QPI1CLKN | Intel QuickPath Interconnect 1-interface reference clock (complement) |
| QPI0TPCLK[0] | Intel QuickPath Interconnect 0 Transmitter forwarded clock 0 |
| QPI0TPCLK[1] | Intel QuickPath Interconnect 0 Transmitter forwarded clock 1 |
| QPI0TNCLK[0] | Intel QuickPath Interconnect 0 Transmitter forwarded clock 0 (complement) |
| QPI0TPCLK[1] | Intel QuickPath Interconnect 0 Transmitter forwarded clock 1 (complement) |
| QPI1TPCLK[0] | Intel QuickPath Interconnect 1Transmitter forwarded clock 0 |
| QPI1TPCLK[1] | Intel QuickPath Interconnect 1Transmitter forwarded clock 1 |
| QPI1TNCLK[0] | Intel QuickPath Interconnect 1Transmitter forwarded clock 0 (complement) |
| QPI1TPCLK[1] | Intel QuickPath Interconnect 1Transmitter forwarded clock 1 (complement) |
| QPIORPCLK[0] | Intel QuickPath Interconnect 0 Receiver forwarded clock 0 |



Table 16-1. Clock Pins (Sheet 2 of 2)

| Pin Name | Pin Descriptions |
|--------------|--|
| QPI0RPCLK[1] | Intel QuickPath Interconnect 0 Receiver forwarded clock 1 |
| QPI0RNCLK[0] | Intel QuickPath Interconnect 0 Receiver forwarded clock 0 (complement) |
| QPI0RPCLK[1] | Intel QuickPath Interconnect 0 Receiver forwarded clock 1 (complement) |
| QPI1RPCLK[0] | Intel QuickPath Interconnect 1 Receiver forwarded clock 0 |
| QPI1RPCLK[1] | Intel QuickPath Interconnect 1 Receiver forwarded clock 1 |
| QPI1RNCLK[0] | Intel QuickPath Interconnect 1 Receiver forwarded clock 0 (complement) |
| QPI1RPCLK[1] | Intel QuickPath Interconnect 1 Receiver forwarded clock 1 (complement) |
| PE0CLKP | PCI Express 0-interface clock |
| PE0CLKN | PCI Express 0-interface clock (complement) |
| PE1CLKP | PCI Express 1-interface clock |
| PE1CLKN | PCI Express 1-interface clock (complement) |
| XDPSTBP_N | Debug Port strobe |
| XDPSTBN_N | Debug Port strobe (complement) |
| XDPCLK1X | 1X XDP clock |
| TCK | TAP clock |
| PEHPSCL | PCI Express hot-plug Virtual Pin Interface clock |
| SMBUSCL | SMBus clock |
| DDR_REFCLK_P | Intel Management Engine clock |
| DDR_REFCLK_N | Intel Management Engine clock (complement) |
| CLCLK | CLINK clock |
| RMII_CLK | RMII clock |
| DDRCLK_P | DDR clock |
| DDRCLK_N | DDR clock (complement) |
| DDRCKE | DDR clock enable |
| DDREDQS | DDR data strobe |
| DDREDQS_N | DDR data strobe (complement) |

15.1.9 High Frequency Clocking Support

15.1.9.1 Spread Spectrum Support

The IOH supports Spread Spectrum Clocking (SSC). SSC is a frequency modulation technique for EMI reduction. Instead of maintaining a constant frequency, SSC modulates the clock frequency/period along a predetermined path, that is, the modulation profile. The IOH supports a nominal modulation frequency of 30 KHz with a downspread of 0.5%.

15.1.9.2 Stop Clock

PLLs in the IOH cannot be stopped.



15.1.9.3 Jitter

Intel QuickPath Interconnect strongly recommends that QPIxCLK cycle-to-cycle jitter delivered to the package ball should be less than or equal to 50 ps (± 25 ps).

PExCLK jitter must be less than 150 ps (± 75 ps)

15.1.9.4 Forwarded Clocks

“Forwarded clocks” are not clocks in the normal sense. Instead, they act as constantly-toggling bit-lanes which supply UI phase information to all associated bit-lane receivers in the channel removing the phase error between the transmitter and receiver due to long-term jitter. The Intel QuickPath Interconnect clocks utilize low-speed (133 MHz) reference clocks for the primary input of their I/O PLLs.

15.1.9.5 External Reference

An external crystal oscillator is the preferred source for the PLL reference clock. A spread spectrum frequency synthesizer that meets the jitter input requirements is recommended.

15.1.9.6 PLL Lock Time

The assertion of the PWRGOOD signal initiates the PLL lock process.

15.1.9.7 Analog Power Supply Pins

Each PLL requires an analog Vcc and Analog Vss pad and external LC filter. The filter is NOT to be connected to the system board Vss. The ground connection of the filter is grounded to on-die Vss.

15.2 Miscellaneous Requirements and Limitations

A reference clock must always be supplied to QPIOCLK{P/N}. A reference clock must be supplied to QPI1CLK{P/N} when a processor is connected to Intel QuickPath Interconnect port 1.

§



16 Reliability, Availability, Serviceability (RAS)

16.1 RAS Overview

This chapter describes the features provided by the IOH for the development of high Reliability, Availability, Serviceability (RAS) systems. RAS refers to three main features associated with system's robustness. These features are summarized as follows:

Reliability: Refers to how often errors occur in the system, and whether the system can recover from an error condition.

Availability: Refers to how flexible the system resources can be allocated or redistributed for the system utilizations and system recovery from errors.

Serviceability: Refers to how well the system reports and handles events related to errors, power management, and hot-plug.

IOH RAS features aim to achieve the following:

- Soft, uncorrectable error detection and recovery on PCI Express and Intel QuickPath Interconnect links.
 - CRC is used for error detection and error recovered by packet retry.
- Clearly identify non-fatal errors whenever possible and minimize/eliminate fatal errors.
 - Synchronous error reporting of the affected transactions by the appropriate completion responses or data poisoning.
 - Asynchronous error reporting for non-fatal and fatal errors via inband messages or outband signals.
 - Enable software to contain and recover from errors.
 - Error logging/reporting to quickly identify failures, contain and recover from errors.
- PCI Express hot-plug (add/remove) to provide better serviceability

IOH RAS features can be grouped into five categories. These features are summarized below and detailed in the subsequent sections:

1. System level RAS

- a. Platform or system level RAS for inband and outband system management features.
- b. Dynamic partitioning and on-line hot add/remove for serviceability.

2. IOH RAS

- a. IOH RAS features for error detection, logging, and reporting.

3. Intel® QuickPath Interconnect RAS

- a. Standard Intel® QuickPath Interconnect RAS features as specified in the Intel® QuickPath Interconnect specification.



4. PCI Express RAS

- a. Standard **PCI Express** RAS features as specified in the **PCI Express** base specification.

5. Hot-Plug (Add/Remove)

- a. CPU, memory, and PCI Express hot-plug (add/remove) support.

16.2 System Level RAS

System level RAS features include the following:

1. Boot processor
2. Inband system management by processor in CM/SMM mode.
3. Outband system management from SMBus by Baseboard Management Controller (BMC).
4. On-Line dynamic hard partitioning.
5. System-Level Debug features.

16.2.1 Boot Processor

IOH is capable of booting from either Intel QuickPath Interconnect ports on the legacy IOH. IOH will advertise firmware agent on the legacy IOH if the firmware strap is set. This allows either of the directly connected processors to fetch flash. The processors may then use a semaphore register in the IOH to determine which processor is designated as the boot processor.

16.2.2 Inband System Management

Inband system management is accomplished by firmware running in high privileged mode. In the event of an error, fault, or hot add/remove, firmware is required to determine the system's condition and service the event accordingly. Firmware may enter CM mode for these events, so that it has the privilege to access the OS invisible configuration registers.

16.2.3 Outband System Management

Outband system management relies on the out-of-band agents to access system configuration registers via outband signals. The outband signals, such as SMBus and JTAG, are assumed to be secured and have the right to access all CSRs within a component. This includes the QPI configuration (QPICFG) registers and PCIe configuration (PCICFG) registers; however, SMBus/JTAG accesses outside of QPICFG or PCICFG space are not permitted.

Both SMBus and JTAG are connected globally to CPUs, IOHs, and ICH – through a shared bus hierarchy for SMBus, or through a serial bit chain for JTAG. By using the outband signals, an outband agent is able to handle events such as hot-plug, partitioning, or error recovery. Outband signals provide the BMC a global path to access the CSRs in the system components, even when the CSRs become inaccessible to processors through the inband mechanisms. Externally, the SMBus is mastered by the BMC and JTAG is controlled by a platform specific mechanism.

To support outband system management, the IOH provides both SMBus and JTAG interfaces. Either interface can access the CSR registers in the IOH (QPICFG and PCICFG) or in the downstream I/O devices (PCICFG).



16.2.4 Dynamic Partitioning

Dynamic partitioning refers to the addition or removal of system components to/from a partition without shutting down the affected partitions. Dynamic partitioning provides greater RAS features:

- Provides greater availability by dynamically allocating and dividing system resources for partitioning.
- Provides greater serviceability through partition isolation and allowing hot-plug (add/remove) of system resources within a partition.
- Provides greater reliability by containing the errors within the partition and not affecting operation of another partition.

Support of dynamic partitioning requires hot add/remove capability. The IOH provides CSR registers for partitioning support.

16.3 IOH RAS Support

The IOH core RAS features are summarized below and detailed in subsequent sections.

1. Error detection of the IOH internal data path and storage structures.
2. Detection, correction, logging, and reporting of system errors and faults.

16.4 IOH Error Reporting

The IOH logs and reports detected errors via “system event” generations. In the context of error reporting, a system event is an event that notifies the system of the error. Two types of system events can be generated - an inband message to the processor, or an outband signal assertion to the platform. In the case of inband messaging, the processor is notified of the error by the inband message (interrupt, failed response, and so on). The processor responds to the inband message and takes the appropriate action to handle the error. Outband signaling (Error Pins and Thermalert_N and Thermtrip_N) informs an external agent of the error events. An external agent such as an SSP or BMC may collect the errors from the error pins to determine the health of the system, sending interrupts to the processor, accordingly. In some severe error cases, when the system no longer responds to inband messages, the outband signaling provides a way to notify the outband system manager of the error. The system manager can then perform a system reset to recover the system functionality.

[Figure 16-1](#) and [Figure 16-2](#) shows examples that the IOH receives PCIe error messages from downstream IO devices, contrasts inband and outband error reporting, and log the errors. On the [Figure 16-1](#), the error is converted to an inband interrupt to the CPU and causes the CPU to enter the interrupt service routine. On the [Figure 16-2](#), the error is converted to an outband error pin assertion. The error pin assertion signals the BMC of the error and causes BMC to service the error. In either case, the service agent (CPU or BMC) would inquire the IOH of the information associated with the error, and takes the appropriate action to recover the system from the error condition.

Figure 16-1. Error Signal Converted to Interrupt Example

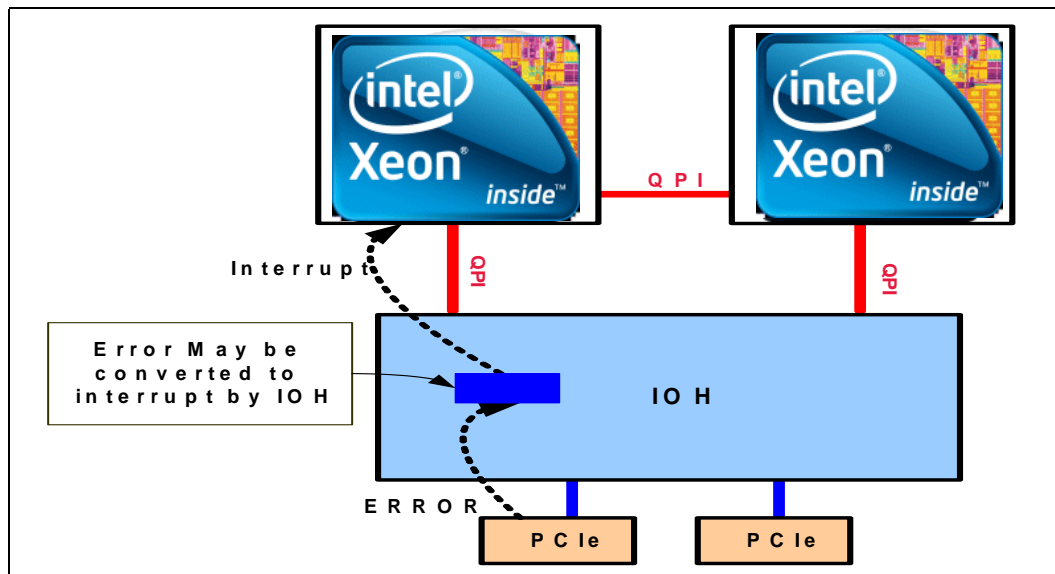
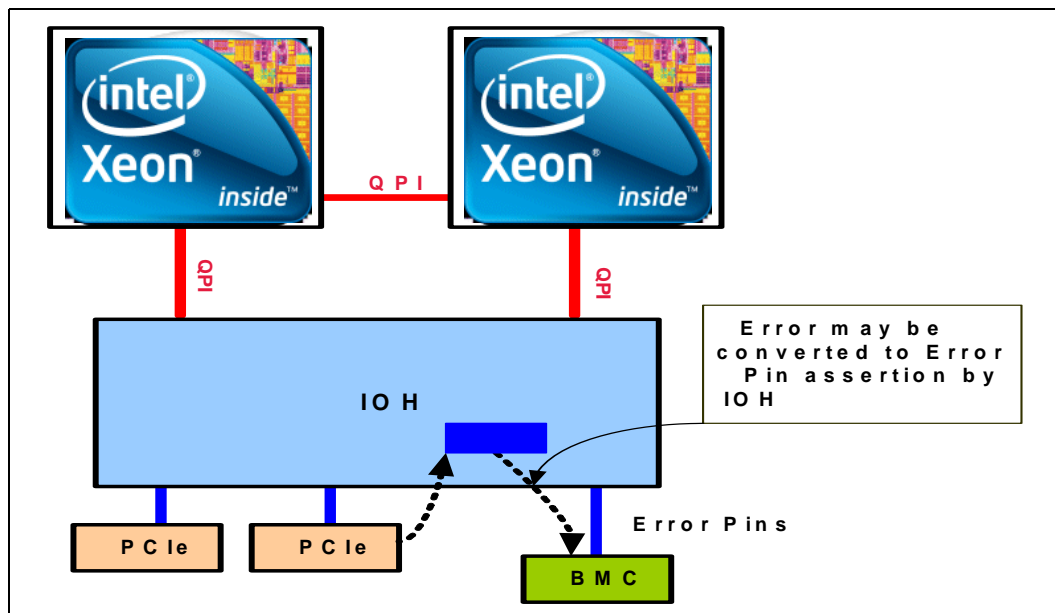


Figure 16-2. Error Signal Converted to Error Pins Example



The IOH detects errors from the PCI Express link, ESI link, Intel QuickPath Interconnect link, or IOH core itself. The error is first logged and mapped to an error severity, and then mapped to a system event(s) for error reporting.

IOH error report features are summarized below and detailed in the following sections:

- Detect and logs Intel QuickPath Interconnect, PCI Express/ESI, and IOH core errors.
- First and Next error detection and logging for fatal and non-fatal errors.
- Allows flexible mapping of the detected errors to different error severities.



- Allows flexible mapping of the error severity to different reporting mechanisms
- Supports PCI Express error reporting mechanism.

16.4.1 Error Severity Classification

16.4.1.1 General Error Severity Classification

In the IOH, general errors are classified into three severities: Correctable, Recoverable, and Fatal. This classification separates those errors resulting in functional failures from those errors resulting in degraded performance. Each severity can trigger a system event according to the mapping defined by the error severity register. This mechanism provides software the flexibility to map an error to the suitable error severity. For example, a platform may choose to respond to uncorrectable ECC errors with low priority, while another platform design may require mapping the same error to a higher severity. The mapping of the error is set to the default mapping at power-on, such that it is consistent with default mapping defined in [Table 16-2](#). The software/firmware can choose to alter the default mapping after power on.

16.4.1.1.1 Correctable Errors (Severity 0 Error)

Hardware correctable errors include those error conditions where the system can recover without any loss of information. Hardware corrects these errors and no software intervention is required. For example, a Link CRC error, which is corrected by Data Link Level Retry, is considered a correctable error.

- Errors corrected by the hardware without software intervention. System operation may be degraded but its functionality is not compromised.
- Correctable errors may be logged and reported in an implementation-specific manner:
 - Upon the immediate detection of the correctable error, or
 - Upon the accumulation of errors reaching a threshold.

16.4.1.1.2 Recoverable Errors (Severity 1 Error)

Recoverable errors are software correctable or software/hardware uncorrectable errors which cause a particular transaction to be unreliable although the system hardware is otherwise fully functional. Isolating recoverable errors from fatal errors provides system management software the opportunity to recover from the error without reset and disturbing other transactions in progress. Devices not associated with the transaction in error are not impacted by the error. An example of a recoverable error is an ECC Uncorrectable error that affects only the data portion of a transaction.

- Error could not be corrected by hardware and may require software intervention for correction, or
- Error could not be corrected. Data integrity is compromised, but system operation is not compromised.
- Requires immediate logging and reporting of the error to the processor.
- OS/Firmware takes the action to contain the error and begin recovery process on affected partition.



Software Correctable Errors

Software correctable errors are considered “recoverable” errors. This includes those error conditions where the system can recover without any loss of information. Software intervention is required to correct these errors.

- Requires immediate logging and reporting of the error to the processor.
- Firmware or other system software layers take corrective actions.
- Data integrity is not compromised with such errors.

16.4.1.1.3 Fatal Errors (Severity 2 Error)

Fatal errors are uncorrectable error conditions which render a related hardware unreliability. For fatal errors, inband reporting to the processor is still possible. A reset of the entire hard partition may be required to return to reliable operation.

- System integrity is compromised and continued operation may not be possible.
- System interface within a hard partition may be compromised.
- Inband reporting is still possible.
- For example, uncorrectable tag error in cache, or permanent PCI Express link failure, or Intel QuickPath Interconnect failure.
- Requires immediate logging and reporting of the error to the processor.

16.4.1.2 Thermal Error Severity Classification

Thermal errors can be classified into one of three severities supported by IOH. Software also can use “Thermal Error Severity Register (THREERSV)” to program thermal error severity to one of the three severities supported by IOH or generate either THERMALERT_N or THERMTRIP_N signal please refer to [Figure 16-7](#).

IOH can set up thermal thresholds to generate thermal alert and trip signals when the IOH temperature monitoring sensor detects throttle or catastrophic temperature threshold reached. Intel recommendation is to keep default values for Thermalert and Thermtrip severities.

Note: Intel recommendation is to keep thermal error severities as default which described in [Section 21.6.7.6, “THREERSV: Thermal Error Severity Register”](#).

16.4.2 Inband Error Reporting

Inband error reporting signals the system of a detected error via inband cycles. There are two complementary inband mechanisms in the IOH. The first mechanism is synchronous reporting, along with transaction responses/completions; the second mechanism is asynchronous reporting of an inband error message or interrupt. These mechanisms are summarized as follows:

Synchronous Reporting

- Data Poison bit indication in the header:
 - Generally for uncorrectable data errors (for example, uncorrectable data ECC error).
- Response status field in response header:
 - Generally for uncorrectable error related to a transaction (for example, failed response due to an error condition).



- No Response
 - Generally for uncorrectable error that has corrupted the requester information and returning a response to the requester becomes unreliable. The IOH silently drops the transaction. The requester will eventually time out and report an error.

Asynchronous Reporting

- Reported through inband error or interrupt messages:
 - A detected error triggers an inband message to the IOH or processor.
 - Errors are mapped to three error severities. Each severity can generate one of the following inband messages (programmable):
 - MCA
 - CPEI *
 - NMI
 - SMI
 - MCA **
 - None (inband message disable)
 - Each error severity can also cause an error pin assertion in addition to the above inband message.
 - Fatal severity can cause viral in addition to the above inband message and error pin assertion.
 - The IOH PCI Express root ports can generate MSI, or forward MSI/INTx messages from downstream devices, per the *PCI Express Base Specification*, Revision 2.0.

16.4.2.1 Synchronous Error Reporting

Synchronous error reporting is generally received by a component, where the receiver attempts to take corrective action without notifying the system. If the attempt fails, or if corrective action is not possible, synchronous error reporting may eventually trigger a system event via the asynchronous reporting mechanisms. Synchronous reporting methods are described in the following sections.

16.4.2.1.1 Completion/Response Status

A Non-Posted Request requires the return of the completion cycle. This provides an opportunity for the responder to communicate to the requester the success or failure of the request. A status field can be attached to the completion cycle and sent back to the requester. A successful status signifies that the request was completed without an error. Conversely, a “failed” status denotes that an error has occurred as the result of processing the request.

16.4.2.1.2 No Response

For errors that have corrupted the requester’s information (for example, requester/source ID in the header), the IOH will not send a response back to the requester. This will eventually cause the original requester to time-out and trigger an error at the requester.

16.4.2.1.3 Data Poisoning

A Posted Request that does not require a completion cycle needs another form of synchronous error reporting. When a receiver detects an uncorrectable data error, it must forward the data to the target with the “bad data” status indication. This form of



error reporting is known as “data poisoning”. The target that receives poisoned data must ignore the data or store it with “poisoned” indication. Both PCI Express and Intel® QuickPath Interconnect provide a poison bit field in the transaction packet that indicates the data is poisoned. Data poisoning is not limited to posted requests. Requests that require completion with data, which can also indicate poisoned data.

Since IOH can be programmed to signal (interrupt or error pin) the detection of poisoned data, software should ensure that the report of the poisoned data should come from one agent, preferably by the original agent that detects the error, that is, the agent that poisoned the data.

In general, the IOH forwards the poisoned indication from one interface to another (for example, Intel QuickPath Interconnect to PCI Express, PCI Express to Intel QuickPath Interconnect, or PCI Express to PCI Express).

16.4.2.1.4 Time-Out

Time-out error indicates that a transaction failed to complete due to expiration of the Time-out counter. This could be a result of corrupted link packets, I/O interface errors, and so on. In the IOH, transaction time-out is tracked from each PCIe root port or internal source. Intel QuickPath Interconnect’s time-out mechanism is not supported in tracking of time-out at the source CPU or I/O interface.

16.4.2.2 IOH Asynchronous Error Reporting

Asynchronous error reporting is used to signal the system of a detected error. For an error that requires immediate attention, an error that is not associated with a transaction, or an error event that requires system handling, asynchronous report is used. Asynchronous error reporting is controlled through the IOH error registers. These registers enable the IOH to report various errors via system events (for example, NMI, etc). In addition, the IOH provides standard sets of error registers specified in the *PCI Express Base Specification*, Revision 2.0.

The IOH error registers provide software the flexibility to map an error to one of the three error severities. Software can associate each of the error severities with one of the supported inband messages or be disabled for inband messaging. The error pin assertion can also be enabled/disabled for each of the error severities. Upon detection of a given error severity, the associated event(s) is triggered, which conveys the error indication through inband and/or outband signaling. Asynchronous error reporting methods are described in the following sections.

16.4.2.2.1 NMI (Non-Maskable Interrupt)

ICH reports NMI through the assertion of the NMI_N pin. When an error triggers NMI, IOH will broadcast a NMI virtual legacy wire cycle to the CPUs via Intel QuickPath Interconnect. IOH converts NMI pin assertion to the Intel QuickPath Interconnect legacy wire cycle on the behalf of the ICH. Refer to [Chapter 8, “Interrupts”](#) for more IOH interrupt handling.

NMI input to IOH can also be routed to MCA message, refer to [Section 21.11.2.25, “QPIPNNMIC: Intel® QuickPath Interconnect Protocol NMI Control”](#) for register descriptions.

16.4.2.2.2 CPEI (Correctable Platform Event Interrupt)

CPEI is associated with an interrupt vector that is programmed in the ICH component. When CPEI is needed for error reporting, the IOH is configured to send CPEI message to the legacy IOH. The message is converted in the Legacy IOH to Error_N[0] pin



assertion that conveys the CPEI event when enabled. As a result, ICH sends a CPU interrupt with the specific interrupt vector and type defined for CPEI. The CPEI message is decoded by the CSI block where the corresponding error logging status bit is set. The CSI block also drives the appropriate error severity signal to the Global Error Escalation block to assert ERR_N[0].

16.4.2.2.3 SMI (System Management Interrupt)

The IOH supports the use of the System Management Interrupt when used in Intel® Xeon® processor based systems. Through the appropriate configuration of the IOH error control and SYSMAP registers, error events within the IOH can be directed to the SMI, allowing the BIOS SMI handler to be the first responder to error events. Refer to the Intel® QuickPath Interconnect Protocol SMI control register [Section 21.11.2.24, “QPIPSMIC: Intel® QuickPath Interconnect Protocol SMI Control”](#) for more detailed description.

16.4.2.2.4 Machine Check Abort (MCA)

The machine check abort (MCA) message is used to indicate severe error conditions in the system that needs immediate attention. This is typically used by the IOH on detection of a severe but contained error to alert one of the processors such that error handling software can take appropriate action to either recover or shutdown the system. System architecture provides a machine check abort mechanism that cannot be masked or disabled by other tasks and provides a more robust mechanism for dealing with errors. The machine check abort message on Intel QuickPath Interconnect enables a processor to utilize this feature. The machine check abort message is delivered on Intel QuickPath Interconnect using the IntPhysical transaction with a machine check delivery mode. This delivery mode is always used with physical destination mode, directed to a single processor context and edge triggered.

When an error triggers an MCA, the IOH will dispatch a SpcInt cycle to the designated processor specified in the MCA CSR. Refer to [Section 21.11.2.26, “QPIPMCAC: Intel QuickPath Interconnect Protocol MCA Control”](#) for register descriptions.

Note: Both Intel Itanium processor 9300 series and Intel Xeon processor 7500 series support MCA.

16.4.2.2.5 None (Inband Message Disable)

The IOH provides the flexibility to disable inband messages on the detection of an error. By disabling the inband messages and enable error pins, IOH can be configured to report the errors exclusively via error pins.

16.4.2.2.6 Error Pins[2:0]

The IOH provides three open-drain error pins for the purpose of error reporting – one pin for each error severity. The error pin can be used in certain class of platforms to indicate various error conditions and can also be used when no other reporting mechanism is appropriate. For example, error signals can be used to indicate error conditions (even hardware correctable error conditions) that may require error pin assertion to notify outband components (such as BMC) in the system. In some extreme error conditions, when inband error reporting is no longer possible, the error pins provide a way to inform the outband agent of the error. Upon detecting error pin assertion, the outband agent interrogates various components in the system and determines the health state of the system. If the system can be gracefully recovered without reset, the BMC performs the proper steps to put the system back to a functional state. However, if the system is unresponsive, the outband agent can assert reset to force the system back to a functional state.



The IOH allows software to enable/disable error pin assertion upon the detection of the associated error severity (in addition to inband message). When a detected error severity triggers an error pin assertion, the corresponding error pin is asserted. Software must clear the error pin assertion, including after a reset, via the error pin status register. The error pins can also be configured as general purpose outputs. In this configuration, software can write directly to the error pin register to cause the assertion and deassertion of the error pin.

16.4.2.2.7 Thermalert_N and Thremtrip_N Pins

There are two open-drain pins reserved for thermal errors. When the IOH is programmed to map thermal alert and thermal trip errors to THERMALERT_N and THERMTRIP_N pins, then these errors can not generate any type of event in the system events.

Notice: The THERMTRIP_N and THERALERT_N pins should not be used for other errors except for thermal errors.

16.4.2.2.8 Viral Alert

Viral alert is a mechanism to indicate fatal error where it is difficult to avoid error propagation without immediately shutting down the system. Viral alert addresses the error propagation issue related to fatal errors and allows the system to be shutdown gracefully and in the process cleaning up the system interface. This reporting mechanism assumes that the Intel QuickPath Interconnect interface is operational and can be used to deliver the error indication. Each Intel QuickPath Interconnect packet header contains a viral alert bit in a profile-dependent part to indicate if a fatal error has compromised the system state. Each protocol layer agent that detects a fatal error or receives a packet that has its viral alert indication set, turns viral and starts setting the viral alert indication on all packets initiating from itself until the agent is reset. Once an agent becomes viral, then it is assumed that its protocol state has been compromised. I/O agents should stop committing any data to permanent storage or I/O devices after it has become viral. The viral alert mechanism is transparent to the Routing and Link layers.

IOH allows the software to enable/disable viral alert upon the detection of Fatal errors (in addition to inband message and the error pin assertion). When viral is enabled for fatal errors, viral alert is triggered upon the detection of the error. When this occurs:

- IOH will initiate subsequent inbound packets (requests and completions) with viral alert indication set.
- Outbound non-posted transactions (for example, read, configuration, and I/O requests) will be completed immediately by the IOH with the viral bit set and a Failed Response status on the Intel QuickPath Interconnect.
- The IOH will drop all data bound to PCIe (outbound writes and inbound read completions). For outbound writes, the IOH returns an Intel QuickPath Interconnect Failed Response to the initiating CPUs or IOHs.
- Transactions from Intel QuickPath Interconnect targeting the IOH's internal registers (configuration space, memory mapped registers, and so on) will be completed as normal assuming the IOH is healthy enough to respond. The viral bit will be set on the Intel QuickPath Interconnect completion.
- Inbound read and write completions will be converted to a completer abort response on PCIe.
- Inbound posted transactions to the IOH in viral mode will be all dropped. This includes all the messages that the IOH receives from the ICH. The only way to come out of this viral mode is to clear the viral status.



Viral is used in severe error conditions to prevent harmful impact for the error (for example, drop all viral affected transactions until system reset). Viral is used for Fatal errors.

The IOH provides a mechanism to disable and clear viral using configuration registers. This would be used by the platform to turn off viral to allow logging and blue screen alert. The register that clears viral mode is located in [Section 21.6.7.16](#).

16.4.2.2.9 PCI Express INTx and MSI Interrupt Messages

PCI Express INTx and MSI interrupt messages are supported through the PCI Express standard error reporting. The IOH forwards the MSI and INTx interrupt message generated downstream from I/O devices to the PCI Express ports. The IOH PCI Express ports themselves also generate MSI interrupts for error reporting, if enabled. Refer to [Chapter 8, "Interrupts"](#) for details on INTx and MSI interrupts. Also refer to the *PCI Express Base Specification*, Revision 2.0 for details on the PCI Express standard and advanced error capabilities.

16.4.2.2.10 PCIe/ESI "Stop and Scream"

There is an enable bit per PCIe port that controls "stop and scream" mode. In this mode the desire is to disallow sending of poisoned data onto PCIe and instead disable the PCIe port that was the target of poisoned data. This is done because in the past there have been PCIe/ESI devices that have ignored the poison bit, and committed the data which can corrupt the I/O device.

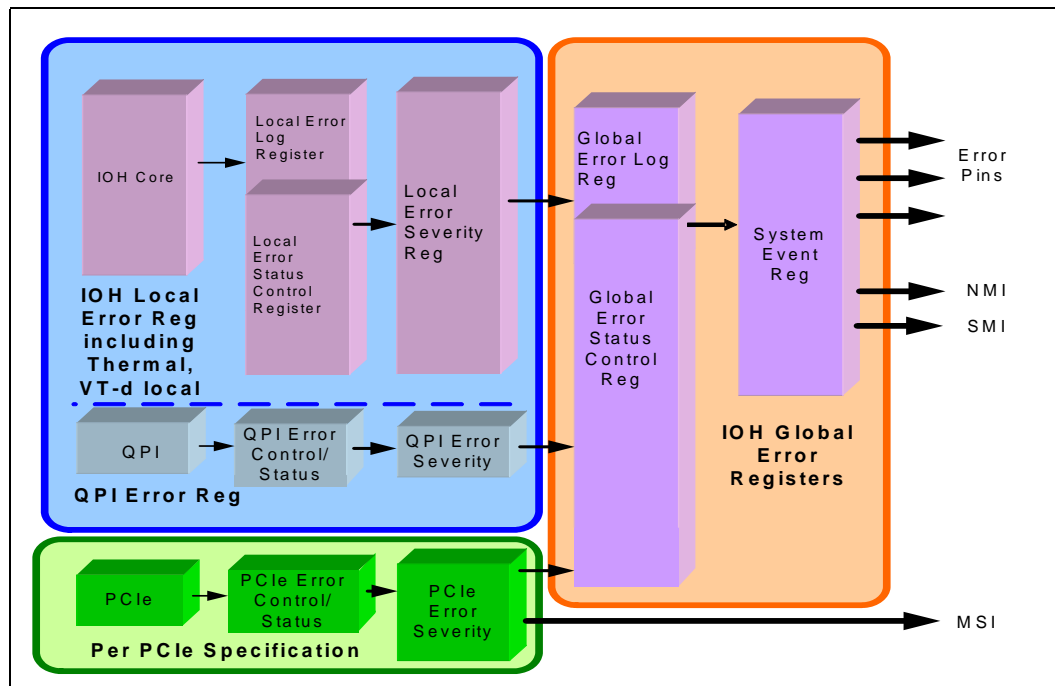
16.4.2.2.11 PCIe "Live Error Recovery"

PCI Express ports support the Live Error Recovery (LER) mode. When errors are detected by the PCIe port, the PCIe port goes into a Live Error Recovery mode. When a root port enters the LER mode, it brings the associated link down and automatically trains the link up.

16.4.3 IOH Error Registers Overview

The IOH contains an extensive set of error registers to support error reporting. These error registers are assumed to be sticky unless specified otherwise, please refer to register attributes for detailed. Sticky means the values of the registers are retained even after a hard reset – they can only be cleared by software or by power-on reset. There are two levels of hierarchy for the error registers – Local and Global. The local error registers are associated with the IOH local clusters (PCI Express, ESI, Intel QuickPath Interconnect, and IOH core). The global error registers collect the errors reported by the local error registers and map them to system events. [Figure 16-3](#) illustrates the high level view of the IOH error registers.

Figure 16-3. IOH Error Registers



16.4.3.1 Local Error Registers

Each IOH local interface contains a set of local error registers. The PCI Express port (including ESI) local error registers are predefined by the *PCI Express Base Specification*, Revision 1.0a. Intel QuickData Technology DMA has a predefined set of error registers inherited from the previous design.

Since Intel QuickPath Interconnect has not defined a set of standard error registers, the IOH has defined the error registers for the Intel QuickPath Interconnect port using the same error control and report mechanism as the IOH core. This is described below. Refer to the [Section 21.8, "IOH Local Error Registers"](#) for the format of these registers. The [Figure 16-4](#) shows the logic diagram of the IOH local error registers.

- **IOH Local Error Status Register (IOHERRST, QPI[1:0]ERRST, QPIP[1:0]ERRST, MIERRST, THRERRST)**

The IOH core provides local error status register for the errors associated with the IOH component. When a specific error occurs in the IOH core, its corresponding bit in the error status register is set. Each error can be individually enabled/disabled by the error control register.

- **IOH Local Error Control Register (IOHERRCTL, QPI[1:0]ERRCTL, QPIP[1:0]ERRST, MIERRST, THRERRCTL)**

The IOH core provides the local error control register for the errors associated with the IOH component. Each error detected by the local error status register can be individually enabled/disabled by the error control register. If an error propagation is disabled, the corresponding status bit will not be set for any subsequent detected error. The error control registers are sticky and they can be reset by COREPWRGOOD reset.

- **Local Error Severity Register (QPI[1:0]ERRSV, QPIP[1:0]ERRSV, IOHERRSV, MIERRSV, THRERRSV, PCIERRSV)**



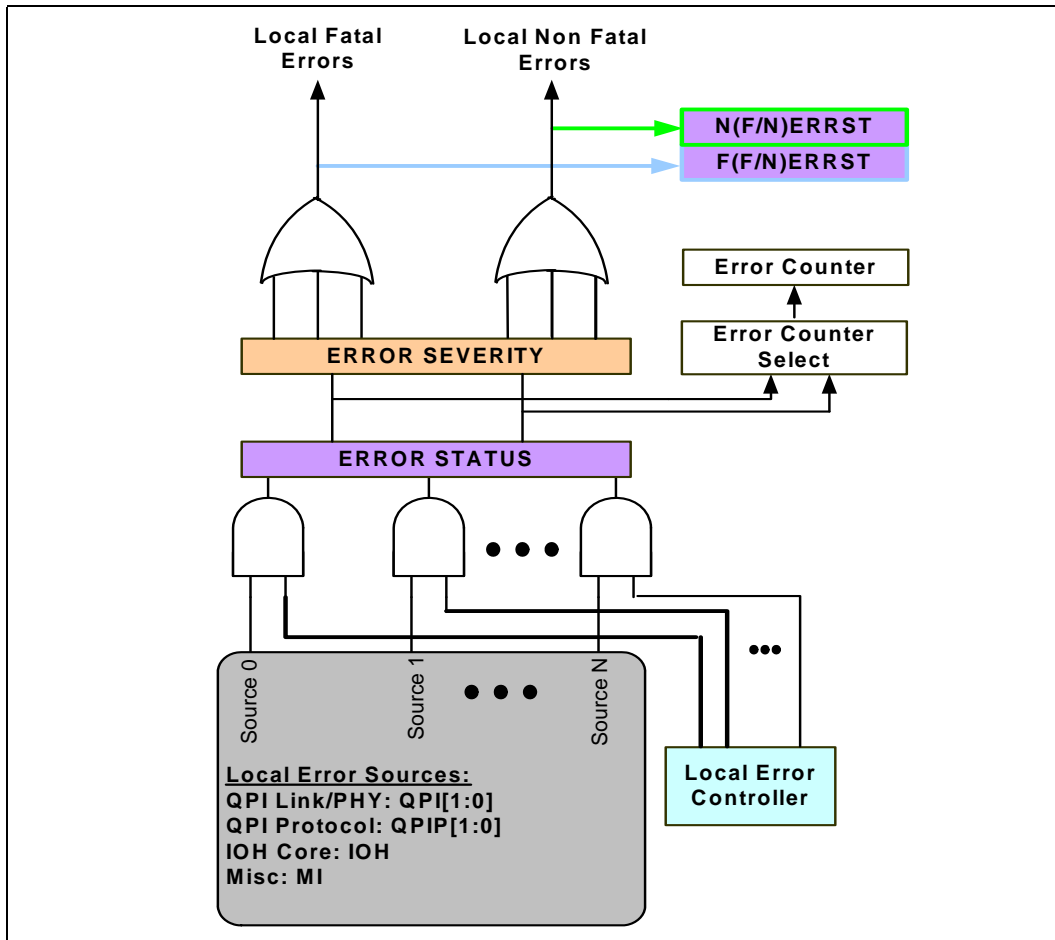
The IOH core provides local error severity registers for the errors associated with the IOH core. IOH internal errors can be mapped to three error severity levels. Intel QuickPath Interconnect and PCI Express error severities are mapped [Table 16-3](#).

- **Local Error Log Register (IOH**ERRST, IOH**ERRHD, QPI[1:0]**ERRST, QPIP[1:0]**ERRRST, QPIP[1:0]**ERRHD, IOHERRCNT, QPI[1:0]ERRCNT, QPIP[1:0]ERRCNT, MI**ERRST, MI**ERRHD, MIERRCNT, THR**ERRST, THRERRCNT)**

The IOH core provides local error log registers for the errors associated with the IOH component. When an error is detected by the IOH, the information related to the error is stored in the log register. IOH core errors are first separated into Fatal and Non-Fatal (Correctable, Recoverable, and Thermal Alert) categories. Each category contains two sets of log registers: First Error (FERR) and Next Error (NERR). The FERR register logs the first occurrence of an error, while the NERR register logs the next occurrences of the errors. NERR does not log header/address or ECC syndrome. Note that FERR/NERR does not log a masked error. The FERR log remains valid and unchanged from the first error detection until the clearing of the corresponding FERR error bit in the error status register by software. The **ERRST registers are only cleared by writing to the corresponding local error status registers.

** : FF (Fatal First Error), FN (Fatal Next Error), NF (Non-Fatal First Error), NN (Non-Fatal Next Error)

Figure 16-4. Local Error Signaling on IOH Internal Errors





16.4.3.2 Global Error Registers

Global error registers collect the errors reported by the local interfaces and convert the error to system events. Refer to the register descriptions in [Section 21.7](#), “Global Error Registers” for bit definitions for each register.

- **Global Error Control/Status Register (GFERRST, GNERRST, GERRCTL)**

The IOH provides two global error status registers to collect the errors reported by the IOH clusters – Global Fatal Error Status (GFERRST) and Global Non-fatal Error Status (GNERRST). Each register has identical format; each bit in the register represents the fatal or non-fatal error reported by its associated interface, e.g., the Intel QuickPath Interconnect port, PCI Express port, or IOH core. Local clusters map the detected errors to three error severities and report them to the global error logic. These errors are sorted into Fatal and Non-fatal, and reported to the respective global error status register, with severity 2 as fatal, severities 0 and 1 reported as non-fatal. When an error is reported by the local cluster, the corresponding bit in the global fatal or non-fatal error status register is set. Each error can be individually masked by the global error control registers. If an error is masked, the corresponding status bit will not be set for any subsequent reported error. The global error control register is non-sticky and cleared by reset.

- **Global Log Registers (GFFERRST, GFNERRST, GNFERRST, GNNERRST, GTIME, G**ERRTIME)**

The GFFERRST logs the first global fatal error while GFNERRST logs the next global fatal errors. Similar for GNFERRST and GNNERRST, the first global non-fatal error is logged in the GNFERRST register while the next global non-fatal errors are logged in the GNNERRST register. The GFFERRST, GFNERRST, GNFERRST and GNNERRST registers have same bit format as GFERRST and GNERRST.

The time stamp log for the first error and next error log registers provides the time when the error was logged. Software can read this register to determine which of the local interfaces have reported the error. The FERR log remains valid and unchanged from the first error detection until the clearing of the corresponding error bit in the FERR by software.

- **Global System Event Registers (GSYSST, GSYSCTL, SYSMAP)**

Errors collected by the global error registers are mapped to system event generations. The system event status bit reflects OR'ed output of all unmasked errors of the associated error severity*. Each system event status bit can be individually masked by the system event control registers. Masking a system event status bit forces the corresponding bit to 0. When a system event status bit is set (transition from 0 to 1), it can trigger one or more system events based on the programming of the system event map register as shown in [Figure 16-6](#). Each severity type can be associated with one of the system events: SMI, NMI, or MCA. In addition, the error pin registers allow error pin assertion for an error. When an error is reported to the IOH, the IOH uses the severity level associated with the error to identify which system event should be sent to the system. For example, error severity 2 may be mapped to NMI with error[2] pin enabled. If an error with severity level 2 is reported and logged by the Global Log Register, then an NMI is dispatched to the processor and IOH error[2] is asserted. The processor or BMC can read the Global and Local Error Log register to determine where the error came from, and how it should handle the error.

At power-on reset, these registers are initialized to their default values. The default mapping of severity and system event is set to be consistent with [Table 16-2](#).

Firmware can choose to use the default values or modify the mapping according to the system requirements.

The system event control register is a non-sticky register that is cleared by hard reset.



The Figure 16-5 shows an example how an error is logged and reported to the system by the IOH.

The Figure 16-6 shows the logic diagram of the IOH global error registers.

Figure 16-5. IOH Error Logging and Reporting Example

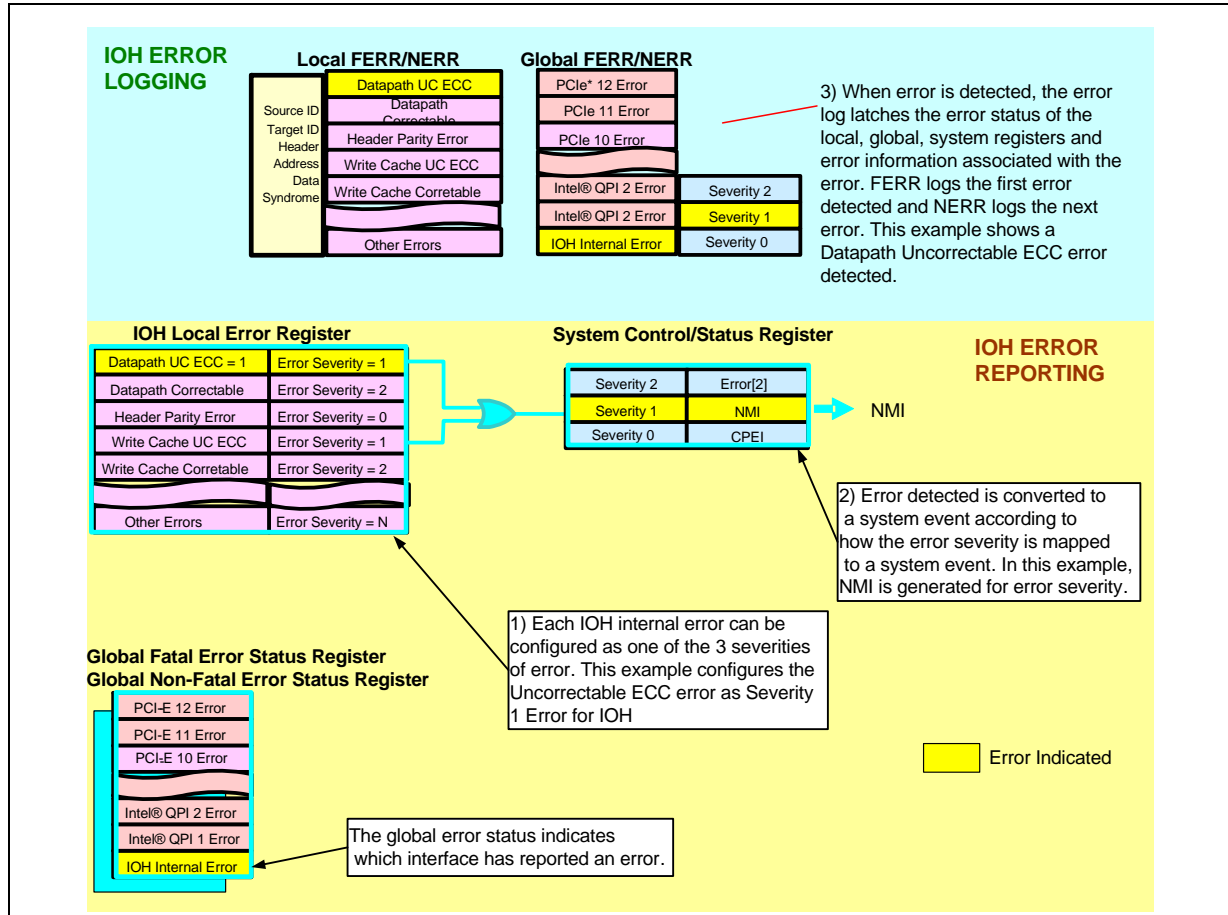




Figure 16-6. Global Error Logging and Reporting

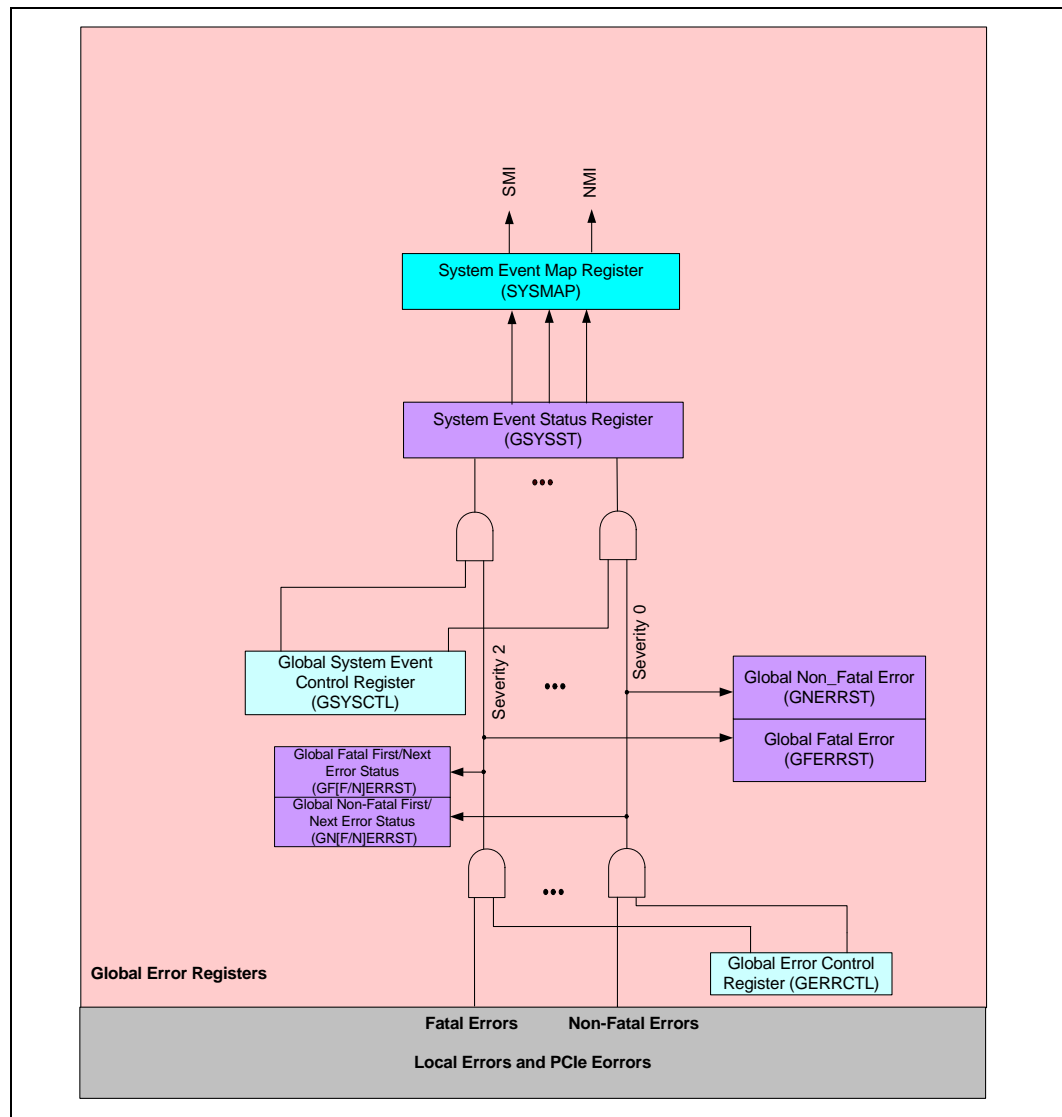
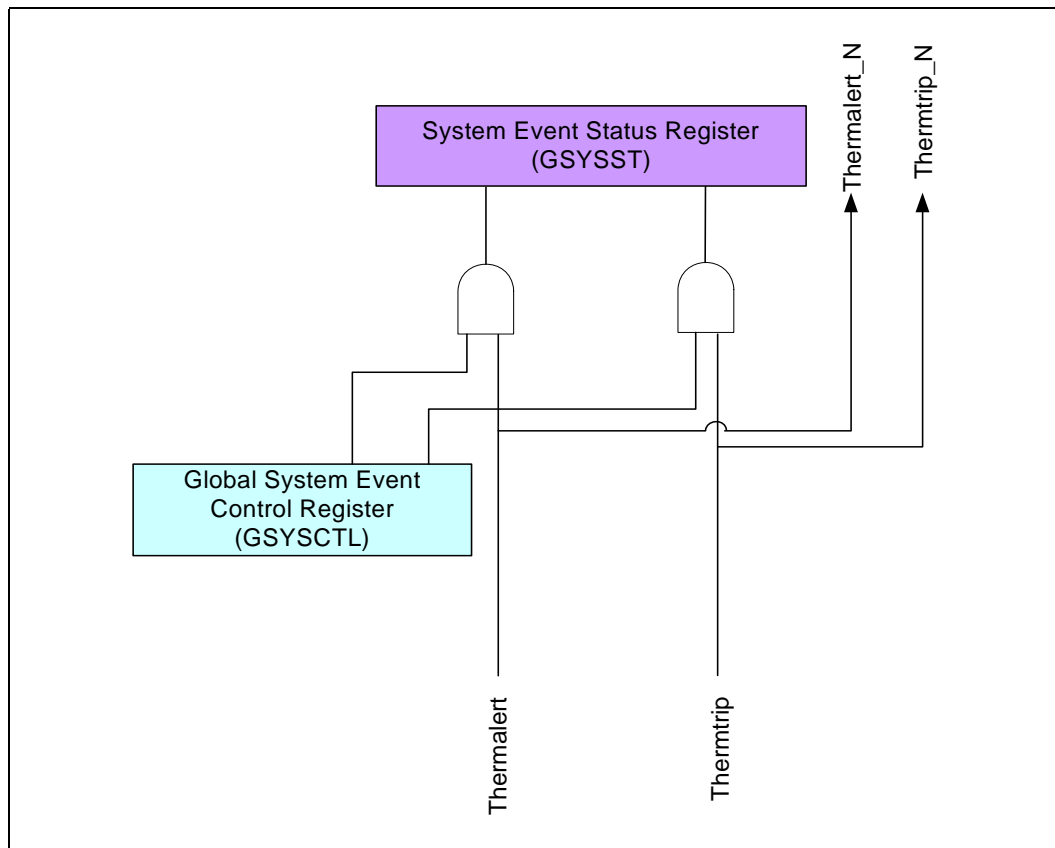


Figure 16-7. Thermalert and Thermtrip Signaling


16.4.3.3 First and Next Error Log Registers

This section describes local error logging (for Intel QuickPath Interconnect, IOH core errors), and the global error logging. PCI Express specifies its own error logging mechanism which will not be described here. Refer to the *PCI Express Base Specification*, Revision 2.0 specification for details.

For error logging, IOH categorizes the detected errors into Fatal and Non-Fatal based on the error severity. Each category includes two sets of error logging – first error register (FERR) and next error register (NERR). FERR register stores the information associated with the first detected error, while NERR stores the information associated with the detected next errors after the first error. Both FERR and NERR logs the error status in the same format. They indicate errors that can be detected by the IOH in the format bit vector with one bit assigned to each error. First error event is indicated by setting the corresponding bit in the FERR status register, a next error(s) is indicated by setting the corresponding bit in the NERR register. In addition, the local FERR register also logs the ECC syndrome, address and header of the erroneous cycle. The FERR register indicates only one error, while the NERR register can indicate second error.

Once the first error and the next error have been indicated and logged, the log registers for that error remains valid until either 1) The first error bit is clear in the associated error status register, or 2) a powergood reset occurs. Software clears an error bit by writing 1 to the corresponding bit position in the error status register.



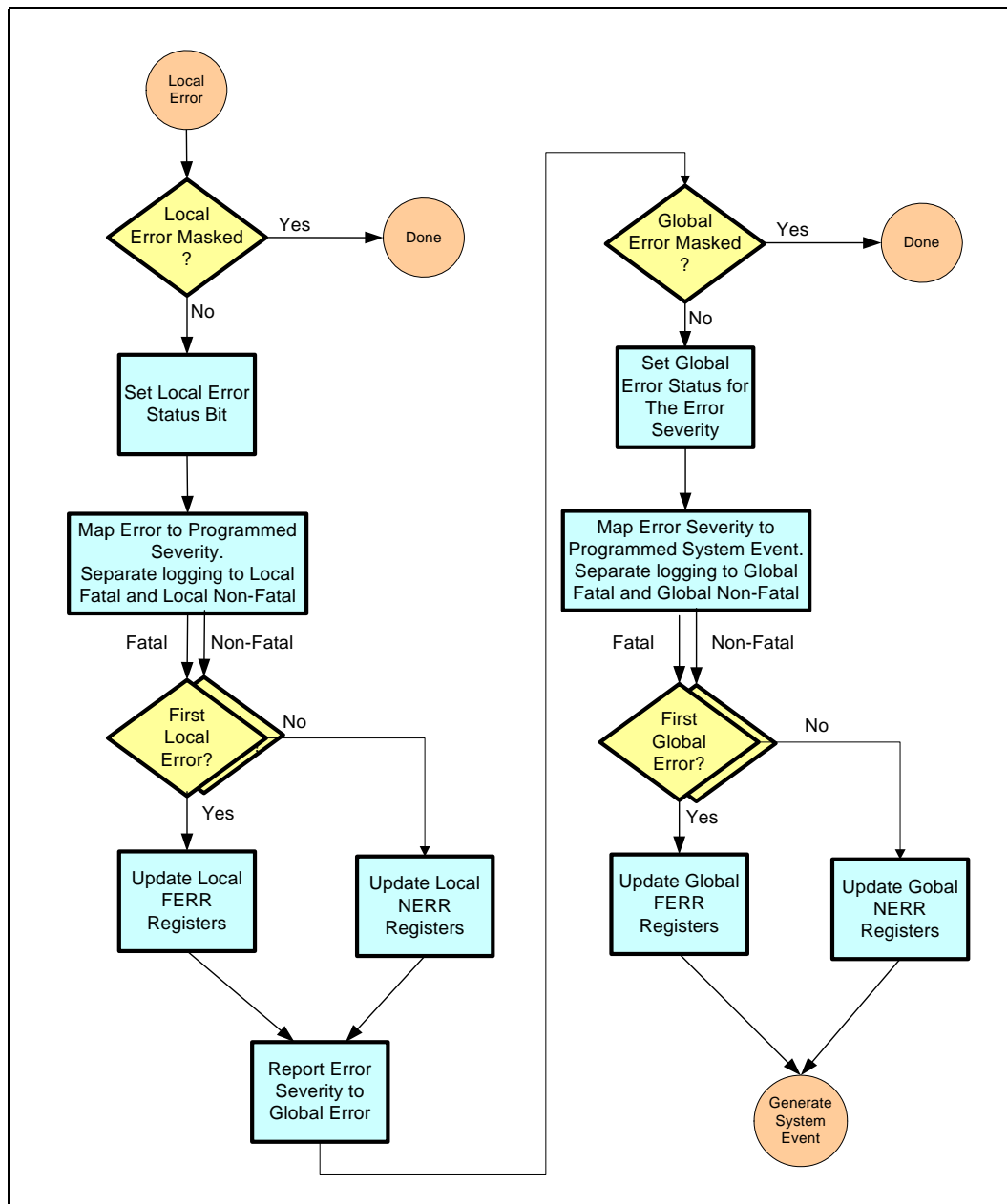
The hardware rules for updating the FERR and NERR registers and error logs are as follows:

1. First error event is indicated by setting the corresponding bit in the FERR status register, a next error is indicated by setting the corresponding bit in the NERR status register.
2. If the same error occurs before the FERR status register bit is cleared, it is not logged in the NERR status register.
Note: There is exception for the Intel QuickPath Interconnect link layer and protocol layer, IOH core error logging. If the first error occurs again, it gets logged again into NERR status register.
3. If multiple error events, sharing the same error log registers, occur simultaneously, then highest error severity has priority over the others for FERR logging. The other errors are indicated in the NERR register.
4. Fatal error is of the highest priority, followed by Recoverable errors and then Correctable errors.
5. Updates to error status and error log registers appear atomic to the software.
6. Once the first error information is logged in the FERR log register, the logging of FERR log registers is disabled until the corresponding FERR error status is cleared by the software.
7. Error control registers are cleared by reset. Error status and log registers are cleared by the power-on reset only. The contents of error log registers are preserved across a reset (while PWRGOOD remains asserted)

16.4.4 Error Logging Summary

Figure 16-8 summarizes the error logging flow for the IOH. As illustrated in the flow chart, the left half depicts the local error logging flow, while the right half depicts the global error logging flow. The local and the global error logging are very similar to each other. Note that for simultaneous events, the IOH serializes the events with higher priority on more severe error.

Figure 16-8. IOH Error Logging Flow



16.4.4.1 Error Registers Flow

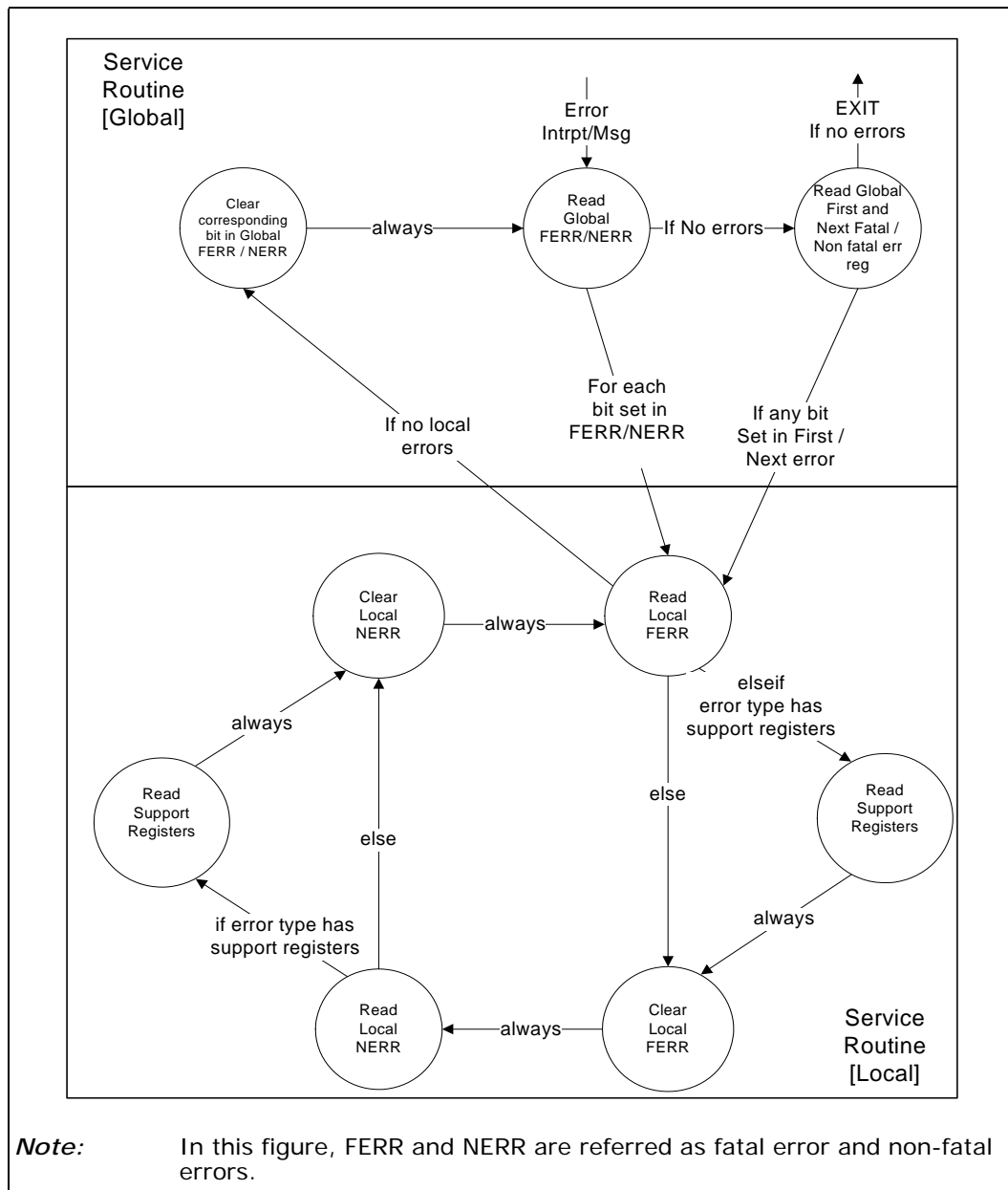
1. Upon a detection of an unmasked local error, the corresponding local error status is set if the error is enabled; otherwise the error bit is not set and the error forgotten.
2. The local error is mapped to its associated error severity defined by the error severity map register. Setting of the local error status bit causes the logging of the error – Severity 0, 1 and 3 are logged in the local non-fatal FERR/NERR registers,



while severity 2 is logged in the local fatal FERR/NERR registers. PCIe errors are logged according to the *PCI Express Base Specification*.

3. The local FERR and NERR logging events are forwarded to the global FERR and NERR registers. The report of local FERR/NERR sets the corresponding global error bit if the global error is enabled; otherwise the global error bit is not set and the error forgotten. The global FERR logs the first occurrence of local FERR/NERR event in the IOH, while the global NERR logs the subsequent local FERR/NERR events.
4. Severity 0 and 1 are logged in the global non-fatal FERR/NERR registers, while severity 2 is logged in the global fatal FERR/NERR registers.
5. The global error register reports the error with its associated error severity to the system event status register. The system event status is set if the system event reporting is enabled for the error severity; otherwise the bit is not set and the error is not reported.
6. Setting of the system event bit triggers a system event generation according to the mapping defined in the system event map register. The associated system event is generated for the error severity and dispatched to the processor/BMC of the error (interrupt for processor or Error Pin for the BMC).
7. The global log and local log registers provide the information to identify the source of the error. Software can read the log registers and clear the global and local error status bits.
8. Since the error status bits are edge triggered, 0 to 1 transition is required to set the bit again. While the error status bit (local, global, or system event) is set to 1, all incoming error reporting to the respective error status register will be ignored (no 0 to 1 transition)
 - a. For a write to clear the local error status bit, the local error register re-evaluates the OR output of its error bits and reports it to the global error register; however, if the global error bit is already set, then the report is ignored,
 - b. For write to clear the global error status bit, the global error register re-evaluates the OR output of its error bits and reports it to the system event status register; however, if the system event status bit is already set, then the report is not generated
 - c. Software can optionally mask or unmask the system event generation (interrupt or error pin) for an error severity in the system event control register while clearing the local and global error registers.
9. Software has the following options for clearing error status registers:
 - a. Read global and local log registers to identify the source of the error. Clear local error bits -- this does not cause generation of an interrupt with the global bit still set. Then, clear global error bit and write to the local error register again with all 0s. Writing 0s to the local status does not clear any status bit, but will cause the re-evaluation of the error status bits. An error will be reported if there is any unclear local error bit.
 - b. Read global and local log registers to identify the source of the error and mask the error reporting for the error severity. Clear system event and global error status bits -- this causes setting of the system event status bit if there are other global bits still set. Then clear local error status bits — this causes setting of the global error status bit if there are other local error bits still set. Then, unmask system event to cause IOH to report the error.
10. FERR logs the information of the first error detected by the associated error status register (local or global). FERR log remains unchanged until all bits in the respective error status register are cleared by the software. When all error bits are cleared, the FERR logging is reenabled.

Figure 16-9. Clearing Global and Local FERR/NERR Registers



16.4.4.2 Error Counters

This feature allows the system management controller to monitor the count of correctable errors. The error RAS structure already provides a first error status and a second error status. Because the response time of system management is on the order of milliseconds, it is not possible to detect short bursts of errors. Over an extended period of time, software uses these error counter values to monitor the rate of change in error occurrences and identify potential degradations, especially with respect to the memory interface.



16.4.4.2.1 Feature Requirements

A register with one-hot encoding will select which error types participate in error counting. The selection register will OR together all of the selected error types to form a single count enable. This means that only one increment of the counter will occur for one or all types selected. Register attributes are set to write 1 to clear.

Each cluster has one set of error counter/control registers.

- Each Intel QuickPath Interconnect port will contain one 7-bit counter (ERRCNT[6:0]).
 - Bit[7] is an overflow bit, all bits are sticky with a write logic 1 to clear.
- The IOH cluster (Core) contains one 7-bit counter (ERRCNT[6:0]).
 - Bit[7] is an overflow bit, all bits are sticky with a write logic 1 to clear.
- The Miscellaneous cluster (MI) contains one 7-bit counter (ERRCNT[6:0]).
 - Bit[7] is an overflow bit, all bits are sticky with a write logic 1 to clear.
- The Thermal Error cluster (THER) contains one 7-bit counter (ERRCNT[6:0]).
 - Bit[7] is an overflow bit, all bits are sticky with a write logic 1 to clear.

Table 16-1. Error Counter Register Locations

| Cluster | Register Reference |
|------------------------------|---|
| Intel QuickPath Interconnect | Intel QuickPath Interconnect Error Counter Selection Register (QPI[1:0]ERRCNTSEL) Intel QuickPath Interconnect Error Counter Register (QPI[1:0]ERRCNT) Intel QuickPath Interconnect Protocol Error Counter Register (QPIP[1:0]ERRCNT) |
| Core IOH | IOH Error Counter Selection Register (IOHERRCNTSEL) |
| Miscellaneous | Miscellaneous Error Counter Selection Register (MIERRCNTSEL) |
| Thermal Error | Thermal Error Counter Selection Register (THRErrCNTSEL) |

16.4.4.3 Stop on Error

The System Event Map register selects the severity levels which activates the Stop on Error (Error Freeze). It requires a reset to clear the event or a configuration write (using JTAG or SMBus) to the stop on error bit in the selection register. Continued operation after Stop on Error is *not* guaranteed. See the System Event Map register (SYSMAP) in the [Chapter 21, “SYSMAP: System Error Event Map Register”](#) for details.

16.5 Intel QuickPath Interconnect Interface RAS

The following sections provide an overview of the Intel QuickPath Interconnect RAS features. The Intel QuickPath Interconnect RAS features are summarized as follows:

1. Link Level 8-bit or 16-bit CRC.
2. Dynamic link retraining and recovery on link failure.
3. Intel QuickPath Interconnect Error detection and logging.
4. Intel QuickPath Interconnect Error reporting.

16.5.1 Link Level CRC and Retry

Cyclic redundancy check (CRC) is a mechanism to ensure the data integrity of a serial stream. The sender of the data generates CRC based on the data pattern and a defined polynomial equation. The resulting CRC is a unique encoding for a specific data stream.



When the data arrives at the receiver, the receiver performs the same CRC calculation using the same polynomial equation. The CRCs are compared to detect bad data. When a CRC error is detected, the receiver will request the sender to retransmit the data. This action is termed “link level retry”, as it is performed by the Link layer logic. The Protocol layer is unaware of this action.

Intel QuickPath Interconnect uses 8 bit CRC per flit (72+8=80 bits), and 16 bit CRC over 2 flits for Intel QuickPath Interconnect packets. The CRC is capable of detecting 1, 2, 3 and odd number of bits in error and errors of burst length up to 8. Flits are logged in a retry buffer until acknowledgment is received. In case of error, the erroneous flit and all subsequent flits are retransmitted. Recovery from permanent partial link failure is supported through dynamic link width reduction (see [Section 16.5.2](#)).

In addition, the IOH tracks and logs link level retry in the error registers. A successful link level retry and successful link reduction is a correctable error, while repetitive retries without success and a link that cannot be further reduced is a fatal error. The flit that contained the error will be logged with 8-bit CRC. With a Rolling CRC (16-bit) failure, both flits contributing to the failed CRC are logged.

16.5.2 Intel QuickPath Interconnect Error Detection, Logging, and Reporting

The IOH implements Intel QuickPath Interconnect error detection and logging that follows the IOH local and global error reporting mechanisms described earlier in this chapter. These registers provide the control and logging of the errors detected on the Intel QuickPath Interconnect interface. IOH Intel QuickPath Interconnect error detection, logging, and reporting provides the following features:

- Error indication by interrupt (CPEI, MCA, SMI, NMI).
- Error indication by response status field in response packets.
- Error indication by data poisoning.
- Error indication by Viral.
- Error indication by Error pin.
- Hierarchical Time-out for fault diagnosis and FRU isolation.

16.6 PCI Express RAS

The *PCI Express Base Specification*, Revision 2.0 defines a standard set of error reporting mechanisms; the IOH supports the standard set, including error poisoning and Advanced Error Reporting. Any exceptions are called out where appropriate. PCI Express ports support the following features:

1. Link Level CRC and retry.
2. Dynamic link width reduction on link failure.
3. PCI Express Error detection and logging.
4. PCI Express Error reporting.

16.6.1 PCI Express* Link CRC and Retry

PCI Express supports link CRC and link level retry for CRC errors. Refer to the *PCI Express Base Specification*, Revision 2.0 for details.



16.6.2 Link Retraining and Recovery

The PCI Express interface provides a mechanism to recover from a failed link, and continue operating at a reduced link widths. The IOH supports PCI Express ports can operate in x16, x8, x4, x2, and x1 link widths. In case of a persistent link failure, the PCI Express link can degrade to a smaller link width in an attempt to recover from the error. A PCI Express x16 link can degrade to x8 link, a x8 link can fall back to a x4 link, a x4 to a x2 link, and then to a x1 link. Refer to the *PCI Express Base Specification*, Revision 2.0 for further details.

16.6.3 PCI Express Error Reporting Mechanism

The IOH supports the standard and advanced PCIe error reporting for its PCIe ports. The IOH PCI Express ports are implemented as root ports. Refer to the *PCI Express Base Specification*, Revision 2.0 for the details of PCIe error reporting. The following sections highlight the important aspects of the PCI Express error reporting mechanisms.

16.6.3.1 PCI Express Error Severity Mapping in IOH

Errors reported to the IOH PCI Express root port can optionally signal to the IOH global error logic according to their severities through the programming of the PCI Express root control register (ROOTCON). When system error reporting is enabled for the specific PCI Express error type, the IOH maps the PCI Express error to the IOH error severity and reports it to the global error status register. PCI Express errors can be classified as two types: Uncorrectable errors and Correctable errors. Uncorrectable errors can further be classified as Fatal or Non-Fatal. This classification is compatible and mapped with the IOH's error classification: Correctable as Correctable, Non-Fatal as Recoverable, and Fatal as Fatal.

16.6.3.2 Unsupported Transactions and Unexpected Completions

If the IOH receives a legal PCI Express defined packet that is not included in PCI Express supported transactions, the IOH treats that packet as an unsupported transaction and follows the PCI Express rules for handling unsupported requests. If the IOH receives a completion with a requester ID set to the root port requester ID and there is no matching request outstanding, it is considered an "Unexpected Completion". The IOH also detects malformed packets from PCI Express and reports them as errors per the [PCI Express Base Specification Revision 1.0a](#) rules.

If the IOH receives a Type 0 Intel Vendor-Defined message that terminates at the root complex and that it does not recognize as a valid Intel-supported message, the message is handled by IOH as an Unsupported Request with appropriate error escalation (as defined in express spec). For Type 1 Vendor-Defined messages which terminate at the root complex, the IOH simply discards the message with no further action.

16.6.3.3 Error Forwarding

PCIe supports Error Forwarding, or Data Poisoning. This feature allows a PCI Express device to forward data errors across an interface without it being interpreted as an error originating on that interface.

The IOH forwards the poison bit from Intel QuickPath Interconnect to PCIe, PCIe to Intel QuickPath Interconnect and between PCIe ports on peer to peer. Poisoning is accomplished by setting the EP bit in the PCIe TLP header.

16.6.3.4 Unconnected Ports

If a transaction targets a PCI Express link that is not connected to any device, or the link is down (DL_Down status), the IOH treats it as a master abort situation. This is required for PCI bus scans to non-existent devices to go through without creating any other side effects. If the transaction is non-posted, IOH synthesizes an Unsupported Request response status (if non-posted) back to any PCIe requester targeting the down link or returns all Fs on reads and a successful completion on writes to any Intel® QuickPath Interconnect requester targeting the down link. Note that software accesses to the root port registers corresponding to a down PCIe interface does not generate an error.

16.6.3.5 PCI Express Error Reporting Specifics

Refer to *PCI Express Base Specification Rev 1.1, post 1.1 Erratas and EC**s* for details of root complex error reporting. Here is a summary of root port 'system event' reporting. Figure 16-10 provides a summary of system event reporting to IOH global error an PCI Express interface error. Refer to Section 21.13 for registers and descriptions. Table 16-10 and Table 16-11 illustrate the error logging and report mechanism.

Figure 16-10. Error Signaling to IOH Global Error Logic on a PCI Express Interface Error

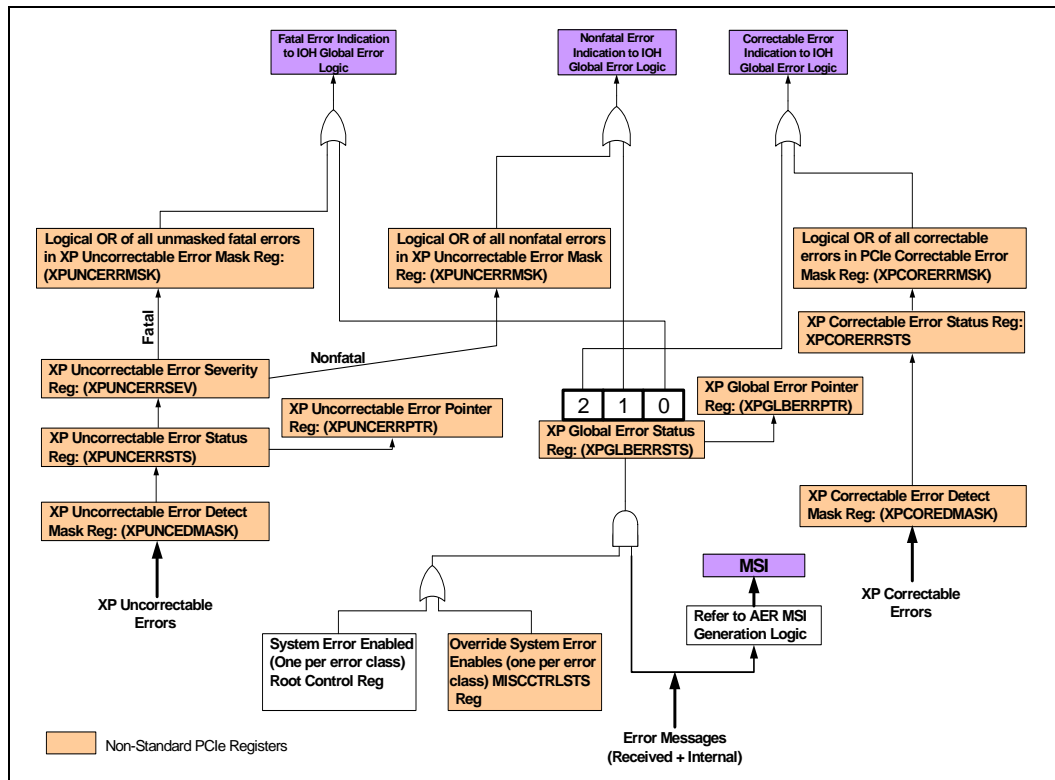
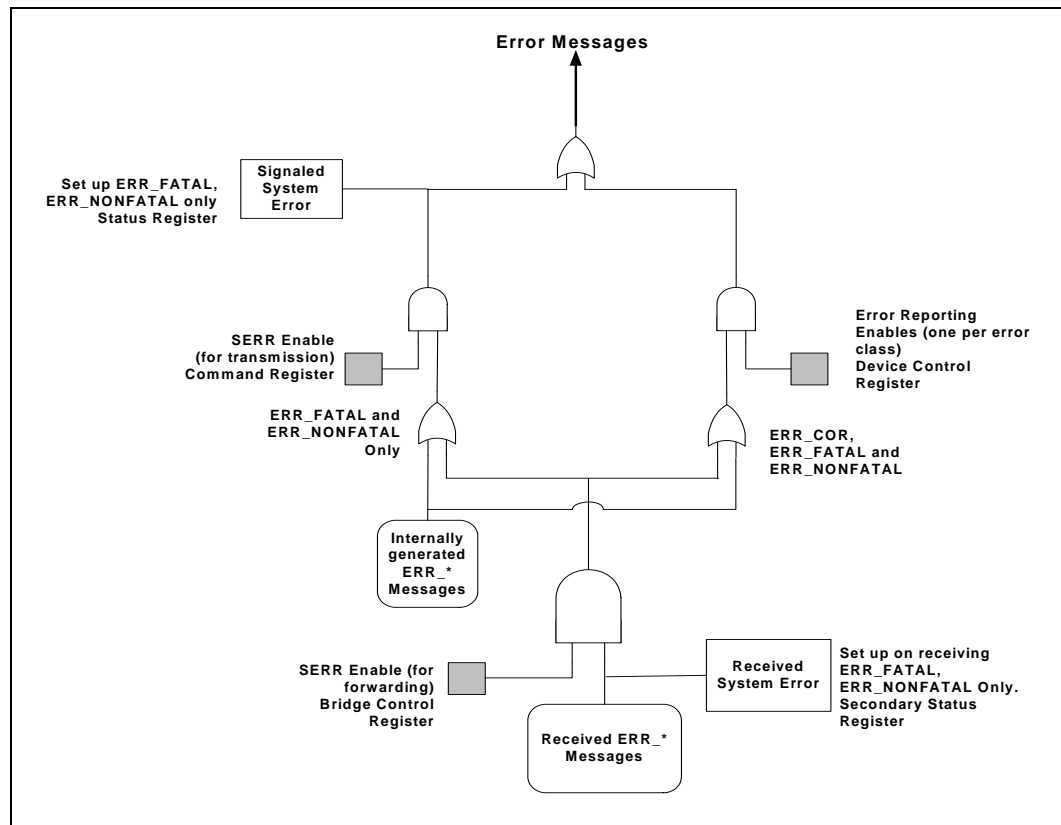




Figure 16-11. PCI Express Error Standard



16.7 IOH Error Handling Summary

The following tables provide a summary of the errors that are monitored by the IOH. The IOH provides a flexible mechanism for error reporting. Software can arbitrarily assign an error to an error severity, and associate the error severity with a system event. Depending on which error severity is assigned by software, the error is logged either in fatal or non-fatal error log registers. Each error severity can be mapped to one of the inband report mechanism as shown in [Table 16-2](#), or generate no inband message at all. In addition, each severity can enable/disable the assertion of its associated error pin for outband error report (for example, severity 0 error triggers Error[0], severity 1 triggers Error[1], ..., and so forth). [Table 16-2](#) shows the default error severity mapping in the IOH and how each error severity is reported, while [Table 16-3](#) summarizes the default logging and responses on the IOH detected errors.

Note: Each error's severity (and therefore, which error registers log the error) is programmable and therefore, the error logging registers used for the error could be different from what is indicated in [Table 16-3](#).



Table 16-2. IOH Default Error Severity Map

| Error Severity | IOH | Intel® QuickPath Interconnect | PCI Express | Inband Error Reporting (programmable) |
|----------------|----------------------------|-------------------------------|-------------------|---------------------------------------|
| 0 | Hardware Correctable Error | Hardware Correctable Error | Correctable Error | NMI/SMI/MCA/CPEI |
| 1 | Recoverable Error | Recoverable Error | Non-Fatal Error | NMI/SMI/MCA/CPEI |
| 2 | Fatal Error | Unrecoverable Error | Fatal Error | NMI/SMI/MCA/CPEI |

Table 16-3. IOH Error Summary (Sheet 1 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|-----------------------------|--|------------------------|--|---|
| IOH Core Errors | | | | |
| C4 | Master Abort Address Error | 1 | IOH Sends completion with MA status and log the error | FERR/NERR is logged in IOH Core and Global Non-Fatal Error Log Registers: |
| C5 | Completer Abort Address Error | | IOH sends completion with CA status and logs the error. | IOHNFERRST IOHNFERRHD IOHNNERRST GNERRST GNFERRST GNFERRTIME GNNERRST IOH core header is logged |
| C6 | FIFO Overflow/ Underflow error | 1 | IOH logs the Error | FERR/NERR is logged in IOH Core and Global Non-Fatal Error Log Registers: IOHNFERRST IOHNFERRHD IOHNNERRST GNERRST GNFERRST GNFERRTIME GNNERRST IOH core header is not logged |
| Miscellaneous Errors | | | | |
| 20 | IOH Configuration Register Parity Error (not including Intel QuickPath Interconnect, PCIe registers which are covered elsewhere) | 0 | No Response – This error is not associated with a cycle. IOH detects and logs the error. | FERR/NERR is logged in Miscellaneous and Global Fatal Error Log Registers: MIFERRST MIFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST No header is logged. |



Table 16-3. IOH Error Summary (Sheet 2 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|--------------------|---|------------------------|---|---|
| 21 | Persistent SMBus retry failure. | 0 | No Response – This error is not associated with a cycle. IOH detects and logs the error. | FERR/NERR is logged in Miscellaneous and Global Fatal Error Log Registers: |
| 22 | Persistent JTAG error. | | | MIFERRST MIFERRHD MIFNERRST |
| 23 | Virtual Pin Port Error. (IOH encountered persistent VPP failure. The VPP is unable to operate.) | | | GFERRST GFFERRST GFFERRTIME GFNERRST No header is logged for this error |
| 24 | Reserved | | | |
| PCIe Errors | | | | |
| 70 | PCIe Receiver Error | 0 | Respond per PCI Express specification | Log error per PCI Express AER requirements for these correctable errors/message. |
| 71 | PCIe Bad TLP | | | Log in XPGLBERRSTS, XPGLBERRPTR registers |
| 72 | PCIe Bad DLLP | | | If PCIe correctable error is forwarded to the global error registers, it is logged in global non-fatal log registers – GNERRST, GNFERRST, GNNERRST, GNFERRTIME |
| 73 | PCIe Replay Time-out | | | |
| 74 | PCIe Replay Number Rollover | | | |
| 75 | Received ERR_COR message from downstream device | | | |
| 76 | PCIe Link Bandwidth changed | | No Response – This error is not associated with a cycle. IOH detects and logs the error. | Log per 'Link bandwidth change notification mechanism' ECN Log in XPCORERRSTS register. If error is forwarded to the global error registers, it is logged in global non-fatal log registers – GNERRST, GNFERRST, GNNERRST, GNFERRTIME |
| 80 | Received 'Unsupported Request' completion status from downstream device | 1 | Intel QuickPath Interconnect to PCIe read: IOH returns all '1s' and normal response to Intel QuickPath Interconnect to indicate master abort Intel QuickPath Interconnect to PCIe NP write: IOH returns normal response PCIe to PCIe read/NP-write: 'Unsupported request' is returned ² to original PCIe requester. SMBus/Jtag accesses: IOH returns 'UR' response status on smbus/jtag | Log in XPUNCERRSTS register If error is forwarded to the global error registers, it is logged in global non-fatal log registers – GNERRST, GNFERRST, GNNERRST, GNFERRTIME |



Table 16-3. IOH Error Summary (Sheet 3 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|----|--|------------------------|--|---|
| 81 | IOH encountered a PCIe 'Unsupported Request' condition, on inbound address decode with the exception of SAD miss (see C6 for SAD miss), and those covered by entry#11. See Table 5-4 | | <p>PCIe read: 'Unsupported request' completion is returned on PCIe</p> <p>PCIe non-posted write: 'Unsupported request' completion is returned on PCIe. The write data is dropped</p> <p>PCIe posted write: IOH drops the write data.</p> | <p>Log error per PCI Express AER requirements for unsupported request. All accesses above address 2[^]51 are logged as UR. In addition Memory reads above 2[^]51 are considered Advisory when UR severity is set to non-fatal. Memory writes in the range 2[^]51 to 2[^]52 are also considered advisory, while Memory writes above 2[^]52 are considered non-fatal, when UR severity is set to non-fatal.</p> <p>Log in XPGLBERRSTS, XPGLBERRPTR registers</p> <p>If PCIe uncorrectable error is forwarded to the global error registers, it is logged in global non-fatal log registers – GNERRST, GNFERRST, GNNERRST, GNFERRTIME</p> |
| 82 | Received 'Completer Abort' completion status from downstream device | | <p>Intel QuickPath Interconnect to PCIe read: IOH returns all '1s' and normal response to Intel QuickPath Interconnect.</p> <p>Intel QuickPath Interconnect to PCIe NP write: IOH returns normal response.</p> <p>PCIe to PCIe read/NP-write: 'Completer Abort' is returned³ to original PCIe requester.</p> <p>SMBus/Jtag accesses: IOH returns 'CA' response status on smbus/jtag</p> | <p>Log in XPUNCERRSTS register</p> <p>If error is forwarded to the global error registers, it is logged in global non-fatal log registers – GNERRST, GNFERRST, GNNERRST, GNFERRTIME</p> |
| 83 | IOH encountered a PCIe 'Completer Abort' condition, on inbound address decode. See Table 5-4 | | <p>PCIe read: 'Completer Abort' completion is returned on PCIe</p> <p>PCIe non-posted write: 'Completer Abort' completion is returned on PCIe. The write data is dropped</p> <p>PCIe posted write: IOH drops the write data.</p> | <p>Log error per PCI Express AER requirements for completer abort.⁴</p> <p>Log in XPGLBERRSTS, XPGLBERRPTR registers</p> <p>If PCIe uncorrectable error is forwarded to the global error registers, it is logged in global non-fatal log registers – GNERRST, GNFERRST, GNNERRST, GNFERRTIME</p> |



Table 16-3. IOH Error Summary (Sheet 4 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|----|---|------------------------|---|---|
| 84 | Completion time-out on NP transactions outstanding on PCI Express/ESI | | <p>Intel QuickPath Interconnect to PCIe read: IOH returns normal response to Intel QuickPath Interconnect and all 1's for read data</p> <p>Intel QuickPath Interconnect to PCIe non-posted write: IOH returns normal response to Intel QuickPath Interconnect</p> <p>PCIe to PCIe read/non-posted write: UR² is returned on PCIe</p> <p>SMBus/Jtag reads: IOH returns a UR status on SMBus/jtag</p> | <p>Log error per PCI Express AER requirements for the corresponding error.</p> <p>Log in XPGLBERRSTS, XPGLBERRPTR registers</p> <p>If PCIe uncorrectable error is forwarded to the global error registers, it is logged in global non-fatal log registers – GNERRST, GNFERRST, GNNERRST, GNFERRTIME</p> <p>Note:</p> <p>a) A poisoned TLP received from PCIE is always treated as advisory-nonfatal error, if the associated severity is set to nonfatal.</p> |
| 85 | Received PCIe Poisoned TLP | 1 | <p>Intel QuickPath Interconnect to PCIe read: IOH returns normal response and poisoned data to Intel QuickPath Interconnect, if Intel QuickPath Interconnect profile supports poisoned data. Otherwise, packet is dropped and no response sent on Intel QuickPath Interconnect.</p> <p>PCIe to Intel QuickPath Interconnect write: IOH forwards poisoned indication to Intel QuickPath Interconnect, if Intel QuickPath Interconnect profile supports poisoned data. Otherwise, write is dropped.</p> <p>PCIe to PCIe read: IOH forwards completion with poisoned data to original requester, if the root port in the outbound direction for the completion packet, is not in 'Stop and Scream' mode. If the root port is in 'Stop and scream' mode, the packet is dropped and the link is brought down immediately (that is, no packets on or after the poisoned data is allowed to go to the link).</p> <p>PCIe to PCIe posted/non-posted write: IOH forwards write with poisoned data to destination link, if the root port of the destination link, is not in 'Stop and Scream' mode. If the root port is in 'Stop and scream' mode, the packet is dropped and the link is brought down immediately (that is, no packets on or after the poisoned data is allowed to go to the link) and a UR² response is returned to the original requester, if the request is non-posted.</p> <p>SMBus/Jtag to IOH accesses requests: IOH returns a UR response status on smbuse/jtag</p> | <p>b) When a poisoned TLP is transmitted down a PCIe link, the IOH does not log that condition in the AER registers.</p> |
| 86 | Received PCIe unexpected Completion | | | <p>Log error per PCI Express AER requirements for the corresponding error/message.</p> <p>Log in XPGLBERRSTS, XPGLBERRPTR registers</p> |
| 87 | PCIe Flow Control Protocol Error ⁵ | | Respond Per PCIe Specification | |
| 88 | Received ERR_NONFATAL Message from downstream device | | | <p>If PCIe uncorrectable error is forwarded to the global error registers, it is logged in global non-fatal log registers – GNERRST, GNFERRST, GNNERRST, GNFERRTIME</p> |



Table 16-3. IOH Error Summary (Sheet 5 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|-------------------|--|------------------------|--|---|
| 90 | PCIe Malformed TLP ⁵ | 2 | Respond Per PCIe Specification | Log error per PCI Express AER requirements for the corresponding error/message. Log in XPGLBERRSTS, XPGLBERRPTR registers If PCIe uncorrectable error is forwarded to the global error registers, it is logged in global non-fatal log registers – GFERRST, GFFERRST, GFNERRST, GFFERRTIME |
| 91 | PCIe Data Link Protocol Error ⁵ | | | |
| 92 | PCIe Receiver Overflow | | | |
| 93 | Surprise Down | | | |
| 94 | Received ERR_FATAL message from downstream device | | | |
| 98 | MSI writes greater than a DWORD | 2 | Drop the transaction | Log in XPUNCERRSTS register If error is forwarded to the global error registers, it is logged in global non-fatal log registers – GFERRST, GFFERRST, GFNERRST, GFFERRTIME |
| Intel VT-d | | | | |
| A1 | All faults except ATS spec defined CA faults (refer to Intel VT-d spec for complete details) | 1 | Unsupported Request response for the associated transaction on the PCI Express interface | Error logged in Intel VT-d Fault Record register. Error logged in XPGLBERRSTS and XPGLBERRPTR registers. Error logging also happens (on the GPA address) per the PCI Express AER mechanism (address logged in AER is the GPA). Errors can also be routed to the IOH global error logic and logged in the global non-fatal registers GNERST GNFERRST GNFERRTIME GNNERRST |
| A2 | ATS spec defined CA faults (refer to Intel VT-d spec for complete details) | 1 | Completer Abort response for the associated transaction on the PCI Express interface | Error logged in Intel VT-d fault record register Error also logged in the VTUNCERRSTS and in VTUNCERRPTR registers. Error logging also happens (on the GPA address) per the PCI Express AER mechanism (address logged in AER is the GPA). Errors can also be routed to the IOH global error logic and logged in the global non-fatal registers GFERRST GFFERRST GFFERRTIME GFNERRST |
| A3 | Fault Reason Encoding 0xFF – Miscellaneous errors that are fatal to Intel VT-d unit operation (for example, parity error in an Intel VT-d cache) | 2 | Drop the transaction. Continued operation of IOH is not guaranteed. | Error logged in Intel VT-d fault record register Error also logged in the VTUNCERRSTS and in VTUNCERRPTR registers. These errors can also be routed to the IOH global error logic and logged in the global fatal registers GFERRST GFFERRST GFFERRTIME GFNERRST |



Table 16-3. IOH Error Summary (Sheet 6 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|----|---|------------------------|----------------------|--|
| A4 | Data parity error while doing a context cache look up | 2 | | Log in VTUNCERRSTS and VTUNCERRPTR registers These errors can also be routed to the IOH global error logic and logged in the global fatal registers. GFERRST GFFERRST GFFERRTIME GFNERRST |
| A4 | Data parity error while doing a L1 lookup | 2 | | Log in VTUNCERRSTS and VTUNCERRPTR registers These errors can also be routed to the IOH global error logic and logged in the global fatal registers GFERRST GFFERRST GFFERRTIME GFNERRST |
| A4 | Data parity error while doing a L2 lookup | 2 | | Log in VTUNCERRSTS and VTUNCERRPTR registers These errors can also be routed to the IOH global error logic and logged in the global fatal registers GFERRST GFFERRST GFFERRTIME GFNERRST |
| A4 | Data parity error while doing a L3 lookup | 2 | | Log in VTUNCERRSTS and VTUNCERRPTR registers. These errors can also be routed to the IOH global error logic and logged in the global fatal registers. GFERRST GFFERRST GFFERRTIME GFNERRST |
| A4 | TLB0 parity error | 2 | | Log in VTUNCERRSTS and VTUNCERRPTR registers These errors can also be routed to the IOH global error logic and logged in the global fatal registers. GFERRST GFFERRST GFFERRTIME GFNERRST |
| A4 | TLB1 parity error | 2 | | Log in VTUNCERRSTS and VTUNCERRPTR registers. These errors can also be routed to the IOH global error logic and logged in the global fatal registers. GFERRST GFFERRST GFFERRTIME GFNERRST |



Table 16-3. IOH Error Summary (Sheet 7 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|--|---|------------------------|--|--|
| A4 | Unsuccessful status received in CSI read completion | 1 | | Log in VTUNCERRSTS and VTUNCERRPTR registers. These errors can also be routed to the IOH global error logic and logged in the global non-fatal registers GNERRST GNFERRST GNFERRTIME GNNERRST |
| A4 | Protected memory region space violated | 2 | | Log in VTUNCERRSTS and VTUNCERRPTR registers. These errors can also be routed to the IOH global error logic and logged in the global fatal registers GFERRST GFFERRST GFFERRTIME GFNERRST |
| Intel QuickPath Interconnect Errors | | | | |
| B0 | Intel QuickPath Interconnect Link Layer detected CRC error – Successful Link Level Retry and Unsuccessful Link Level Retry (entered LLR abort state) | 0 | IOH processes and responds the cycle as normal. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers: QPINFERRST QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST No header is logged for this error |
| B1 | Intel QuickPath Interconnect Link Layer detected CRC error -- Successful Link Level Retry after PHY reinit | 0 | IOH processes and responds the cycle as normal | FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers: QPINFERRST QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST No header is logged for this error |
| B2 | Intel QuickPath Interconnect Physical Layer Detected an Intel QuickPath Interconnect Inband Reset (either received or driven by the IOH) and re-initialization completed successfully | 0 | No Response – This event is not associated with a cycle. IOH detects and logs the event. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers and Intel QuickPath Interconnect Physical layer register: QPINFERRST QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST QPIPHPIIS |



Table 16-3. IOH Error Summary (Sheet 8 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|----|--|------------------------|---|--|
| B3 | Intel QuickPath Interconnect Protocol Layer Received CPEI message from Intel QuickPath Interconnect (see Chapter 8, "Interrupts" for detailed flow). | 0 | No Response <i>Note:</i> This is really not an error condition but exists for monitoring by an external management controller. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers and Intel QuickPath Interconnect Physical layer register: QPINFERRST QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST QPIHPIS |
| B4 | Intel QuickPath Interconnect Write Cache Detected ECC Correctable Error | 0 | IOH processes and responds the cycle as normal | FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers: QPIPINFERRST QPIPNNERRST GNERRST GNFERRST GNFERRTIME GNNERRST No header is logged for this error |
| B5 | Potential spurious CRC error on L0s/L1 exit | 1 | In the event CRC errors are detected by link layer during L0s/L1 exit, it will be logged as "Potential spurious CRC error on L0s/L1 exit". IOH processes and responds the cycle as normal | FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers and Intel QuickPath Interconnect Link layer register: QPINFERRST QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST |
| B6 | Intel QPI Link Layer CRC error | 0 | In the event CRC errors are detected by link layer, it will be logged as "QPI Link Layer CRC error". IOH processes and responds the cycle as normal | FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers and Intel QuickPath Interconnect Link layer register: QPINFERRST QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST |



Table 16-3. IOH Error Summary (Sheet 9 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|----|---|------------------------|--|--|
| C0 | Intel QuickPath Interconnect Link Layer Detected CRC error -- Unsuccessful Link Level Retry (entered LLR abort state) | 2 | Link goes down and ORB timeout occurs. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers: QPINFERRST QPINNERRST QPIFERRFLIT GNERRST GNFERRST GNFERRTIME GNNERRST No header is logged for this error |
| C1 | Intel QuickPath Interconnect Protocol Layer Received Poisoned packet | 1 | Intel QuickPath Interconnect to PCIe write: IOH returns normal response to Intel QuickPath Interconnect and forwards poisoned data to PCIe. Intel QuickPath Interconnect to IOH write: IOH returns normal response to Intel QuickPath Interconnect and drops the write data. PCIe to Intel QuickPath Interconnect read: IOH forwards the poisoned data to PCIe IOH to Intel QuickPath Interconnect read: IOH drops the data. IOH to Intel QuickPath Interconnect read for RFO: IOH completes the write. If the bad data chunk is not overwritten, IOH corrupts write cache ECC to indicate the stored data chunk (64-bit) is poisoned. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers: QPIPNFERRST QPIPNFERRHD QPIPNNERRST GNERRST GNFERRST GNFERRTIME GNNERRST Intel QuickPath Interconnect header is logged |
| C2 | IOH Write Cache uncorrectable Data ECC error | 1 | Write back includes poisoned data. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers: QPIPNFERRST QPIPNERRST GNERRST GNFERRST GNFERRTIME GNNERRST |
| C3 | IOH CSR access crossing 32-bit boundary | 1 | Intel QuickPath Interconnect read: IOH returns all '1s' and normal response to Intel QuickPath Interconnect to indicate master abort Intel QuickPath Interconnect write: IOH returns normal response and drops the write | FERR/NERR is logged in IOH Core and Global Non-Fatal Error Log Registers: QPIPNFERRST QPIPNFERRHD QPIPNNERRST GNERRST GNFERRST GNFERRTIME GNNERRST Intel QuickPath Interconnect header is logged |



Table 16-3. IOH Error Summary (Sheet 10 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|----|---|------------------------|---|--|
| C7 | Intel QuickPath Interconnect Physical Layer Detected a Intel QuickPath Interconnect Inband Reset (either received or driven by the IOH) and re-initialization completed successfully but width is changed | 1 | No Response -- This event is not associated with a cycle. IOH detects and logs the event. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers and Intel QuickPath Interconnect physical layer register: QPINFERRST QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST QPIPHPIIS |
| D0 | Intel QuickPath Interconnect Physical Layer Detected Drift Buffer Alarm | 2 | No Response – This error is not associated with a cycle. IOH detects and logs the error. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers, and Intel QuickPath Interconnect Physical layer CSRs: |
| D1 | Intel QuickPath Interconnect Physical Layer Detected Latency Buffer Rollover (Only supported for tester determinism) | | | QPIFFERRST QPIFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST |
| D2 | Intel QuickPath Interconnect Physical Layer Initialization Failure | | | No header logged for this error |
| D3 | Intel QuickPath Interconnect Link Layer Detected Control Error (Buffer Overflow or underflow, illegal or unsupported LL control encoding, credit underflow) | 2 | No Response – This error is not associated with a cycle. IOH detects and logs the error. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers: QPIFFERRST QPIFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST No header logged for this error |
| D4 | Intel QuickPath Interconnect Parity Error (Link or Physical layer) | 2 | No Response – This error is not associated with a cycle. IOH detects and logs the error. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers: QPIFFERRST QPIFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST |



Table 16-3. IOH Error Summary (Sheet 11 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|----|---|------------------------|--|--|
| D5 | Intel QuickPath Interconnect Protocol Layer Detected Time-out in ORB | 2 | <p>Intel QuickPath Interconnect read: return completer abort.</p> <p>Intel QuickPath Interconnect non-posted write: IOH returns completer abort</p> <p>Intel QuickPath Interconnect posted write: no action</p> | <p>FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers:</p> <p>QPIPFERRST QPIPFERRHD QPIPFNERRST</p> <p>GFERRST GFFERRST GFFERRTIME GFNERRST</p> <p>Intel QuickPath Interconnect header is logged</p> |
| D6 | Intel QuickPath Interconnect Protocol Layer Received Failed Response | | | <p>FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers:</p> <p>QPIPFERRST QPIPFERRHD QPIPFNERRST</p> <p>GFERRST GFFERRST GFFERRTIME GFNERRST</p> <p>Intel QuickPath Interconnect header is logged</p> |
| D7 | Intel QuickPath Interconnect Protocol Layer Received Unexpected Response/Completion | 2 | <p>Drop Transaction, No Response. This will cause time-out in the requester.</p> | <p>FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers:</p> <p>QPIPFERRST QPIPFERRHD QPIPFNERRST</p> <p>GFERRST GFFERRST GFFERRTIME GFNERRST</p> <p>Intel QuickPath Interconnect header is logged</p> |
| D8 | Intel QuickPath Interconnect Protocol Layer Received illegal packet field or incorrect target Node ID | | | <p>FERR/NERR is logged in Intel QuickPath Interconnect and Global Non-Fatal Error Log Registers:</p> <p>QPIPFERRST QPIPFNERRST</p> <p>GFERRST GFFERRST GFFERRTIME GFNERRST</p> <p>No header logged for this error</p> |
| D9 | Intel QuickPath Interconnect protocol received Viral indication in the Intel QuickPath Interconnect Packet. | 2 | <p>The Intel QuickPath Interconnect port become viral causing all subsequent packets, sent or received, to have viral set. Outbound request to PCIe will be dropped and a Failed Response sent to the originating Intel QPI agent. If the request to PCIe is initiated by the SMBus agent, it proceeds as normal. Inbound read completions will be converted to completer abort. Inbound writes are either dropped or forwarded.</p> | <p>FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers:</p> <p>QPIPFERRST QPIPFNERRST</p> <p>GFERRST GFFERRST GFFERRTIME GFNERRST</p> <p>No header logged for this error</p> |
| DA | Intel QuickPath Interconnect Protocol Layer Queue/Table Overflow or Underflow | 2 | <p>No Response – This error is not associated with a cycle. IOH detects and logs the error.</p> | <p>FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers:</p> <p>QPIPFERRST QPIPFNERRST</p> <p>GFERRST GFFERRST GFFERRTIME GFNERRST</p> <p>No header logged for this error</p> |



Table 16-3. IOH Error Summary (Sheet 12 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|----|--|------------------------|--|---|
| DB | Intel QuickPath Interconnect Protocol Parity Error. | 2 | No Response – This error is not associated with a cycle. IOH detects and logs the error. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers: QPIPFERRST QPIPFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST |
| DC | IOH SAD illegal or non-existent memory for outbound snoop | 2 | Drop Transaction, No Response. This will cause time-out in the requester for non-posted requests. (for example, completion time-out in Intel QuickPath Interconnect request agent, or PCIe request agent.) | FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers: |
| DE | IOH Routing Table invalid or non-existent entry reference | | | QPIPFERRST QPIPFERRHD QPIPFNERRST |
| DF | Illegal inbound request (includes VCp/VC1 request when they are disabled) | | | GFERRST GFFERRST GFFERRTIME GFNERRST Intel QuickPath Interconnect header is logged |
| DG | Intel QuickPath Interconnect Link Layer detected unsupported/undefined packet (e.g., RSVD_CHK, message class, opcode, vn, viral) | 2 | No Response -- This error is not associated with a cycle. IOH detects and logs the error. | FERR/NERR is logged in Intel QuickPath Interconnect and Global Fatal Error Log Registers: QPIFFERRST QPIFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST No header logged for this error |
| DH | Intel QuickPath Interconnect Protocol Layer Detected unsupported/undefined packet Error (message class, opcode and vn only) | 2 | No Response -- This error is not associated with a cycle. IOH detects and logs the error. | FERR/NERR is logged in Intel QuickPath Interconnect Protocol and Global Fatal Error Log Registers: QPIPFERRST QPIPFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST |



Table 16-3. IOH Error Summary (Sheet 13 of 13)

| ID | Error | Default Error Severity | Transaction Response | Default Error Logging ¹ |
|----------------------|----------------------------|------------------------|---|--|
| Thermal Error | | | | |
| F0 | Thermal Alert | 1 | No Response -- This error is not associated with a cycle. IOH detects and logs the error. | |
| F1 | TSMAX Updated | 0 | | FERR/NERR is logged in Thermal and Global Non-Fatal Error Log Registers: CTSTS.PKTRK |
| F2 | Catastrophic Thermal Event | 2 | | FERR/NERR is logged in Thermal and Global Fatal Error Log Registers: CTSTS.THRMTRIP |

Notes:

1. This column notes the logging registers used assuming the error severity default remains. The error's severity dictates the actual logging registers used upon detecting an error.
2. It is possible that when a UR response is returned to the original requester, the error is logged in the AER of the root port connected to the requester.
3. It is possible that when a CA response is returned to the original requester, the error is logged in the AER of the root port connected to the requester.
4. Note that in some cases, IOH might not be able to log the error/header in AER when it signals CA back to the PCIe device.
5. Not all cases of this error are detected by IOH.

16.8 IOH PCIe Hot Add/Remove Support

Hot add/remove is the ability to add or remove a component without requiring the system to reboot. There are two types of hot add/remove:

- **Physical Hot add/remove**

This is the conventional hot-plug of a physical component in the system. For example, an operator may decide to add a CPU to a running system. He/she may issue a hot-plug request to the system through the console interface. The console informs the operator of the appropriate steps of inserting the CPU. Once the CPU is inserted, the system incorporates the new CPU into the system.

- **Logical Hot add/remove**

Logical hot add/remove differs from physical hot add/remove by not requiring physical removal or addition of a component. A component can be taken out of the system without the physical removal. Similarly, a disabled component can be hot added to the system. Logical hot add/remove enables dynamic partitioning, and allows resources to move in and out of a partition.

The IOH supports both physical and logical hot add/remove of various components in the system. These include:

- **CPU**

Intel® 7500 chipset based platforms support CPU socket hot add/remove including the memory behind the CPU. CPU hot add/remove allows physical hot-plug/removal of a CPU, and enables dynamic partitioning of CPU components. Support of hot add/removal of CPU is expected to be restricted in some topologies due to hardware and software constraints. These constraints are explained in [Section 16.8.1](#).

- **Memory**

Intel 7500 chipset based platforms support memory hot add/remove with memory mirroring. Hot add/remove of memory allows physical hot-plug/removal of a memory component. Memory is a subcomponent of the CPU, hence hot add and remove of memory is expected to be coordinated by the CPUs or BMC, and does not involve IOHs.



- **IOH**

Intel 7500 chipset based platforms support IOH hot add/remove. This feature allows physical hot-plug/removal of a IOH, and enables dynamic partitioning of IOH components. Support of hot add/removal of IOH is expected to be restricted in some topologies due to the hardware and software constraints. These constraints are explained in [Section 16.8.1](#).

Note: Hot-plug events where the dual IOH configuration must be changed require a reset.

Note: The IOH does not support hot add/remove of the Legacy IOH and Legacy Bridge. The legacy IOH and its associated legacy bridge must be active at all times to provide legacy functions to the partition. The legacy IOH and its legacy bridge cannot be hot added/removed without shutting down the entire partition.

- **PCI Express and IO devices**

Intel 7500 chipset based platforms support PCIe and IO device hot add/remove. This feature allows physical hot-plug/removal of an PCIe device connected to the IOH, and enables dynamic partitioning of the PCIe ports in IOH. In addition, physical hot-plug/remove for other IO devices downstream to IOH may be supported by downstream bridges. Hot-plug of PCIe and IO devices are well defined in PCIe/PCI specifications.

16.8.1 Hot Add/Remove Rules

1. Legacy IOH cannot be hot added/removed without shutting down the system.
2. Legacy bridge (ICH) itself cannot be hot added/removed from the IOH (no ESI hot-plug support).
3. Hot add/remove of the IOH is done through Intel QuickPath Interconnect hot add/remove (see [Section 16.10.1](#).)
4. Hot add/remove of a field replaceable unit (FRU) is done through Intel QuickPath Interconnect hot add/remove (see [Section 16.10](#).)
5. IOH is accessed via Intel QuickPath Interconnect link. Hot removing CPU or IOH must not cut off all Intel QuickPath Interconnect paths of any active IOH unless all devices below the orphaned IOH are first disabled (hot-removed).
6. IOH does not provide Intel QuickPath Interconnect route-through. Hot removing CPUs must not cut off all CPU-CPU Intel QuickPath Interconnect paths to an active CPU.

16.8.2 PCI Express Hot-Plug

PCI Express hot-plug is supported through the standard PCI Express native hot-plug mechanism. The IOH supports the sideband hot-plug signals; it does not support inband hot-plug messages. The IOH contains a Virtual Pin Port (VPP) that serially shifts the sideband PCI Express hot-plug signals in and out. External platform logic is required to convert the IOH serial stream to parallel. The virtual pin port is implemented via a dedicated SMBus port. The PCI Express hot-plug model implies a hot-plug controller per port, which is identified to software as a PCI Express capability of the peer-to-peer Bridge configuration space. Refer to the *PCI Express Base Specification*, Revision 2.0 for further details.

Summary of IOH PCI Express hot-plug support:

- Support for up to nine hot-plug slots, selectable by BIOS.
- Support for serial mode hot-plug only, using smbus devices such as PCA9555.



- Single SMBus is used to control hot-plug slots.
- Support for CEM/SIOM/Cable form factors.
- Support MSI or ACPI paths for hot-plug interrupts.
- Hot-plug is not supported on a non-legacy IOH ESI port or when the ESI port is used as a PCIe port.
- The IOH does not support inband hot-plug messages on PCIe:
 - The IOH does not issue these and the IOH discards them silently if received.
- A hot-plug event cannot change the number of ports of the PCIe interface (that is, bifurcation).

16.8.2.1 PCI Express Hot-Plug Interface

Table 16-4 describes the hot-plug signals supplied by the IOH for each PCI Express port. These signals are controlled and reflected in the PCI Express root port hot-plug registers.

Table 16-4. Hot-Plug Interface (Sheet 1 of 2)

| Signal Name | Description | Action |
|-------------|---|---|
| ATNLED | This indicator is connected to the Attention LED on the baseboard. For a precise definition refer to <i>PCI Express Base Specification, Revision 1.0a and the associated set of Erratas and EC*s</i> . | Indicator can be off, on, or blinking. The required state for the indicator is specified with the Attention Indicator Register. The IOH blinks this LED at 1Hz. |
| PWRLED | This indicator is connected to the Power LED on the baseboard. For a precise definition refer to <i>PCI Express Base Specification, Revision 1.0a and the associated set of Erratas and EC*s</i> . | Indicator can be off, on, or blinking. The required state for the indicator is specified with the Power Indicator Register. The IOH blinks this LED at 1 Hz. |
| BUTTON# | Input signal per slot which indicates that the user wishes to hot remove or hot add a PCI Express card/module. | If the button is pressed (BUTTON# is asserted), the Attention Button Pressed Event bit is set and either an interrupt or a general-purpose event message Assert/Deassert_HPGPE to the ICH is sent. ¹ |
| PRSNT# | Input signal that indicates if a hot-pluggable PCI Express card/module is currently plugged into the slot. | When a change is detected in this signal, the Presence Detect Event Status register is set and either an interrupt or a general-purpose event message Assert/Deassert_HPGPE is sent to the ICH. ¹ |
| PWRFLT# | Input signal from the power controller to indicate that a power fault has occurred. | When this signal is asserted, the Power Fault Event Register is set and either an interrupt or a general-purpose event message Assert/Deassert_HPGPE message is sent to the ICH. ¹ |
| PWREN | Output signal allowing software to enable or disable power to a PCI Express slot. | If the Power Controller Register is set, the IOH asserts this signal. |
| MRL#/EMILS | Manual retention latch status or Electro-mechanical latch status input indicates that the retention latch is closed or open. Manual retention latch is used on the platform to mechanically hold the card in place and can be open/closed manually. Electromechanical latch is used to electromechanically hold the card in place and is operated by software. MRL# is used for card-edge and EMLSTS# is used for SIOM formfactors. | Supported for the serial interface and MRL change detection results in either an interrupt or a general-purpose event message Assert/Deassert_HPGPE message is sent to the ICH. ¹ |



Table 16-4. Hot-Plug Interface (Sheet 2 of 2)

| Signal Name | Description | Action |
|-------------|--|---|
| EMIL | Electromechanical retention latch control output that opens or closes the retention latch on the board for this slot. A retention latch is used on the platform to mechanically hold the card in place. Refer to <i>PCI Express Server/Workstation Module Electromechanical Spec Rev 0.5a</i> for details of the timing requirements of this pin output. | Supported for the serial interface and is used only for the SIOM form-factor. |

Notes:

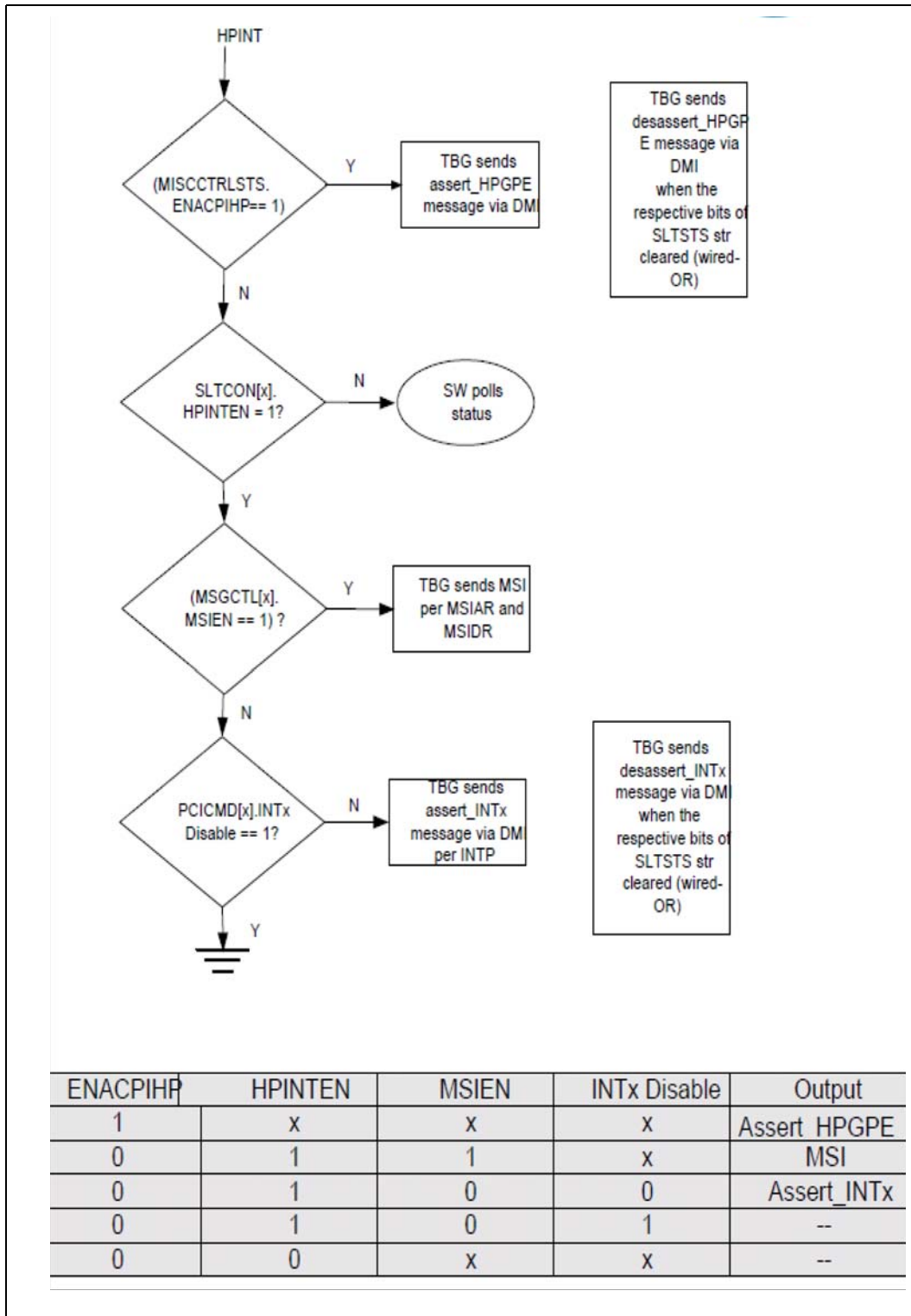
1. For legacy operating systems, the described Assert_HPGPE/Deassert_HPGPE mechanism is used to interrupt the platform for PCI Express hot-plug events. For newer operating systems, this mechanism is disabled and the MSI capability is used by the IOH instead.

16.8.2.2 PCI Express Hot-Plug Interrupts

The IOH generates an Assert/Deassert_HPGPE message to the ICH over the ESI link or an MSI when a hot-plug event occurs on any of its standard PCI Express interfaces. Refer to [Figure 16-12](#) for the hot-plug interrupt flow priority. The GPE messages are selected when bit 3 in the Miscellaneous Control and Status Register (MISCCTRLSTS) is set. Refer to [Section 21.12.3.13](#). If this bit is clear, the MSI method is selected (note that the MSI Enable bit in the (MSIX)MSGCTRL register does not control selection of GPE versus MSI method). Refer to the *PCI Express Base Specification, Revision 2.0* for details of MSI generation on a PCI Express Hot-Plug event. This section covers how the GPE event is generated for PCI Express hot-plug events.

PCI Express hot-plug events are defined as a set of actions: Command completed, Presence Detect changed, MRL sensor changed, power fault detected, Attention button pressed and data Link layer state changed events. Each of these hot-plug events have a corresponding bit in the PCI Express Slot status and control registers. The IOH processes hot-plug events using the wired-OR (collapsed) mechanism to emulate the level sensitive requirement for the legacy interrupts on ESI. When the wired-OR output is set, the Assert_HPGPE is sent to the ICH. When software clears all the associated register bits (that are enabled to cause an event) across the ports, the IOH will generate a Deassert_HPGPE message to the ICH. Refer to [Chapter 8, "Interrupts,"](#) for details of how these messages are routed to the ICH. Note that Assert/Deassert_HPGPE messages could be received from downstream of a PCIe port (when that port connects to a downstream IOH) and these messages are collapsed with internally generated PMEGPE virtual wires as well.

Figure 16-12. PCI Express Hot-Plug Interrupt Flow





16.9 Virtual Pin Ports (VPP)

The IOH contains a VPP that serially shifts the sideband PCI Express Hot-Plug signals in and out. VPP is a dedicated 100 KHz SMBus interface that connects to a number of serial to parallel I/O devices, such as the PCA9555. The PCA9555 supports 16 GPIOs structured as two 8-bit ports, with each GPIO configurable as an input or an output. Reading or writing to the PCA9555 component with a specific command value reads or writes the GPIOs or configures the GPIOs to be either input or output. The IOH supports up to nine PCIe Hot-Plug ports through the VPP interface with maximum of 5 PCA9555, or similar devices, populated.

The IOH VPP supports SMBus devices with command sequence as shown [Table 16-5](#). Each PCI Express port is associated with one of the 8-bit ports of the serial-to-parallel I/O device. The mapping is defined by a Virtual Pin Port register field in the VPP control register (VPPCTRL) for each PCIe slot. The VPP register holds the SMBus address and Port (0 or 1) of the I/O Port associated with the PCI Express port. A[1:0] pins on each I/O Extender (that is, PCA9555, and so on) connected to the IOH must be strapped uniquely.

Table 16-5. I/O Port Registers in On-Board SMBus Devices Supported by IOH

| Command | Register | IOH Usage |
|---------|---------------------------|-----------------------------------|
| 0 | Input Port 0 | Continuously Reads Input Values |
| 1 | Input Port 1 | |
| 2 | Output Port 0 | Continuously Writes Output Values |
| 3 | Output Port 1 | |
| 4 | Polarity Inversion Port 0 | Never written by IOH |
| 5 | Polarity Inversion Port 1 | |
| 6 | Configuration Port 0 | Direction (Input/Output) |
| 7 | Configuration Port 1 | |

16.10 Operation

When the IOH comes out of Powergood reset, the I/O ports are inactive. The IOH is not aware of how many I/O extenders are connected to the VPP, what their addresses are, nor what PCI Express ports are hot-pluggable. The IOH does not master any commands on the SMBus until a VPP enable bit is set.

For PCI Express 1.0a slots, an additional form factor (FF) bit the VPP control register (VPPCTRL) is used to differentiate card, module or cable hot-plug support. When BIOS sets the Hot-Plug Capable bit in the root port PCI Express capability register for the first time, the IOH initializes the associated VPP corresponding to that root port with direction and logic level configuration. From then on, the IOH continually scans in the inputs and scans out the outputs corresponding to that port. VPP registers for PCI Express 1.0a ports which do not have the VPP enable bit set are invalid and ignored.

[Table 16-6](#) defines how the eight hot-plug signals are mapped to pins on the I/O extender's GPIO pins. When the IOH is not doing a direction or logic level write (which would happen when a PCI Express port is first setup for hot-plug), it performs input register reads and output register writes to all valid VPPs. This sequence repeats indefinitely until a new VPP enable bit is set. To minimize the completion time of this sequence, both ports in the external device are written or read in any sequence. If only



one port of the external device has yet been associated with a hot-plug capable root port, the value read from the other port of the external device are discarded and only de-asserted values are shifted out for the outputs. See [Table 16-6](#) for the details.

Table 16-6. Hot-Plug Signals on the Virtual Pin Port

| Bit | Direction | Voltage Logic Table | Signal | Logic True Meaning | Logic False Meaning |
|-------|-----------|------------------------|------------|---|--|
| Bit 0 | Output | High_True | ATNLED | ATTN LED is to be turned ON | ATTN LED is to be turned OFF |
| Bit 1 | Output | High_True | PWRLED | PWR LED is to be turned ON | PWR LED is to be turned OFF |
| Bit 2 | Output | High_True | PWREN | Power is to be enabled on the slot | Power is NOT to be enabled on the slot |
| Bit 3 | Input | Low_True | BUTTON# | ATTN Button is pressed | ATTN Button is NOT pressed |
| Bit 4 | Input | Low_True | PRSNT# | Card Present in slot | Card NOT Present in slot |
| Bit 5 | Input | Low_True | PWRFLT# | PWR Fault in the VRM | NO PWR Fault in the VRM |
| Bit 6 | Input | Low_True/ High_True | MRL#/EMILS | MRL is closed/ EMILS is disengaged | MRL is open/EMILS is engaged |
| Bit 7 | Output | High_True | EMIL | Toggle interlock state -Pulse output 100 ms when '1' is written | No effect |

[Table 16-7](#) describes the sequence generated for a write to an I/O port. Both 8-bit ports are always written. If a VPP is valid for the 8-bit port, the output values are updated as per the PCI Express Slot Control register for the associated PCI Express slot.

Table 16-7. Write Command

| Bits | IOH Drives | I/O Port Drives | Comment |
|------|--------------|-----------------|---|
| 1 | Start | | SDL falling followed by SCL falling |
| 7 | Address[6:0] | | [6:3] = 0100 [2:0] = <Per the VPP Control Register (VPPCTL)> |
| 1 | 0 | | indicates write. |
| 1 | | ACK | If NACK is received, IOH completes with stop and sets status bit in the VPP Status Register (VPPSTS). |
| 8 | Command Code | | Register Address see Table 16-5 [7:3]=00000,[2:1] = 01 for Output, 11 for Direction [0] = 0 |
| 1 | | ACK | If NACK is received, IOH completes with stop and sets in the VPP Status Register (VPPSTS). |
| 8 | Data | | One bit for each I/O as per Table 16-6 . |
| 1 | | ACK | If NACK is received, IOH completes with stop and sets in the VPP Status Register (VPPSTS). |
| 8 | Data | | One bit for each I/O as per Table 16-6 |
| 1 | | ACK | If NACK is received, IOH completes with stop and sets in the VPP Status Register (VPPSTS). |
| 1 | Stop | | |



The IOH issues Read Commands to update the PCIe Slot Status register from the I/O port. The I/O port requires that a command be sent to sample the inputs, then another command is issued to return the data. The IOH always reads inputs from both 8-bit ports. If the VPP is valid, the IOH updates the associated PEXSLOTSTS (for PCIe) register according to the values of MRL#/EMLSTS#, BUTTON#, PWRFLT# and PRSNT# read from the value register in the I/O port. Results from invalid VPPs are discarded. [Table 16-8](#) defines the read command format.

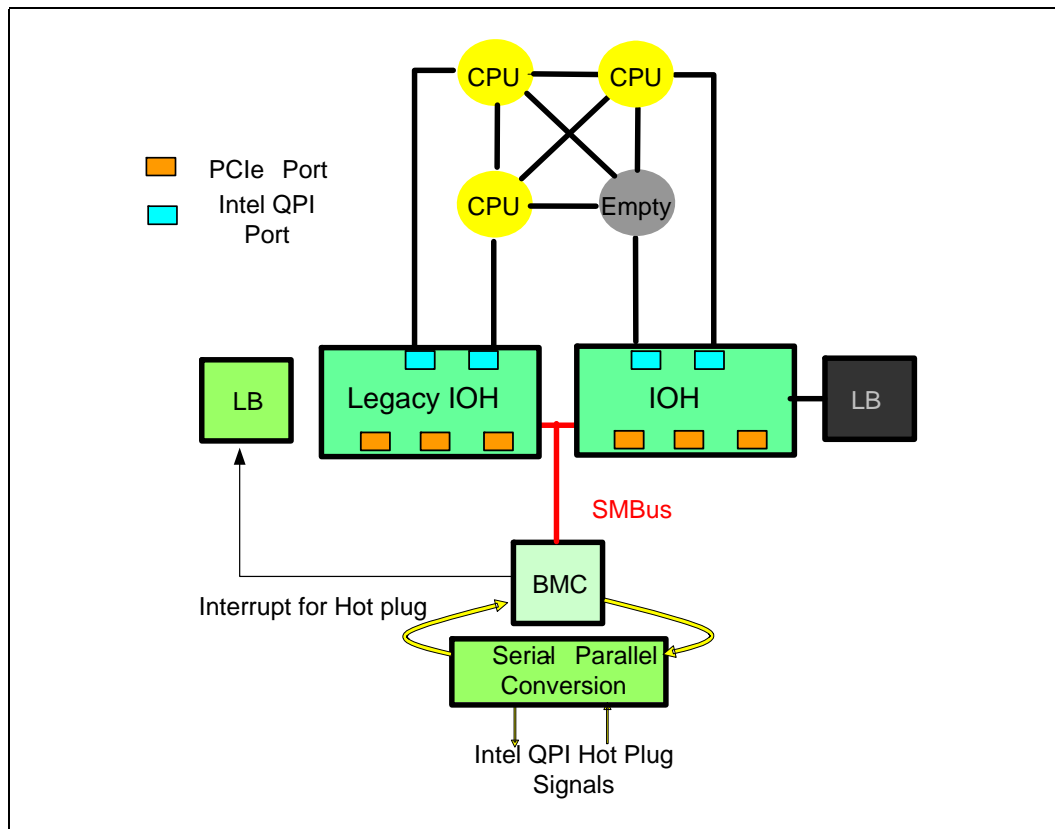
Table 16-8. Read Command

| Bits | IOH Drives | I/O Port Drives | Comment |
|------|--------------|-----------------|--|
| 1 | Start | | SDL falling followed by SCL falling. |
| 7 | Address[6:0] | | [6:2] = 01000 [1:0] = <per the VPP Control Register (VPPCTL)> |
| 1 | 0 | | indicates write. |
| 1 | | ACK | If NACK is received, IOH completes with stop and sets in the VPP Status Register (VPPSTS). |
| 8 | Command Code | | Register Address [2:0] = 000 |
| 1 | | ACK | If NACK is received, IOH completes with stop and sets in the VPP Status Register (VPPSTS). |
| 1 | Start | | SDL falling followed by SCL falling. |
| 7 | Address[6:0] | | [6:2] = 01000 [1:0] = per the VPP Control Register (VPPCTL)> |
| 1 | 1 | | indicates read. |
| 8 | | Data | One bit for each I/O as per Table 16-6 . The IOH always reads from both ports. Results for invalid VPPs are discarded. |
| 1 | ACK | | |
| 8 | | Data | One bit for each I/O as per Table 16-6 . The IOH always reads from both ports. Results for invalid VPPs are discarded. |
| 1 | NACK | | |
| 1 | Stop | | |

16.10.1 Intel QuickPath Interconnect Hot-Plug

Refer to [Figure 16-13](#). In the IOH, Intel QPI hot-plug is performed by the BMC. IOH provides a minimal set of hardware to assist Intel QPI hot-plug operation. In this model, BMC monitors and controls all hot-plug signals for Intel QPI agents (CPU and IOH). BMC is required to generate interrupts to request CPU for hot-plug event services. Interrupt to CPU is delivered through the assertion of GPIO pins provided by the ICH. The supported interrupts include PMI/SMI and SCI.

Figure 16-13. Intel QPI Hot Add/Remove Support



16.10.1.1 IOH Hardware Support for Intel QPI Hot-Plug

IOH core provides the following features for Intel QPI hot add/remove support.

1. Intel QPI participant list
IOH provides a participant list (QPIPSB, QPIPNCB, QPIPLKBL, QPIQBL registers). This is realized through a 32-bit vector, with each bit representing a NodeID. This means IOH supports only local broadcast where the maximum number of the node IDs is 32. The participation list will be modified by firmware for hot add/remove of IOH or CPU.
2. System Quiescence support
Refer to [Section 4.7](#), IOH provides a mechanism to quiesce the system through combination of inband message cycle and CSR bits. Writing to IOH quiescence CSR bit (QPIPQC) triggers the IOH to issue SpcStopReq messages to all nodes in the partition of the hot add/remove operation. Quiescence is required because the participation lists, source decoders, and routing tables of CPUs and IOHs need to change to reflect the new configuration. While the tables are being changed, the system must not allow traffic to reference these tables. Once the StartReq phase begins, the new information is used for routing stalled and subsequent traffic.
3. IOH write cache flushing and queue draining
Hot add/remove requires flushing of the write cache and draining of queues. IOH provides CSR bits (QPIPWRF) for firmware to initiate cache flushing and queue draining operation. Upon setting of the CSR bit, IOH initiates the associated operation. The completion of the operation clears the corresponding CSR bit. See [Section 4.8](#) for the details of write cache operations.



4. Intel QPI Link Layer Control/Status Registers
IOH defines a set of CSRs associated with Intel QPI link layer (QPI*LS). These CSRs are located in the QPICFG space. The CSRs provide control/status of the link and interrupt generation due to a link change state. The interrupt is programmable to SMI, PMI, SCI, or disabled. Intel QPI physical layer registers (QPI*PHIS) are in DFX space.
5. Intel QPI Link Stall Before Physical Initialization
IOH provides a BMCINIT strap pin that causes the IOH to stall before Intel QPI physical layer initialization. This stall allows the BMC to initialize the IOH (for example, Intel QPI Node ID, Agent Type, Routing Table,..., and so forth) prior to the Intel QPI link parameter exchange phase. Once IOH is setup properly, BMC writes a 1 to the PhyInitBegin bit in the QPI[1:0]PH_CTR register to start Intel QPI physical layer training and proceed to the Link layer initialization phase. In a system without a BMC, IOH can also be configured to complete the Intel QPI PHY and Link initialization rather than being stalled. In this case, the values for the Intel QPI parameters are derived from the straps and default CSR values. The method of holding the CPU in reset to stall the Intel QPI links can be used instead of using the BMCINIT strap.
6. Scratch Pad and Semaphore Registers
IOH contains 16 scratch pad and semaphore registers in the unprivileged space (SR, CWR, IR, TSR TOR). 16 scratch pad and semaphore register in the privilege space (PSR, PCWR, PIR, PTSR, PTOR). These registers provide means of communication between the CPUs and BMC. Hot add/remove operations are coordinated between the CPUs, and between CPUs and BMC using these registers.

16.10.2 IOH Hot-plug

IOH component can be hot added/removed through the support of the Intel QPI or PCIe hot-plug features. By design, add/remove of an IOH causes either Intel QPI hot-plug event, or PCIe hot-plug event, but never both. Refer to the Intel QPI hot-plug described in this chapter and PCIe hot-plug described in the PCIe specification for more details.

Note: Above prescribes the limitations of IOH hot-plug including the requirement that the legacy IOH cannot be removed from the system. For any non-legacy IOHs which connect to an additional ICH, there is the further requirement that the IOH and ICH are both on the same FRU and are powered off together. ESI hot-plug is not supported.

16.10.3 SMBus and Memory Hot-plug

Both IOH and CPU have SMBus interfaces. When IOH or CPU is hot added/removed, it creates an SMBus hot-plug event in addition to PCIe or Intel QPI hot-plug. The support of SMBus hot-plug is done entirely out-of-band by the BMC. IOH provides no special hardware to support SMBus hot-plug.

CPU provides memory hot-plug feature. IOH provides no special hardware to support for memory hot-plug.

§



Reliability, Availability, Serviceability (RAS)



17 Intel® Trusted Execution Technology

17.1 Introduction

Intel's technology for safer computing, Intel® Trusted Execution Technology (Intel® TXT), defines platform-level enhancements that provide the building blocks for creating trusted platforms.

Whenever the word trust is used, there must be a definition of who is doing the trusting and what is being trusted. This enhanced platform helps to provide the authenticity of the controlling environment such that those wishing to rely on the platform can make an appropriate trust decision. The enhanced platform determines the identity of the controlling environment by accurately measuring the controlling software.

Another aspect of the trust decision is the ability of the platform to resist attempts to change the controlling environment. The enhanced platform will resist attempts by software processes to change the controlling environment or bypass the bounds set by the controlling environment.

Intel TXT is a set of extensions designed to provide a measured and controlled launch of system software that will then establish a protected environment for itself and any additional software that it may execute.

These extensions enhance two areas:

- The launching of the Measured Launched Environment (MLE).
- The protection of the MLE from potential corruption.

The enhanced platform provides these launch and control interfaces using Safer Mode Extensions (SMX). The SMX interface includes the following functions:

- Measured/Verified launch of the MLE.
- Mechanisms to ensure the above measurement is protected and stored in a secure location.
- Protection mechanisms that allow the MLE to control attempts to modify itself.

For more information refer to the *Intel® TXT BIOS Writer's Guide* and *Intel® TXT Measured Launched Environment Developer's Guide*. Also refer to <http://download.intel.com/technology/security/downloads/315168.pdf> for the configuration register settings and recommendations needed to support this feature.

§





18 Intel® Virtualization Technology

18.1 Introduction

Intel® Virtualization Technology (Intel® VT) is the technology that makes a single system appear as multiple independent systems to software. This allows for multiple independent operating systems to be running simultaneously on a single system. The first revision of this technology, Intel Virtualization Technology (Intel VT) for IA-32 Intel® Architecture (Intel VT-x) added hardware support in the processor to improve the virtualization performance and robustness. The second revision of this specification, Intel Virtualization Technology (Intel VT) for Directed I/O (Intel VT-d) adds chipset hardware implementation to improve I/O performance and robustness.

18.2 Intel VT-d

Features Supported

- Support for root entry, context entry and default context
- 48 bit max guest address width and 41/46/51 bit max host address width for non-isoch traffic, in MP profiles
- Support for 4K page sizes only
- Support for register based fault recording only and support for MSI interrupts for faults
 - Support for fault collapsing based on Requester ID, OS-visible Intel ME PCI devices and CB3 DMA
- Support for both leaf and non-leaf caching
- Support for boot protection of default page table
- Support for non-caching of invalid page table entries
- Support for Intel VT-d read prefetching/snarfing, that is, translations within a cacheline are stored in an internal buffer for reuse for subsequent transactions.

18.3 Intel VT-d2 Features

- Support for PCISIG endpoint caching (ATS)
- Support for interrupt remapping
- Support for queue-based invalidation interface
- Support for Intel VT-d read prefetching/snarfing e.e. translations within a cacheline are stored in an internal buffer for reuse for subsequent transactions
- Intel VT-d Features Not Supported
- No support for advance fault reporting
- No support for super pages
- No support for 1 or 2 level page walks for 1, 2, or 3 level walks for non-isoch remap engine



- No support for Intel VT-d translation bypass address range (such usage models need to be resolved with VMM help in setting up the page tables correctly)
- Support for queue-based invalidation interface

18.4 Other Virtualization Features Supported

- Support for FLR for CB DMA per PCIE ECR
- Support for V, B, C, R and U bits in ACS ECN
 - IOH performs error checking for the V and B bits. C, R and U bits have no impact on IOH, that is, they are simply capability bits that IOH advertises and nothing more
- ARI ECN compliant to support IOV devices





19 Signal List

This chapter lists all the logical signals which interface to the IOH. This chapter should not be explicitly used to calculate the pin count for the IOH.

19.1 Conventions

The terms *assertion* and *deassertion* are used extensively when describing signals, to avoid confusion when working with a mix of active-high and active-low signals. The term *assert*, or *assertion*, indicates that the signal is active, independent of whether the active level is represented by a high or low voltage. The term *deassert*, or *deassertion*, indicates that the signal is inactive.

Signal names may or may not have a “_N” appended to them. The “_N” symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When “_N” is not present after the signal name the signal is asserted when at the high voltage level.

When discussing data values used inside the component, the logical value is used; that is, a data value described as “1101b” would appear as “1101b” on an active-high bus, and as “0010b” on an active-low bus. When discussing the assertion of a value on the actual pin, the physical value is used; that is, asserting an active-low signal produces a “0” value on the pin.

Table 19-1 and Table 19-2 list the reference terminology used later for buffer technology types (for example, HCSL, and so on) used and buffering signal types (for example, input, output, and so on) used.

Table 19-1. Buffer Technology Types

| Buffer Type | Description |
|------------------------------|--|
| Intel QuickPath Interconnect | Current-mode 6.4 GT/s forwarded-clock Intel QuickPath Interconnect signaling |
| PCIEX2 | Current-mode 5 GHz PCI Express 2nd-generation signaling |
| PCIEX | Current-mode 2.5 GHz PCI Express 1st-generation signaling |
| HCSL | Current-mode differential reference clock input |
| GPIO (SMBus) | 3.3 V 100 KHz SMBus Open Drain output with Schmidt trigger input |
| CMOS | 1.1 V 200 MHz CMOS totem-pole output with Schmidt trigger input |
| GPIO (JTAG) | 1.1 V 20 MHz CMOS open-drain output with Schmidt trigger input |
| Analog | Typically a voltage reference or specialty power supply |
| DDR | 1.8 V VDDR2 reference |

Table 19-2. Buffer Signal Directions

| Buffer Direction | Description |
|------------------|----------------------------------|
| I | Input pin |
| O | Output pin |
| I/O | Bidirectional (input/output) pin |



Some signals or groups of signals have multiple versions. These signal groups may represent distinct but similar ports or interfaces, or may represent identical copies of the signal used to reduce loading effects. Table 19-3 shows the conventions the IOH uses.

Table 19-3. Signal Naming Conventions

| Convention | Definition |
|---------------------|--|
| SIG{0/1/2}XX | Expands to: SIG0XX, SIG1XX, and SIG2XX |
| SIG[2:0] | Denotes a bus and expands to: SIG[2], SIG[1], and SIG[0]. |
| SIG(0/1/2) | Denotes multiple electrical copies of the same output signal and expands to: SIG2, SIG1, and SIG0. |
| SIG_N or SIG[2:0]_N | Denotes an active low signal or bus. |

19.2 Signal List

In the following tables Signal Group column is used for signals with specific DC characteristics.

Table 19-4. JTAG Signals

| Signal Name | Type | Direction | Signal Group | Description |
|-------------|--------------|-----------|--------------|---|
| TCK | GPIO(JTAG) | I | (u) | JTAG Test Clock: Clock input used to drive Test Access Port (TAP) state machine during test and debugging. Internal pullup |
| TDI | GPIO(JTAG) | I | (u) | JTAG Test Data In: Data input for test mode. Used to serially shift data and instructions into TAP. Internal pullup |
| TDO | GPIO(JTAG) | O | (u) | JTAG Test Data Out: Data: Data output for test mode. Used to serially shift data out of the device. Internal pullup. This pin is not tristated when POWERGOOD is not asserted. Therefore, if tristateing this signal at the platform level is required, use either a discrete component or a PLD device. |
| TMS | GPIO(JTAG) | I | (u) | Test Mode Select: This signal is used to control the state of the TAP controller. |
| TRST_N | GPIO(JTAG) | I | (u) | Test Reset: This signal resets the TAP controller logic. |
| SMBSCl | GPIO (SMBus) | I/O | (t) | SMBus Clock: Provides synchronous operation for the SMBus. |
| SMBSDA | GPIO (SMBus) | I/O | (t) | SMBus Addr/Data: Provides data transfer and arbitration for the SMBus. |



Table 19-5. Intel QuickPath Interconnect Signals

| Signal Name | Type | Direction | Signal Group | Description |
|-----------------------|------------------------------|-----------|--------------|--|
| QPI{0}R{P/N}DAT[19:0] | Intel QuickPath Interconnect | I | (a) | Intel QuickPath Interconnect Data Input (Outbound) |
| QPI{0}R{P/N}CLK | Intel QuickPath Interconnect | I | (d) | Intel QuickPath Interconnect Received Clock (Outbound) |
| QPI{0}T{P/N}DAT[19:0] | Intel QuickPath Interconnect | O | (a) | Intel QuickPath Interconnect Data Output (Inbound) |
| QPI{0}T{P/N}CLK | Intel QuickPath Interconnect | O | (d) | Intel QuickPath Interconnect Forwarded Clock (Inbound) |
| QPI{0}{R/I}COMP | Analog | I/O | (c) | Intel QuickPath Interconnect Compensation: Used for the external impedance matching resistors. |
| QPI{0/1}RXBG[1:0] | Analog | I/O | | Intel QuickPath Interconnect external reference voltage signal. This is back-up mode in case of RX band-gap circuit failure |
| QPI{0/1}TXBG[1:0] | Analog | I/O | | Intel QuickPath Interconnect external reference voltage signal. This is back-up mode in case of TX band-gap circuit failure |
| QPI{0}REFCLK{P/N} | HCSL | I | (d) | Intel QuickPath Interconnect Reference Clock: Differential reference clock pair input. |
| QPIFREQSEL{1/0} | CMOS | I | (e) | Intel QuickPath Interconnect frequency selection: Used for determining the normal Intel QuickPath Interconnect operating frequency. |
| QPI{0/1}VRMVREFRX0 | CMOS | I | | Intel QuickPath Interconnect RX external VRM vref. IOH does not need this pin to be driven when QPI{0/1}VRMVREF is being internally generated. Leave pin as No Connect. |
| QPI{0/1}VRMVREFRX1 | CMOS | I | | Intel QuickPath Interconnect RX from external VRM vref. IOH does not need this pin to be driven when QPI{0/1}VRMVREF is being internally generated. Leave pin as No Connect. |
| QPI{0/1}VRMVREFRX2 | CMOS | I | | Intel QuickPath Interconnect RX from external VRM vref. IOH does not need this pin to be driven when QPI{0/1}VRMVREF is being internally generated. |
| QPI{0/1}VRMVREFRX3 | CMOS | I | | Intel QuickPath Interconnect RX from external VRM vref. IOH does not need this pin to be driven when QPI{0/1}VRMVREF is being internally generated. |
| QPI{0/1}VRMVREFTX | CMOS | I | | Intel QuickPath Interconnect TX from external VRM vref. IOH does not need this pin to be driven when QPI{0/1}VRMVREF is being internally generated. |

Table 19-6. PCI Express Signals (Sheet 1 of 2)

| Signal Name | Type | Direction | Signal Group | Description |
|----------------|--------|-----------|--------------|---------------------------------|
| PE1T{P/N}[1:0] | PCIEX2 | O | (g) | PCI Express outbound data port1 |
| PE1R{P/N}[1:0] | PCIEX2 | I | (f) | PCI Express inbound data port1 |
| PE2T{P/N}[1:0] | PCIEX2 | O | (g) | PCI Express outbound data port2 |
| PE2R{P/N}[1:0] | PCIEX2 | I | (f) | PCI Express inbound data port2 |



Table 19-6. PCI Express Signals (Sheet 2 of 2)

| Signal Name | Type | Direction | Signal Group | Description |
|-------------------------------|--------------|-----------|--------------|---|
| PE3T{P/N}[3:0] | PCIEX2 | O | (g) | PCI Express outbound data port3 |
| PE3R{P/N}[3:0] | PCIEX2 | I | (f) | PCI Express inbound data port3 |
| PE4T{P/N}[3:0] | PCIEX2 | O | (g) | PCI Express outbound data port4 |
| PE4R{P/N}[3:0] | PCIEX2 | I | (f) | PCI Express inbound data port4 |
| PE5T{P/N}[3:0] | PCIEX2 | O | (g) | PCI Express outbound data port5 |
| PE5R{P/N}[3:0] | PCIEX2 | I | (f) | PCI Express inbound data port5 |
| PE6T{P/N}[3:0] | PCIEX2 | O | (g) | PCI Express outbound data port6 |
| PE6R{P/N}[3:0] | PCIEX2 | I | (f) | PCI Express inbound data port6 |
| PE7T{P/N}[3:0] | PCIEX2 | O | (g) | PCI Express outbound data port7 |
| PE7R{P/N}[3:0] | PCIEX2 | I | (f) | PCI Express inbound data port7 |
| PE8T{P/N}[3:0] | PCIEX2 | O | (g) | PCI Express outbound data port8 |
| PE8R{P/N}[3:0] | PCIEX2 | I | (f) | PCI Express inbound data port8 |
| PE9T{P/N}[3:0] | PCIEX2 | O | (g) | PCI Express outbound data port9 |
| PE9R{P/N}[3:0] | PCIEX2 | I | (f) | PCI Express inbound data port9 |
| PE10T{P/N}[3:0] | PCIEX2 | O | (g) | PCI Express outbound data port10 |
| PE10R{P/N}[3:0] | PCIEX2 | I | (f) | PCI Express inbound data port10 |
| PE{0/1}CLK{P/N} | HCSSL | I | (j) | PCI Express Reference Clock: Differential reference clock pair input. |
| PE{0/1}JCLK{P/N} | HCSSL | I | | PCI Express PLL jitter injection clock. Recommend to leave these pins floating on external platforms. |
| PE{0/1}{RCOMPO/ICOMPO/ICOMPI} | ANALOG | I/O | (h) | PCI Express Compensation: Used for the external impedance matching resistors. |
| PE{0/1}RBIAS | ANALOG | I/O | (h) | PCI Express clock: External resistance to generate 500uA absolute reference current bias. |
| PEHPSCL | GPIO (SMBus) | O | | PCI Express Hot-Plug SMBus Clock: Provides PCI Express Hot-Plug via dedicated SMBus. |
| PEHPSDA | GPIO (SMBus) | I/O | | PCI Express Hot-Plug SMBus Data: Provides PCI Express Hot-Plug via dedicated SMBus. |

Table 19-7. Signals

| Signal Name | Type | Direction | Signal Group | Description |
|-------------|-------|-----------|--------------|---------------|
| R{P/N}[3:0] | PCIEX | I | (k) | Inbound Data |
| T{P/N}[3:0] | PCIEX | O | (l) | Outbound Data |

Table 19-8. MISC Signals (Sheet 1 of 3)

| Signal | Type | Direction | Signal Group | Description |
|-------------|------|-----------|--------------|--|
| XDPDQ[15:0] | DDR | I/O | (y) | XDP data bus |
| XDPCLK1XP | DDR | O | (y) | XDP clock. Clock 1x reference for XDP |
| XDPCLK1XN | DDR | O | (y) | XDP clock. Clock 1x reference for XDP complement |
| XDPRDYACK_N | DDR | O | (y) | XDP TO XDP Ready Acknowledge |
| XDPRDYREQ_N | DDR | I | (y) | XDP Ready Acknowledge |



Table 19-8. MISC Signals (Sheet 2 of 3)

| Signal | Type | Direction | Signal Group | Description |
|------------------|----------------|-----------|--------------|--|
| XDPDQS{P/N}[1:0] | DDR | I/O | (y) | XDP Strobe |
| VCCXDP18 | Analog | PWR | | XDP Power |
| VCCXDP | Analog | PWR | | XDP on-die termination |
| A20M_N | GPIO1.1 (CMOS) | I | (r) | A20M: Legacy signal from ICH. Translated to Intel QuickPath Interconnect message to CPU: "MASK ADDRESS BIT 20" |
| BMCINIT | GPIO1.1 (CMOS) | I | (r) | BMC initialized: Intel QuickPath Interconnect ports stall indefinitely on power-up physical initialization until an external agent (BMC) releases them. Strap |
| CORERST_N | GPIO3.3 (CMOS) | I | (q) | Reset input: Reset input driven by the system. |
| COREPWRGOOD | GPIO3.3 (CMOS) | I | (q) | Core power good: Clears the IOH. This signal is held low until all power supplies and reference clocks are in specification. This signal is followed by CORERST_N de-assertion |
| AUXPWRGOOD | GPIO3.3 (CMOS) | I | (q) | Auxiliary power good: Clears the IOH. This signal is held low until all power supplies and reference clocks are in specification. This signal is followed by CORERST_N de-assertion. |
| ESILEGACY_N | CMOS | I | | Legacy Port: Indicates that ESI is connected to the legacy ICH. |
| PLLWRDET | GPIO3.3 (CMOS) | I | (q) | Auxiliary PLL power detect: Power and master clocks are stable so PLL's can commence lock. Asserted prior to AUXPWRGOOD. |
| COREPLLWRDET | GPIO3.3 (CMOS) | I | (q) | Core PLL power detect: Power and master clocks are stable so PLL's can commence lock. Asserted prior to COREPWRGOOD. |
| DDRFREQ[3:2] | GPIO1.1 (CMOS) | I | (r) | DDRFREQ[3:2] as DDR frequency selection defined as: "00" = 133MHz input "01" = 100 MHz input "10" = RSVD "11" = RSVD |
| ERR_N[2:0] | Smbus | O | (s) | Error output signals. Minimum assertion time is 12 cycles. |
| EXTSYSTRIG | GPIO | I/O | (u) | External System Trigger. Primary input to the on-die debug trigger mechanism. |
| FERR_N | GPIO | O | (r) | FERR: Legacy signal to ICH. Translated into Intel QuickPath Interconnect message to CPU: "FLOATING POINT ERROR" |
| INIT_N | GPIO | I | (r) | INIT: Legacy signal from ICH. Translated into Intel QuickPath Interconnect message to CPU: "INTERRUPT TO RESET VECTOR" |
| INTR | GPIO | I | (r) | INTR: Legacy signal from ICH. Translated into Intel QuickPath Interconnect message to CPU: "INTERRUPT" |
| LEGACYIOH | GPIO1.1 (CMOS) | I/O | (r) | Used to determine legacy or non-legacy selection: '1': Legacy IOH '0': Non-legacy IOH |
| LTRESET_N | GPIO | O | (s) | LT reset is a mechanism that stops the platform due to a security violation. It is a trigger for a HARD reset. |
| NMI | GPIO1.1 (CMOS) | I | (r) | NMI: Legacy signal from ICH. Translated into Intel QuickPath Interconnect message to CPU: "NON-MASKABLE INTERRUPT" |
| MP/NODEID[3:2] | I CMOS | N/A | | Straps: MP strap. Node ID[3:2] Indicates QPI NodeID bits [3:2]. Indicates SMBus ID bits [2:1]. SMBus ID[3] will always be 0. |



Table 19-8. MISC Signals (Sheet 3 of 3)

| Signal | Type | Direction | Signal Group | Description |
|-----------------------------|-------------------|-----------|--------------|---|
| PESBLCSEL | GPIO | I | (i) | PESBLCSEL: 0 = PCIE LC PLL (default); 1 = PCIE SB PLL (backup) |
| PEWIDTH[5:0] | GPIO1.1 (CMOS) | I/O | (i) | PCIe Link Width Select. |
| QPICKFAIL | I CMOS | N/A | (e) | Intel QPI Clock Failover Mode: Sets Intel QPI physical clock failover support. '0' - Disable Clock Failover '1' - Enable Clock Failover Needs to be disabled for compatibility with components that do not support the same clock failover bits on the Intel QPI interface. |
| QPIFREQSEL[1:0] | CMOS | I | (e) | Intel QuickPath Interconnect frequency selection: Used for determining the normal Intel QuickPath Interconnect operating frequency. "QPIFREQSEL1 & QPIFREQSEL0" decoded as follows: "00" = 4.8 GT/s "01" = 5.867 GT/s (Intel Xeon processor 7500 series based only) "10" = 6.4 GT/s "11" = RSVD |
| QPISBLCSEL | CMOS | I | (e) | QPISBLCSEL: 0 = QPI LC PLL (default); 1 = QPI SB PLL (backup) |
| RESETO_N | JTAG/GPIO1.1 (OD) | O | (u) | RESETO_N: Reset signal to the CPU synchronized to QPICK |
| SMBUSID | GPIO | I | (r) | SMBus ID: Indicates SMBus ID bits [7:4]. '1' indicates an upper-address ID of 1110 (0xE). '0' indicates an upper-address ID of 1100 (0xC). Strap |
| SMI_N | GPIO1.1 (CMOS) | I | (r) | SMI: Legacy signal from ICH. Translated into Intel QuickPath Interconnect message to CPU: "SYSTEM MANAGEMENT INTERRUPT" |
| TEST[4:0] | Analog | I/O | | See the <i>Intel® Xeon® Processor 7500 Series-Based Platform Design Guide</i> for default settings. |
| TESTLO[26-21]; TESTLO[19-1] | GPIO | I | | See the <i>Intel® Xeon® Processor 7500 Series-Based Platform Design Guide</i> for default settings |
| TESTHI[3:1] | GPIO | I | | See the <i>Intel® Xeon® Processor 7500 Series-Based Platform Design Guide</i> for default settings |
| VRMEN | GPIO1.1 (CMOS) | I | (r) | Voltage regulator module enable '0': QPI PLL uses on-die voltage regulator '1': QPI PLL uses LC-filtered power supplied to the socket |
| THERMALERT_N | GPIO3.3 (OD) | O | (s) | The THERMALERT_N (Therm Alert) will go active when the IOH temperature monitoring sensor detected that the IOH has reached its throttle threshold high. |
| THERMTRIP_N | GPIO3.3 (OD) | I/O | (s) | Assertion of THERMTRIP_N (Thermal Trip) indicates the IOH junction temperature has reached a level beyond which permanent silicon damage may occur. |
| TSIREF | Analog | I | | Thermal sensor current reference connected to external resistor of 2.5 KOhm to GND. |

Table 19-9. Controller Link Signals

| Signal Name | Type | Direction | Signal Group | Description |
|-------------|------|-----------|--------------|----------------------------|
| CLCLK | CMOS | I/O | (m) | Clink bi-directional clock |
| CLDATA | CMOS | I/O | (m) | Clink bi-directional data |
| CLRST_N | CMOS | I | (m) | Active low Clink reset |



Table 19-10. RMI Signals

| Signal Name | Type | Direction | Signal Group | Description |
|----------------|-------------|-----------|--------------|------------------------------------|
| RMII_TXD[1:0] | RMII (GPIO) | O | (y) | Transmit data |
| RMII_RXD[1:0] | RMII (GPIO) | I | (y) | Receive data |
| RMII_TXEN | RMII (GPIO) | O | (y) | Transmit enable |
| RMII_CRSDV | RMII (GPIO) | I | (y) | Carrier sense/receive data valid |
| RMII_CLK | RMII (GPIO) | I | (y) | Reference clock |
| RMII_MDIO | RMII (GPIO) | I/O | (y) | Data signal for PHY management bus |
| RMII_MDC | RMII (GPIO) | O | (y) | Clock for PHY management bus |
| RMII_CLKREFOUT | RMII (GPIO) | O | (y) | 50 MHz clock reference output |

Table 19-11. Power and Ground (Sheet 1 of 2)

| Signal Name | Voltage | Description |
|---------------------|---------|--|
| VCCAQPI{0}TX | 1.1V | Intel QuickPath Interconnect analog power supply |
| VCCAQPI{0}PLL | 1.1V | Intel QuickPath Interconnect analog supply voltage for PLL core |
| VCCAQPI{0}RX | 1.1V | Intel QuickPath Interconnect analog power supply |
| VCCQPI{0/1}VRMTXOP0 | 1.1V | Intel QuickPath Interconnect TX VRM power. When VRMEN pin is set to 0, VCCQPIxVRMRXOPx pins are outputs; when set to 1, VCCQPIxVRMRXOPx are inputs. In normal mode, these are debug/observation signals to measure output of the internal VRMs. In bypass mode, internal VRMs are disabled and these pins supply external VCC 1.1 for TX PLL. They should not be shorted and connected as a power pin. |
| VCCQPI{0/1}VRMRXOP0 | 1.1V | Intel QuickPath Interconnect RX VRM power. When VRMEN pin is set to 0, VCCQPIxVRMRXOPx pins are outputs; when set to 1, VCCQPIxVRMRXOPx are inputs. In normal mode, these are debug/observation signals to measure output of the internal VRMs. In bypass mode, internal VRMs are disabled and these pins supply external VCC 1.1 for to 4 sets of RX DLLs. They should not be shorted and connected as a power pin. |
| VCCQPI{0/1}VRMRXOP1 | 1.1V | Intel QuickPath Interconnect RX VRM power. When VRMEN pin is set to 0, VCCQPIxVRMRXOPx pins are outputs; when set to 1, VCCQPIxVRMRXOPx are inputs. In normal mode, these are debug/observation signals to measure output of the internal VRMs. In bypass mode, internal VRMs are disabled and these pins supply external VCC 1.1 for to 4 sets of RX DLLs. They should not be shorted and connected as a power pin. |
| VCCQPI{0/1}VRMRXOP2 | 1.1V | Intel QuickPath Interconnect RX VRM power. When VRMEN pin is set to 0, VCCQPIxVRMRXOPx pins are outputs; when set to 1, VCCQPIxVRMRXOPx are inputs. In normal mode, these are debug/observation signals to measure output of the internal VRMs. In bypass mode, internal VRMs are disabled and these pins supply external VCC 1.1 for to 4 sets of RX DLLs. They should not be shorted and connected as a power pin. |
| VCCQPI{0/1}VRMRXOP3 | 1.1V | Intel QuickPath Interconnect RX VRM power. When VRMEN pin is set to 0, VCCQPIxVRMRXOPx pins are outputs; when set to 1, VCCQPIxVRMRXOPx are inputs. In normal mode, these are debug/observation signals to measure output of the internal VRMs. In bypass mode, internal VRMs are disabled and these pins supply external VCC 1.1 for to 4 sets of RX DLLs. They should not be shorted and connected as a power pin. |
| VCCAQPI{0/1}RXBG | 1.1V | Intel QuickPath Interconnect analog power supply used exclusively by RX band-gap block. |
| VCCQPI{0/1}VRMRX0 | 1.8V | Intel QuickPath Interconnect external power, connect to 1.8 V VRM power bump |
| VCCQPI{0/1}VRMRX1 | 1.8V | Intel QuickPath Interconnect external power, connect to 1.8 V VRM power bump |
| VCCQPI{0/1}VRMRX2 | 1.8V | Intel QuickPath Interconnect external power, connect to 1.8 V VRM power bump |
| VCCQPI{0/1}VRMRX3 | 1.8V | Intel QuickPath Interconnect external power, connect to 1.8 V VRM power bump |
| VCCAPE | 1.1V | VCC for PCI Express analog circuits |



Table 19-11. Power and Ground (Sheet 2 of 2)

| Signal Name | Voltage | Description |
|--------------|---------|--|
| VCCAPEBG | 1.5V | Analog VCC for PCI Express band gap circuit |
| VCCAPE1BG | 1.5V | Analog VCC for PCI Express band gap circuit |
| VCCAPEPLL | 1.1V | Analog VCC for PCI Express PLL analog core |
| VCCPEVRM | 1.5V | PCI Express VRM power supply |
| VCCAPE1PLL | 1.1V | Analog VCC for PCI Express PLL analog core. Note that the power to this pin is only used when the Internal PCIe VRM is not being used. |
| VCCDPE1PLL | 1.1V | PCI Express PLL digital power supply |
| VCCPE1VRM | 1.5V | PCI Express VRM power supply. Note that the power to this pin is only used when the Internal PCIe VRM is not being used. |
| VCCDDR18 | 1.8 | 1.8 V FOR DDR2 |
| VTTDDR | -- | See the Platform Design Guide for connection information |
| VCCDDR18 | 1.8V | 1.8 V DDR I/O supply |
| VTTDDR | 0.9V | 0.9 V Termination Power for DDR IO |
| VCCXDP18 | 1.8V | XDP I/O voltage |
| VTTXDP | 0.9V | 1/2 of XDP |
| VCCCLPWRP | 1.1V | CLINK I/O power |
| VCCEPW | 1.1V | 1.1 V AUX domain |
| VCCMISC33 | 3.3V | GPIO 3.3 V power |
| VCCMISC33EPW | 3.3V | 3.3 V AUX domain |
| VCCTS | 1.5V | Thermal sensor high voltage power supply 1.5 V \pm 5% |
| VSS | 0V | Ground |



19.3 PCI Express Width Strapping

Table 19-12. PEWIDTH[5:0] Strapping Options

| PEWIDTH[5:0] | IOU2 | Port1 | Port2 | IOU0 | Port3 | Port4 | Port5 | Port6 | IOU1 | Port7 | Port8 | Port9 | Port10 |
|--------------|--------------|-------------|-------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 0 | x2 | x2 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 |
| 1 | x2 | x2 | x4 | x4 | x4 | x4 | x4 | x4 | x8 | Not present | x4 | x4 | x4 |
| 10 | x2 | x2 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x8 | Not present | x4 |
| 11 | x2 | x2 | x4 | x4 | x4 | x4 | x4 | x4 | x8 | Not present | x8 | Not present | Not present |
| 100 | x2 | x2 | x8 | Not present | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 |
| 101 | x2 | x2 | x8 | Not present | x4 | x4 | x4 | x4 | x8 | Not present | x4 | x4 | x4 |
| 110 | x2 | x2 | x8 | Not present | x4 | x4 | x4 | x4 | x4 | x4 | x8 | Not present | Not present |
| 111 | x2 | x2 | x8 | Not present | x4 | x4 | x4 | x8 | Not present | x8 | Not present | x8 | Not present |
| 1000 | x2 | x2 | x4 | x4 | x8 | Not present | x4 | x4 | x4 | x4 | x4 | x4 | x4 |
| 1001 | x2 | x2 | x4 | x4 | x8 | Not present | x8 | Not present | x8 | Not present | x4 | x4 | x4 |
| 1010 | x2 | x2 | x4 | x4 | x8 | Not present | x8 | Not present | x4 | x4 | x8 | Not present | Not present |
| 1011 | x2 | x2 | x4 | x4 | x8 | Not present | x8 | Not present | x8 | Not present | x8 | Not present | Not present |
| 1100 | x2 | x2 | x8 | Not present | x8 | Not present | x4 | x4 | x4 | x4 | x4 | x4 | x4 |
| 1101 | x2 | x2 | x8 | Not present | x8 | Not present | x8 | Not present | x8 | Not present | x4 | x4 | x4 |
| 1110 | x2 | x2 | x8 | Not present | x8 | Not present | x4 | x4 | x8 | Not present | x8 | Not present | Not present |
| 1111 | x2 | x2 | x8 | Not present | x8 | Not present | x8 | Not present | x8 | Not present | x8 | Not present | Not present |
| 10000 | x2 | x2 | x16 | Not present | Not present | Not present | Not present | x4 | x4 | x4 | x4 | x4 | x4 |
| 10001 | x2 | x2 | x16 | Not present | Not present | Not present | Not present | x8 | Not present | x8 | Not present | x4 | x4 |
| 10010 | x2 | x2 | x16 | Not present | Not present | Not present | Not present | x4 | x4 | x8 | Not present | x8 | Not present |
| 10011 | x2 | x2 | x16 | Not present | Not present | Not present | Not present | x8 | Not present | x8 | Not present | Not present | Not present |
| 10100 | x2 | x2 | x4 | x4 | x4 | x4 | x4 | x16 | Not present | Not present | Not present | Not present | Not present |
| 10101 | x2 | x2 | x8 | Not present | x4 | x4 | x4 | x16 | Not present | Not present | Not present | Not present | Not present |
| 10110 | x2 | x2 | x4 | x4 | x8 | Not present | x16 | Not present | Not present | Not present | Not present | Not present | Not present |
| 10111 | x2 | x2 | x8 | Not present | x8 | Not present | x16 | Not present | Not present | Not present | Not present | Not present | Not present |
| 11000 | x2 | x2 | x16 | Not present | Not present | Not present | Not present | x16 | Not present | Not present | Not present | Not present | Not present |
| 11001 | x2 | x2 | x16 | Not present | Not present | Not present | Not present | x16 | Not present | Not present | Not present | Not present | Not present |
| 11010 | x2 | x2 | x16 | Not present | Not present | Not present | Not present | x16 | Not present | Not present | Not present | Not present | Not present |
| 11011 | x2 | x2 | x16 | Not present | Not present | Not present | Not present | x16 | Not present | Not present | Not present | Not present | Not present |
| 11100 | Wait-on-BIOS | | | | | | | | | | | | |
| 11101 | Wait-on-BIOS | | | | | | | | | | | | |
| 11110 | Wait-on-BIOS | | | | | | | | | | | | |
| 11111 | Wait-on-BIOS | | | | | | | | | | | | |
| 100000 | x4 | Not present | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 |
| 100001 | x4 | Not present | x4 | x4 | x4 | x4 | x4 | x8 | Not present | x4 | x4 | x4 | x4 |
| 100010 | x4 | Not present | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x8 | Not present | x8 | Not present |
| 100011 | x4 | Not present | x4 | x4 | x4 | x4 | x4 | x8 | Not present | x8 | Not present | Not present | Not present |
| 100100 | x4 | Not present | x8 | Not present | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 | x4 |
| 100101 | x4 | Not present | x8 | Not present | x4 | x4 | x4 | x8 | Not present | x4 | x4 | x4 | x4 |
| 100110 | x4 | Not present | x8 | Not present | x4 | x4 | x4 | x4 | x4 | x4 | x8 | Not present | Not present |
| 100111 | x4 | Not present | x8 | Not present | x4 | x4 | x4 | x8 | Not present | x8 | Not present | Not present | Not present |
| 101000 | x4 | Not present | x4 | x4 | x8 | Not present | Not present | x4 | x4 | x4 | x4 | x4 | x4 |
| 101001 | x4 | Not present | x4 | x4 | x8 | Not present | Not present | x8 | Not present | x4 | x4 | x4 | x4 |
| 101010 | x4 | Not present | x4 | x4 | x8 | Not present | Not present | x4 | x4 | x8 | Not present | Not present | Not present |
| 101011 | x4 | Not present | x4 | x4 | x8 | Not present | Not present | x8 | Not present | x8 | Not present | Not present | Not present |
| 101100 | x4 | Not present | x8 | Not present | x8 | Not present | Not present | x4 | x4 | x4 | x4 | x4 | x4 |
| 101101 | x4 | Not present | x8 | Not present | x8 | Not present | Not present | x8 | Not present | x8 | Not present | x4 | x4 |
| 101110 | x4 | Not present | x8 | Not present | x8 | Not present | Not present | x4 | x4 | x8 | Not present | Not present | Not present |
| 101111 | x4 | Not present | x8 | Not present | x8 | Not present | Not present | x8 | Not present | x8 | Not present | Not present | Not present |
| 110000 | x4 | Not present | x16 | Not present | Not present | Not present | Not present | x4 | x4 | x4 | x4 | x4 | x4 |
| 110001 | x4 | Not present | x16 | Not present | Not present | Not present | Not present | x8 | Not present | x8 | Not present | x4 | x4 |
| 110010 | x4 | Not present | x16 | Not present | Not present | Not present | Not present | x4 | x4 | x8 | Not present | Not present | Not present |
| 110011 | x4 | Not present | x16 | Not present | Not present | Not present | Not present | x8 | Not present | x8 | Not present | Not present | Not present |
| 110100 | x4 | Not present | x4 | x4 | x4 | x4 | x4 | x16 | Not present | Not present | Not present | Not present | Not present |
| 110101 | x4 | Not present | x8 | Not present | x4 | x4 | x4 | x16 | Not present | Not present | Not present | Not present | Not present |
| 110110 | x4 | Not present | x4 | x4 | x8 | Not present | Not present | x16 | Not present | Not present | Not present | Not present | Not present |
| 110111 | x4 | Not present | x8 | Not present | x8 | Not present | Not present | x16 | Not present | Not present | Not present | Not present | Not present |
| 111000 | x4 | Not present | x16 | Not present | Not present | Not present | Not present | x16 | Not present | Not present | Not present | Not present | Not present |
| 111001 | x4 | Not present | x16 | Not present | Not present | Not present | Not present | x16 | Not present | Not present | Not present | Not present | Not present |
| 111010 | x4 | Not present | x16 | Not present | Not present | Not present | Not present | x16 | Not present | Not present | Not present | Not present | Not present |
| 111011 | x4 | Not present | x16 | Not present | Not present | Not present | Not present | x16 | Not present | Not present | Not present | Not present | Not present |



19.4 IOH Signal Strappings

| Pin Name | Location | Connection |
|-----------|----------|--|
| TEST0 | A2 | In Circuit Test: This signal should be connected to a test point on the motherboard. It is internally shorted to the package ground and can be used to determine if the corner ball on the IOH are correctly soldered down to the motherboard. This signal should NOT connect to ground on the motherboard. If TEST0 is not going to be used, it should be left as No Connect. |
| TEST1 | A36 | In Circuit Test: This signal should be connected to a test point on the motherboard. It is internally shorted to the package ground and can be used to determine if the corner ball on the IOH are correctly soldered down to the motherboard. This signal should NOT connect to ground on the motherboard. If TEST1 is not going to be used, it should be left as No Connect. |
| TEST2 | B1 | In Circuit Test: This signal should be connected to a test point on the motherboard. It is internally shorted to the package ground and can be used to determine if the corner ball on the IOH are correctly soldered down to the motherboard. This signal should NOT connect to ground on the motherboard. If TEST2 is not going to be used, it should be left as No Connect. |
| TEST3 | AT1 | In Circuit Test: This signal should be connected to a test point on the motherboard. It is internally shorted to the package ground and can be used to determine if the corner ball on the IOH are correctly soldered down to the motherboard. This signal should NOT connect to ground on the motherboard. If TEST3 is not going to be used, it should be left as No Connect. |
| TEST4 | AT36 | In Circuit Test: This signal should be connected to a test point on the motherboard. It is internally shorted to the package ground and can be used to determine if the corner ball on the IOH are correctly soldered down to the motherboard. This signal should NOT connect to ground on the motherboard. If TEST4 is not going to be used, it should be left as No Connect. |
| TESTHI1 | P29 | If IntelME used: Pull up to P1V1_STBY_IOH via a 10K ohm \pm 5% resistor. If Intel ME not used: Pull up to P1V1_STBY_IOH or P1V1_VCC via a 10K ohm \pm 5% resistor. |
| TESTHI2 | U28 | Connect to debug port XDP. If IntelME used: Pull up to P1V1_STBY_IOH via a 51 ohm \pm 1% resistor. If Intel ME not used: Pull up to P1V1_STBY_IOH or P1V1_VCC via a 51 ohm \pm 1% resistor. |
| TESTHI3 | R29 | If Intel ME used: Pull up to P1V1_STBY_IOH=VCCEPW via a 10K Ohm \pm 5% resistor. If Intel ME not used: Pull up to P1V1_STBY_IOH=VCCEPW or P1V1_VCC via a 10K Ohm \pm 5% resistor. |
| TESTLO1 | AR12 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO2 | AN9 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO3 | AN8 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO4 | AM6 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO5 | AJ34 | Pull down via 1K ohm \pm 1% resistor |
| NODEID2 | AH34 | Pull down via 100 ohm \pm 1% resistor |
| TESTHI[4] | AH33 | Pull high via 10K ohm \pm 1% resistor |
| TESTLO8 | AF35 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO9 | AF34 | Pull down via 0 ohm \pm 5% resistor |
| TESTLO10 | AF32 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO11 | AE34 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO12 | AC32 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO13 | AB30 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO14 | AA29 | Pull down via 100 ohm \pm 1% resistor |
| TESTLO15 | Y28 | Pull down via 1K ohm \pm 1% resistor |

Signal List



| Pin Name | Location | Connection |
|-----------|----------|---|
| TESTLO16 | W27 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO17 | V32 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO18 | T27 | Pull down via 100 ohm \pm 1% resistor |
| TESTLO19 | R35 | Pull down via 1K ohm \pm 1% resistor |
| TESTLO21 | AD33 | Pull down via 10K ohm \pm 1% resistor |
| TESTLO22 | C33 | Pull down via 10K ohm \pm 1% resistor |
| TESTLO23 | AC29 | Pull down via 10K ohm \pm 1% resistor |
| TESTLO24 | AA26 | Pull down via 10K ohm \pm 1% resistor |
| TESETLO26 | D36 | Pull down via 10K ohm \pm 1% resistor |
| XOROUT | AE33 | Pull down via 10K ohm \pm 1% resistor |

§





20 DC Electrical Specifications

In this section, each interface is broken down into groups of signals that have similar characteristics and buffer types.

20.1 DC Characteristics

This section documents the DC characteristics of Intel® 7500 chipset. The specification is split into several sections:

- Clocks
- PCI Express/ESI
- GPIO (CMOS) 1.1v I/O
- GPIO 3.3v (CMOS) I/O
- GPIO 3.3v (OD) I/O
- SMBus Interface
- JTAG Interface
- ME RMI Interface

Table 20-1. Clock DC Characteristics (Sheet 1 of 2)

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|-------------------------|--------------|--------------------------|---|-------|---|------|-------|
| 133 MHz | | | | | | | |
| V _{IL} | (d) | Input Low Voltage | -0.150 | 0 | 0.150 | V | 1 |
| V _{IH} | (d) | Input High Voltage | 0.660 | 0.700 | 0.850 | V | |
| V _{CROSS(abs)} | (d) | Absolute Crossing Point | 0.250 | | 0.550 | V | 2, 7 |
| V _{CROSS(rel)} | (d) | Relative Crossing Point | $0.250 + 0.5 \times (V_{Havg} - 0.700)$ | | $0.550 - 0.5 \times (0.700 - V_{Havg})$ | V | 7, 8 |
| ΔV _{CROSS} | (d) | Range of Crossing Points | | | 0.140 | V | |
| V _{OS} | (d) | Overshoot | | | V _{IH} + 0.300 | V | 3 |
| V _{US} | (d) | Undershoot | -0.300 | | | V | 4 |
| V _{RBM} | (d) | Ringback Margin | 0.200 | | | V | 5 |
| V _{TR} | (d) | Threshold Region | V _{CROSS} - 0.100 | | V _{CROSS} + 0.100 | V | 6 |
| 100 MHz | | | | | | | |
| V _{IL} | (j) | Input Low Voltage | -0.150 | 0 | | V | |
| V _{IH} | (j) | Input High Voltage | 0.660 | 0.700 | 0.850 | V | |
| V _{CROSS(abs)} | (j) | Absolute Crossing Point | 0.250 | | 0.550 | V | 2, 7 |
| V _{CROSS(rel)} | (j) | Relative Crossing Point | $0.250 + 0.5 \times (V_{Havg} - 0.700)$ | | $0.550 + 0.5 \times (V_{Havg} - 0.700)$ | V | 7, 8 |
| ΔV _{CROSS} | (j) | Range of Crossing Points | | | 0.140 | V | 2 |
| V _{OS} | (j) | Overshoot | | | V _{IH} + 0.300 | V | 3 |
| V _{US} | (j) | Undershoot | -0.300 | | | V | 4 |



Table 20-1. Clock DC Characteristics (Sheet 2 of 2)

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|------------------|--------------|------------------|----------------------------|-----|----------------------------|------|-------|
| V _{RBM} | (j) | Ringback Margin | 0.200 | | | V | 5 |
| V _{TR} | (j) | Threshold Region | V _{CROSS} – 0.100 | | V _{CROSS} + 0.100 | V | 6 |

Notes:

1. Refer to [Figure 20-1](#) Differential Clock Crosspoint Specification and [Figure 20-2](#) Differential Clock Waveform.
2. Crossing voltage is defined as the instantaneous voltage when the rising edge of CORECLKP is equal to the falling edge of CORECLKN.
3. Overshoot is defined as the absolute value of the maximum voltage.
4. Undershoot is defined as the absolute value of the minimum voltage.
5. Ringback Margin is defined as the absolute voltage difference between the maximum Rising Edge Ringback and the maximum Falling Edge Ringback. Both maximum Rising and Falling Ringbacks should not cross the threshold region.
6. Threshold Region is defined as a region centered around the crossing point voltage in which the differential receiver switches. It includes input threshold hysteresis.
7. The crossing point must meet the absolute and relative crossing point specifications simultaneously.
8. V_{Havg} (the average of V_{IH}) can be measured directly using “Vtop” on Agilent* scopes and “High” on Tektronix* scopes.

20.2 PCI Express* / ESI Interface DC Characteristics

Table 20-2. PCI Express / ESI Differential Transmitter (Tx) Output DC Characteristics

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|-----------------------------|--------------|--|-----|-----|-----|------|-------|
| VTX-CM-DC-ACTIVE-IDLE-DELTA | (f) (k) | Absolute Delta of DC Common Mode Voltage During L0 and Electrical Idle | 0 | | 100 | mV | 1 |
| VTX-CM-DC-LINE-DELTA | (f) (k) | Absolute Delta of DC Common Mode Voltage between D+ and D- | 0 | | 25 | mV | 1 |
| VTX-IDLE-DIFFp | (f) (k) | Electrical Idle Differential Peak Output Voltage | | | 20 | mV | 1 |
| VTX-RCV-DETECT | (f) (k) | The amount of voltage change allowed during Receiver Detection | | | 600 | mV | |
| VTX-DC-CM | (f) (k) | The TX DC Common Mode Voltage | 0 | | 3.6 | V | 1 |
| ITX-SHORT | (f) (k) | The Short Circuit Current Limit | | | 90 | mA | |
| ZTX-DIFF-DC | (f) (k) | DC Differential TX Impedance | 80 | 100 | 120 | Ω | |
| ZTX-DC | (f) (k) | Transmitter DC Impedance | 40 | | | Ω | |

Notes:

1. Specified at the measurement point into a timing and voltage compliance test load and measured over any 250 consecutive TX UIs.

Table 20-3. PCI Express / ESI Differential Receiver (Rx) Input DC Characteristics

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|--------------------|--------------|---|-----|-----|-----|------|-------|
| ZRX-DIFF-DC | (g) (l) | DC Differential Input Impedance | 80 | 100 | 120 | Ω | 3 |
| ZRX-DC | (g) (l) | DC Input Impedance | 40 | 50 | 60 | Ω | 1, 2 |
| ZRX-High-Imp-DC | (g) (l) | Power Down DC Input Common Mode Impedance | 200 | | | kΩ | 4 |
| VRX-IDLE-DET-DIFFp | (g) (l) | Electrical Idle Detect Threshold | 65 | | 175 | mV | |

**Notes:**

- Specified at the measurement point and measured over any 250 consecutive UIs. If the clock to the RX and TX are not derived from the same reference clock, the TX UI recovered from 3500 consecutive UI must be used as a reference for the eye diagram.
- A TRX-EYE=0.40UI provides for a total sum of 0.60 UI deterministic and random jitter budget for the Transmitter and interconnect collected any 250 consecutive UIs. The TRX-EYE-MEDIAN-to-MAX-JITTER specification ensures a jitter distribution in which the median and the maximum deviation from the median is less than half of the total .6 UI jitter budget collected over any 250 consecutive TX UIs. It should be noted that the median is not the same as the mean. The jitter median describes the point in time where the number of jitter points on either side is approximately equal as opposed to the averaged time value. If the clocks to the RX and TX are not derived from the same reference clock, the TX UI recovered from 3500 consecutive UI must be used as the reference for the eye diagram.
- Impedance during all LTSSM states. When transitioning from a Fundamental Reset to Detect (the initial state of the LTSSM) there is a 5 ms transition time before Receiver termination values must be met on all un-configured Lanes of a Port.
- The RX DC Common Mode Impedance that exists when no power is present or Fundamental Reset is asserted. This helps ensure that the Receiver Detect circuit will not falsely assume a Receiver is powered on when it is not. This term must be measured at 300 mV above the RX ground.

20.3 Miscellaneous DC Characteristics

Table 20-4. CMOS, JTAG, SMBUS, GPIO3.3V, and MISC DC Characteristics (Sheet 1 of 2)

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|-----------------------------------|--------------|---------------------|---------------------------|-----|---------------------------|------|-------|
| GPIO1.1 (CMOS) I/O Signals | | | | | | | |
| V _{OH_CMOS} | (e,i,m,n,r) | Output High Voltage | 0.25*V _{cc} 1.1V | | | V | |
| V _{OL_CMOS} | (e,i,m,n,r) | Output Low Voltage | | | 0.25*V _{cc} 1.1V | V | |
| V _{IH_CMOS} | (e,i,m,n,r) | Input High Voltage | 0.35*V _{cc} 1.1V | | V _{cc} 1.1v+0.2 | V | |
| V _{IL_CMOS} | (e,i,m,n,r) | Input Low Voltage | -0.2 | | 0.35*V _{cc} 1.1v | V | |
| I _{OH_CMOS} | (e,i,m,n,r) | Output High Current | 4 | | 4 | mA | |
| I _{OL_CMOS} | (e,i,m,n,r) | Output Low Current | 4 | | 4 | mA | |
| I _{LEAK_CMOS} | (e,i,m,n,r) | Leakage Current | | | 15 | μA | |
| C _{PAD_CMOS} | (e,i,m,n,r) | Pad Capacitance | | | 7 | pF | |
| GPIO3.3(OD) Signals | | | | | | | |
| V _{OH_GPIO3.3} | (s) | Output High Voltage | N/A | | N/A | V | 1 |
| V _{OL_GPIO3.3} | (s) | Output Low Voltage | | | 0.4 | V | |
| V _{IH_GPIO3.3} | (s) | Input High Voltage | 2.1 | | | V | |
| V _{IL_GPIO3.3} | (s) | Input Low Voltage | | | 0.8 | V | |
| I _{OL_GPIO3.3} | (s) | Output Low Current | | | 4 | mA | |
| I _{LEAK_GPIO3.3} | (s) | Leakage Current | | | 15 | μA | |
| C _{PAD_GPIO3.3} | (s) | Pad Capacitance | | | 10 | pF | |
| SMBUS Signals | | | | | | | |
| V _{OH_SMBUS} | (t) | Output High Voltage | N/A | | | V | 1 |
| V _{OL_SMBUS} | (t) | Output Low Voltage | | | 0.4 | V | |
| V _{IH_SMBUS} | (t) | Input High Voltage | 2.1 | | | V | |
| V _{IL_SMBUS} | (t) | Input Low Voltage | | | 0.8 | V | |
| I _{OL_SMBUS} | (t) | Output Low Current | | | 4 | mA | |
| I _{LEAK_SMBUS} | (t) | Leakage Current | | | 10 | μA | |
| C _{PAD_SMBUS} | (t) | Pad Capacitance | | | 10 | pF | |

Table 20-4. CMOS, JTAG, SMBUS, GPIO3.3V, and MISC DC Characteristics (Sheet 2 of 2)

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|--|--------------|---------------------------|---------------------------|-----|---------------------------|------|-------|
| JTAG and Misc GPIO1.1(OD) Signals | | | | | | | |
| V _{OH_JTAG} | (u) | Output High Voltage | N/A | | N/A | V | 1 |
| V _{OL_JTAG} | (u) | Output Low Voltage | | | 0.25*V _{cc} 1.1V | V | |
| V _{IH_JTAG} | (u) | Input High Voltage | 0.65*V _{cc} 1.1V | | | V | |
| V _{IL_JTAG} | (u) | Input Low Voltage | | | 0.35*V _{cc} 1.1V | V | |
| I _{OL_JTAG} | (u) | Output Low Current | 16 | | | mA | |
| I _{LEAK_JTAG} | (u) | Leakage Current | | | 15 | μA | |
| C _{PAD_JTAG} | (u) | Pad Capacitance | | | 7 | pF | |
| RMI Signals | | | | | | | |
| V _{REF} | (y) | Bus High Reference | 3.0 | 3.3 | 3.6 | V | |
| V _{ABS} | (y) | Signal Voltage Range | -0.3 | | 3.765 | V | |
| V _{IL} | (y) | Input Low Voltage | | | 0.8 | V | |
| V _{IH} | (y) | Input High Voltage | 2 | | | V | |
| V _{OH} | (y) | Output High Voltage | 2.4 | | | V | |
| V _{OL} | (y) | Output Low Voltage | 0 | | 400 | mV | |
| I _{IH} | (y) | Input High Current | 0 | | 200 | μA | |
| I _{IL} | (y) | Input Low Current | -20 | | 0 | μA | |
| I _{LEAK} | (y) | Leakage Current | -20 | | 20 | μA | |
| V _{CKM} | (y) | Clock Midpoint Ref. Level | | | 1.4 | V | |

Notes:

1. N/A for Open Drain pins.

Figure 20-1. Differential Measurement Point for Rise and Fall Time

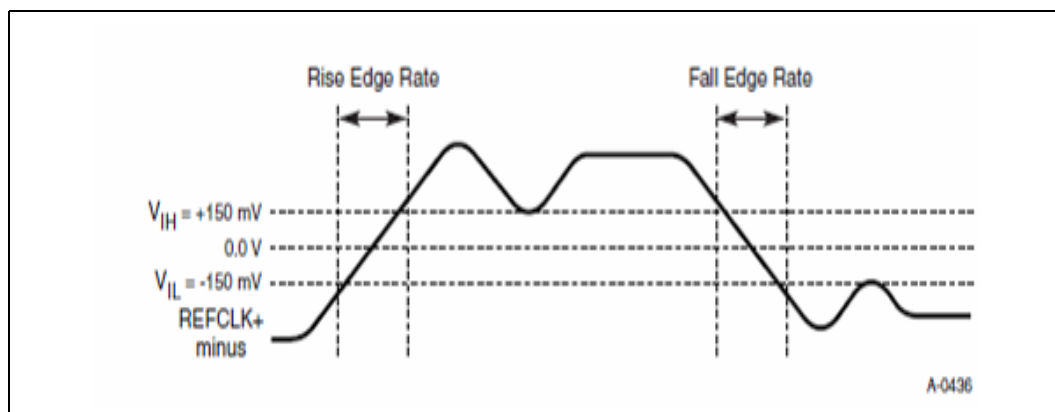
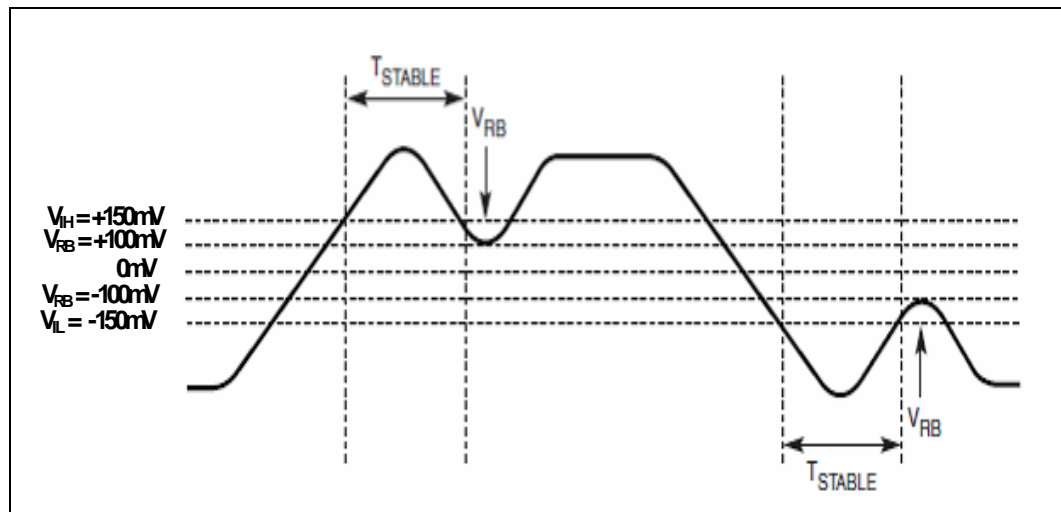




Figure 20-2. Differential Measurement Point for Ringback



§





21 Configuration Register Space

This chapter describes both the PCI configuration space and CSRCFG configuration space registers.

21.1 Device Mapping: Functions Specially Routed by the IOH

All devices on the IOH reside on Bus 0. The following table describes the devices and functions that the IOH implements or routes specially.

Table 21-1. Functions Specially Handled by the IOH

| Register Group | DID | Device | Function | Comment |
|--|----------------------|--------|----------|--|
| ESI (Dev 0 in ESI mode) | 0011_0100_0000_0xxxh | 0 | 0 | The ESI port will have the last 3 bits of the DID (xxx) Intel® 7500 Chipset = 111 |
| PCI Express Root Port 0 (Dev#0 in PCIe mode) | 3420h or 3421h | 0 | 0 | Dev#0 will work as a X4 Gen1 port. DID is depending on LEGACYIOH straps. |
| PCI Express Root Port 1 | 3408h | 1 | 0 | x4 or x2 max link width |
| PCI Express Root Port 2 | 3409h | 2 | 0 | x2 max link width |
| PCI Express Root Port 3 | 340Ah | 3 | 0 | x8, or x4 max link width |
| PCI Express Root Port 4 | 340Bh | 4 | 0 | x4 max link width |
| PCI Express Root Port 5 | 340Ch | 5 | 0 | x8 or x4 max link width |
| PCI Express Root Port 6 | 340Dh | 6 | 0 | x4 max link width |
| PCI Express Root Port 7 | 340Eh | 7 | 0 | x8, or x4 max link width |
| PCI Express Root Port 8 | 340Fh | 8 | 0 | x4 max link width |
| PCI Express Root Port 9 | 3410h | 9 | 0 | x8 or x4 max link width |
| PCI Express Root Port 10 | 3411h | 10 | 0 | x4 max link width |
| Intel QuickPath Interconnect Port 0 | 3425h | 16 | 0 | |
| Intel QuickPath Interconnect Port 0 | 3426h | 16 | 1 | |
| Intel QuickPath Interconnect Port 1 | 3427h | 17 | 0 | |
| Intel QuickPath Interconnect Port 1 | 3428h | 17 | 1 | |
| IOxAPIC | 342Dh | 19 | 0 | |
| Core | 342Eh | 20 | 0 | Address mapping, Intel VT-d, Ctrl/Status, Misc. Registers |
| Core | 3422h | 20 | 1 | Scratchpads and GPIO registers |
| Core | 3423h | 20 | 2 | IOH control/status and RAS registers |
| Core | 3438h | 20 | 3 | Throttling registers |



21.2 Unimplemented Devices/Functions and Registers

Configuration reads to unimplemented functions and devices will return all ones emulating a master abort response. There is no asynchronous error reporting when a configuration read master aborts. Configuration writes to unimplemented functions and devices will return a normal response to Intel QuickPath Interconnect.

Software should not attempt or rely on reads or writes to unimplemented registers or register bits. Unimplemented registers return all zeroes when read. Writes to unimplemented registers are ignored. For configuration writes to these registers, the completion is returned with a normal completion status (not master-aborted).

21.2.1 Register Attribute Definition

The bits in the configuration register descriptions will all be assigned attributes. The following table defines all the attributes types. All bits will be set to their default value by any reset that resets the IOH core, except the Sticky bits. Sticky bits are only reset by the PWRGOOD reset.

Table 21-2. Register Attributes Definitions (Sheet 1 of 2)

| Attr | Description |
|-------|--|
| RO | Read Only: These bits can only be read by software, writes have no effect. The value of the bits is determined by the hardware only. |
| RW | Read / Write: These bits can be read and written by software. |
| RWO | Read / Write Once: These bits can be read by software. After reset, these bits can only be written by software once, after which the bits becomes 'Read Only'. |
| RWL | Read / Write Lock: These bits can be read and written by software. Hardware can make these bits 'Read Only' via a separate configuration bit or other logic. |
| RW1C | Read / Write 1 to Clear: These bits can be read and cleared by software. Writing a '1' to a bit clears it, while writing a '0' to a bit has no effect. |
| RC | Read Clear: These bits can only be read by software, but a read causes the bits to be cleared. <i>Note: Use of this attribute type is deprecated, as reads with side-effects are harmful for debug.</i> |
| RCW | Read Clear / Write: These bits can be read and written by software, but a read causes the bits to be cleared. <i>NOTE: Use of this attribute type is deprecated, as reads with side-effects are harmful for debug.</i> |
| ROS | RO Sticky: These bits can only be read by software, writes have no effect. The value of the bits is determined by the hardware only. These bits are only re-initialized to their default value by a PWRGOOD reset. |
| RWS | R / W Sticky: These bits can be read and written by software. These bits are only re-initialized to their default value by a PWRGOOD reset. |
| RW1CS | R / W1C Sticky: These bits can be read and cleared by software. Writing a '1' to a bit clears it, while writing a '0' to a bit has no effect. These bits are only re-initialized to their default value by a PWRGOOD reset. |
| RV | Reserved: These bits are reserved for future expansion and their value must not be modified by software. When writing these bits, software must preserve the value read. The bits are read-only must return '0' when read. |
| RWD | RW, value written will take effect on the next Link Layer init. |
| RWDS | RW, RW and sticky. Re-initialized to default value only with POWERGOOD reset. Value written will take effect on the next Link layer init. |
| RWDN | Reset to default only after next hard Intel QuickPath Interconnect link layer initialization occurs |
| RWNN | RW, reset to default when soft Intel QuickPath Interconnect link layer initialization occurs |
| RW1CN | RW1C, reset to default when hard Intel QuickPath Interconnect link layer initialization occurs |
| RONN | RO, reset to default when soft Intel QuickPath Interconnect link layer initialization occurs |
| RWP | RW, reset to default when hard Intel QuickPath Interconnect physical layer initialization occurs |



Table 21-2. Register Attributes Definitions (Sheet 2 of 2)

| Attr | Description |
|-----------|--|
| RWDP | RW, reset to default only after next hard Intel QuickPath Interconnect physical layer initialization occurs |
| RWPP | RW, reset to default when soft Intel QuickPath Interconnect physical layer initialization occurs |
| ROPP | RO, reset to default when soft Intel QuickPath Interconnect physical layer initialization occurs |
| RW1CPP | RW1C, reset to default when soft Intel QuickPath Interconnect physical layer initialization occurs |
| Modifiers | These can be appended to the end of base modifiers. Some of the attributes above include the modifiers |
| G | General modifier: This modifier is applicable to register attribute, for example, RWOG. Registers bits with G modifier are not specific to the Function and so are only reinitialized to their default value by a Conventional Reset (not Function Level Reset). |
| S | Sticky. For example RWS means R/W sticky |
| N | Reset to default when hard Intel QuickPath Interconnect link layer initialization occurs |
| NN | Reset to default when soft Intel QuickPath Interconnect link layer initialization occurs |
| P | Reset to default when hard Intel QuickPath Interconnect physical layer initialization occurs |
| PP | Reset to default when soft Intel QuickPath Interconnect physical layer initialization occurs |
| D | Late action on link/phy init. Typically value written will take effect on the next Link/Phy init. |
| DP | Reset to default only after next hard QPI physical layer initialization occurs. This attribute is mainly used by Dfx spec. |
| DS | Re-initialized to default value only with POWERGOOD reset. Value written will take effect on the next Link layer init. |
| 1C | 1 clear: Writing a '1' to a bit clears it, while writing a '0' to a bit has no effect. |
| L | Lock: Hardware can make these bits 'Read Only' via a separate configuration bit or other logic. |



21.3 RID Implementation in IOH

21.3.1 Background

Historically, a new value has been assigned to Revision ID (RID in PCI header space) for every stepping of a chipset. RID provides a way for software to identify a particular component stepping when a driver change or patch unique to that stepping is needed.

Operating systems detect RID during device enumeration to notify the user on the presence of new hardware. This may create problems for OEM when installing OS images on a new stepping of a product that is essentially identical to the previous stepping. In some cases, “New Hardware Found” messages may disrupt end user IT customer software images.

The solution is to implement a mechanism to select one of the two possible values to be read from the RID register. The default power-on value for the RID register will be called the Stepping Revision ID (SRID). When necessary, the BIOS can select a second value, called the Compatible Revision ID (CRID), to be read from the RID register.

Stepping Revision ID (SRID): This is the default power on value for mask/metal steppings.

Compatible Revision ID (CRID): The CRID functionality gives BIOS the flexibility to load OS drivers optimized for a previous revision of the silicon instead of the current revision of the silicon in order to reduce drivers updates and minimize changes to the OS image for minor optimizations to the silicon for yield improvement, or feature enhancement reasons that do not negatively impact the OS driver functionality.

21.3.2 Stepping Revision ID (SRID)

The SRID is a 4-bit hardwired value assigned by Intel, based on product's stepping. The SRID is not a directly addressable PCI register. The SRID value is reflected through the RID register when appropriately addressed. The 4 bits of the SRID are reflected as the two least significant bits of the major and minor revision field respectively.

21.3.3 Conceptual Description

Following reset, the SRID value may be read from the RID register at offset 08h of all devices and functions in the IOH chipset, which reflects the actual product stepping. To select the CRID value, BIOS/configuration software writes a 32-bit key value of 00000069h to Bus 0, Device 0, Function 0 (ESI port) of the IOH's RID register at offset 08h. Through a comparator, the written value is matched with a key value of “00000069h”. The comparator output is flopped and controls the selection of either CRID or RID. Subsequent reads to RID register at offset 08h will return CRID if the comparator flop is set. Otherwise it will always return the SRID when the comparator flop is reset. The internal RID comparator flop in the ESI port (Bus 0 device 0 Function 0) is a “write-once” register and gets locked after the first write to offset 08h.

The RID values for all devices and functions in IOH are changed together by writing the key value (00000069h) to the RID register in Bus 0, Device 0, Function 0. Writing to the RID register of other devices has no effect. A reset will change the RID selection back to SRID. The CRID values are programmed during manufacture to suit the customer needs and the BIOS can set the comparator as described above for software to read the CRID values.



21.4 Standard PCI Configuration Space (0x0 to 0x3F) - Type 0/1 Common Configuration Space

This section covers registers in the 0x0 to 0x3F region that are common to all the devices 0 to 22. Comments at the top of the table indicate what devices/functions the description applies to. Exceptions that apply to specific functions are noted in the individual bit descriptions.

21.4.1 Configuration Register Map

Table 21-3. PCIe Capability Registers for Devices with PCIe Extended Configuration Space

| | | | | | |
|----------------|---------------------|--------------|---------|------|-----|
| DID | VID | 00h | PEXCAPH | 100h | |
| PCISTS | PCICMD | 04h | | | |
| CCR | RID | 08h | | | |
| HDR | CLS | 0Ch | | | |
| | | 10h | | | |
| | | 14h | | | |
| | | 18h | | | |
| | | 1Ch | | | |
| | | 20h | | | |
| | | 24h | | | |
| | | 28h | | | |
| SID | SVID | 2Ch | | | |
| | | 30h | | | |
| | CAPPTR ¹ | 34h | | | |
| | | 38h | | | |
| | INTP | INTL | | | 3Ch |
| <i>EXPCAP</i> | <i>NXTPTR</i> | <i>CAPID</i> | | | 40h |
| <i>DEVCAP</i> | | | | | 44h |
| <i>DEVSTS</i> | <i>DEVCON</i> | | | | 48h |
| <i>LNKCAP</i> | | | | | 4Ch |
| <i>LNKSTS</i> | <i>LNKCON</i> | | | | 50h |
| <i>SLTCAP</i> | | | | | 54h |
| <i>SLTSTS</i> | <i>SLTCON</i> | | | | 58h |
| <i>ROOTCAP</i> | <i>ROOTCON</i> | | | | 5Ch |
| <i>ROOTSTS</i> | | | | | 60h |
| <i>DEVCAP2</i> | | | | | 64h |
| <i>DEVSTS2</i> | <i>DEVCON2</i> | | | | 68h |
| <i>LNKCAP2</i> | | | | | 6Ch |
| <i>LNKSTS2</i> | <i>LNKCON2</i> | | 70h | | |
| <i>SLTCAP2</i> | | | 74h | | |
| <i>SLTSTS2</i> | <i>SLTCON2</i> | | 78h | | |
| | | 7Ch | | | |

Notes:

1. CAPPTR points to the first capability block which is at 0x40h
Italics indicates register only present in devices/functions with extended configuration space.
 For the PCI Express port registers, please refer to the PCI Express register section.



21.4.2 Register Definitions - Common

This section describes the common header registers that are present in all PCI Express devices. It covers registers from offset 0x0 to 0x3F. Note that the PCI Express ports and DMA registers are being defined in their own sections and should be used instead of this section.

21.4.2.1 VID: Vendor Identification Register

The Vendor Identification Register contains the Intel identification number.

| Device: 16, 17 Function: 0, 1 | | | |
|----------------------------------|------|---------|---|
| Device: 20 Function: 0-3 | | | |
| Offset: 00h | | | |
| Bit | Attr | Default | Description |
| 15:0 | RO | 8086h | Vendor Identification Number The value is assigned by PCI-SIG to Intel. |

21.4.2.2 DID: Device Identification Register

Device ID register with IOH-specific device IDs.

| Device: 16, 17 Function: 0, 1 | | | |
|----------------------------------|------|--------------------------------|---|
| Device: 20 Function: 0-3 | | | |
| Offset: 02h | | | |
| Bit | Attr | Default | Description |
| 15:0 | RO | See Table 21-1 | Device Identification Number The value is assigned by Intel to each product. IOH will have a unique device id for each of its single function devices and a unique device id for each function in the multi-function devices. IOH will also have a unique Device ID for Device#0. |



21.4.2.3 PCICMD: PCI Command Register

This register defines the PCI 3.0 compatible command register values applicable to PCI Express space.

| Device: 16, 17 Function: 0, 1 Device: 20 Function: 0-3 Offset: 04h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:11 | RV | 0 | Reserved (by PCI SIG) |
| 10 | RO | 0 | Interrupt Disable Controls the ability of DMA to generate legacy INTx interrupt (when legacy INTx mode is enabled). This bit does not affect the ability of the Express port to route interrupt messages received at the PCI Express port. 1: Legacy Interrupt message generation is disabled 0: Legacy Interrupt message generation is enabled If this bit transitions from 1->0 when a previous Assert_INTx message was sent but no corresponding Deassert_INTx message sent yet, a Deassert_INTx message is sent on this bit transition. |
| 9 | RO | 0 | Fast Back-to-Back Enable Not applicable to PCI Express and is hardwired to 0 |
| 8 | RO | 0 | SERR Enable For PCI Express/ESI ports, this field enables notifying the internal core error logic of occurrence of an uncorrectable error (fatal or non-fatal) at the port. The internal core error logic of IOH then decides if/how to escalate the error further (pins/message and so on). This bit also controls the propagation of PCI Express ERR_FATAL and ERR_NONFATAL messages received from the port to the internal IOH core error logic. 1: Fatal and Non-fatal error generation and Fatal and Non-fatal error message forwarding is enabled 0: Fatal and Non-fatal error generation and Fatal and Non-fatal error message forwarding is disabled Refer to <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s</i> for details of how this bit is used in conjunction with other control bits in the Root Control register for forwarding errors detected on the PCI Express interface to the system core error logic. This bit has no impact on error reporting from the other devices - DMA, I/OxAPIC registers. |
| 7 | RO | 0 | IDSEL Stepping/Wait Cycle Control Not applicable to internal IOH devices. Hardwired to 0. |
| 6 | RO | 0 | Parity Error Response For PCI Express/ESI ports, IOH ignores this bit and always does ECC/parity checking and signaling for data/address of transactions both to and from IOH. This bit though affects the setting of bit 8 in the PCISTS (see bit 8 in Section 21.4.2.4) register. This bit has no impact on error reporting from the other devices - DMA, I/OxAPIC registers. |
| 5 | RO | 0 | VGA palette snoop Enable Not applicable to internal IOH devices. Hardwired to 0. |
| 4 | RO | 0 | Memory Write and Invalidate Enable Not applicable to internal IOH devices. Hardwired to 0. |
| 3 | RO | 0 | Special Cycle Enable Not applicable to PCI Express. Hardwired to 0. |



| Device: 16, 17 Function: 0, 1 Device: 20 Function: 0-3 Offset: 04h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 2 | RO | 0 | Bus Master Enable Controls the ability of the PCI Express/ESI port in generating/forwarding memory (including MSI writes) or I/O transactions (and not messages) or configuration transactions from the secondary side to the primary side. For DMA and I/OxAPIC, this bit enables them to generate memory write/MSI and memory read transactions (read applies only to DMA). 1: Enables the PCI Express/ESI port, I/OxAPIC or DMA to generate/forward memory, config or I/O read/write requests. 0: The Bus Master is disabled. When this bit is 0, IOH root ports will treat upstream PCI Express memory writes/reads, IO writes/reads, and configuration reads and writes as unsupported requests (and follow the rules for handling unsupported requests). This behavior is also true towards transactions that are already pending in the IOH root port's internal queues when the BME bit is turned off. I/OxAPIC and DMA cannot generate any memory transactions when this bit is 0. |
| 1 | RO | 0 | Memory Space Enable 1: Enables a PCI Express/ESI port's memory range registers, internal I/OxAPIC's MBAR register (ABAR range decode is not enabled by this bit) or DMA device's memory BARs to be decoded as valid target addresses for transactions from primary side. 0: Disables a PCI Express/ESI port's memory range registers (excluding the IOxAPIC range registers), internal I/OxAPIC's MBAR register (but not ABAR register) or DMA device's memory BARs to be decoded as valid target addresses for transactions from primary side. Note that if a PCI Express/ESI port's MSE bit is clear, that port can still be target of any memory transaction if subtractive decoding is enabled on that port. |
| 0 | RO | 0 | IO Space Enable Applies only to PCI Express/ESI ports 1: Enables the I/O address range, defined in the IOBASE and IOLIM registers of the PCI-to-PCI bridge header, for target decode from primary side 0: Disables the I/O address range, defined in the IOBASE and IOLIM registers of the PCI-to-PCI bridge header, for target decode from primary side Note that if a PCI Express/ESI port's IOSE bit is clear, that port can still be target of an I/O transaction if subtractive decoding is enabled on that port. |



21.4.2.4 PCISTS: PCI Status Register

The PCI Status register is a 16-bit status register that reports the occurrence of various events associated with the primary side of the “virtual” PCI-PCI bridge embedded in PCI Express ports and also primary side of the other devices on the internal IOH bus.

| Device: 19 Function: 0 Device: 16, 17 Function: 0, 1 Device: 20 Function: 0-3 Offset: 06h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15 | RW1C | 0 | Detected Parity Error This bit is set by a device when it receives a packet on the primary side with an uncorrectable data error (that is, a packet with poison bit set or an uncorrectable data ECC error was detected at the XP-DP interface when ECC checking is done) or an uncorrectable address/control parity error. The setting of this bit is regardless of the Parity Error Response bit (PERRE) in the PCICMD register. |
| 14 | RO | 0 | Signaled System Error 1: The device reported fatal/non-fatal (and not correctable) errors it detected on its PCI Express interface. Software clears this bit by writing a '1' to it. For Express ports, this bit is also set (when SERR enable bit is set) when a FATAL/NON-FATAL message is forwarded from the Express link. Note that IOH internal 'core' errors (like parity error in the internal queues) are not reported via this bit. 0: The device did not report a fatal/non-fatal error |
| 13 | RO | 0 | Received Master Abort This bit is set when a device experiences a master abort condition on a transaction it mastered on the primary interface (IOH internal bus). Note that certain errors might be detected right at the PCI Express interface and those transactions might not 'propagate' to the primary interface before the error is detected (for example, accesses to memory above TOCM in cases where the PCIe interface logic itself might have visibility into TOCM). Such errors do not cause this bit to be set, and are reported via the PCI Express interface error bits (secondary status register). Conditions that cause bit 13 to be set, include: <ul style="list-style-type: none"> • Device receives a completion on the primary interface (internal bus of IOH) with Unsupported Request or master abort completion Status. This includes UR status received on the primary side of a PCI Express port on peer-to-peer completions also. • Device accesses to holes in the main memory address region that are detected by the Intel® QuickPath Interconnect source address decoder. • Other master abort conditions detected on the IOH internal bus amongst those listed in Chapter 7, “System Address Map”. |
| 12 | RO | 0 | Received Target Abort This bit is set when a device experiences a completer abort condition on a transaction it mastered on the primary interface (IOH internal bus). Note that certain errors might be detected right at the PCI Express interface and those transactions might not 'propagate' to the primary interface before the error is detected (for example, accesses to memory above VTCSRBASE). Such errors do not cause this bit to be set, and are reported via the PCI Express interface error bits (secondary status register). Conditions that cause bit 12 to be set, include: <ul style="list-style-type: none"> • Device receives a completion on the primary interface (internal bus of IOH) with completer abort completion Status. This includes CA status received on the primary side of a PCI Express port on peer-to-peer completions also. • Accesses to Intel QuickPath Interconnect that return a failed completion status • Other completer abort conditions detected on the IOH internal bus amongst those listed in Chapter 7, “System Address Map”. |



| Device: 19 Function: 0 | | | |
|--|------|---|--|
| Device: 16, 17 Function: 0, 1 | | | |
| Device: 20 Function: 0-3 | | | |
| Offset: 06h | | | |
| Bit | Attr | Default | Description |
| 11 | RO | 0 | Signaled Target Abort This bit is set when a device signals a completer abort completion status on the primary side (internal bus of IOH). This condition includes a PCI Express port forwarding a completer abort status received on a completion from the secondary side and passed to the primary side on a peer2peer completion. |
| 10:9 | RO | 0h | DEVSEL# Timing Not applicable to PCI Express. Hardwired to 0. |
| 8 | RO | 0 | Master Data Parity Error This bit is set by a device if the Parity Error Response bit in the PCI Command register is set and it receives a completion with poisoned data from the primary side or if it forwards a packet with data (including MSI writes) to the primary side with poison. |
| 7 | RO | 0 | Fast Back-to-Back Not applicable to PCI Express. Hardwired to 0. |
| 6 | RV | 0 | <i>Reserved</i> |
| 5 | RO | 0 | 66 MHz capable Not applicable to PCI Express. Hardwired to 0. |
| 4 | RO | Dev_fun: def 16_1:0h 17_1:0h 20_3:0h 21_0:0h else:1h | Capabilities List This bit indicates the presence of a capabilities list structure |
| 3 | RO | 0 | INTx Status Indicates that a legacy INTx interrupt condition is pending internally in the DMA device. This bit has meaning only in the legacy interrupt mode. This bit is always 0 when MSI-X (see Section 21.12.4.7) has been selected for DMA interrupts. Note that the setting of the INTx status bit is independent of the INTx enable bit in the PCI command register, that is, this bit is set anytime the DMA engine is setup by its driver to generate any interrupt and the condition that triggers the interrupt has occurred, regardless of whether a legacy interrupt message was signaled to the ICH or not. Note that the INTx enable bit has to be set in the PCICMD register for DMA to generate a INTx message to the ICH. This bit is not applicable to PCI Express and ESI ports and this bit does not get set for interrupts forwarded from a PCI Express port to the ICH from downstream devices. This bit also does not apply to Perf Mon, I/OxAPIC and DF* register devices. |
| 2:0 | RV | 0h | <i>Reserved</i> |



21.4.2.5 RID: Revision Identification Register

This register contains the revision number of the IOH. The revision number steps the same across all devices and functions, that is, individual devices do not step their RID independently. Note that the revision id for the JTAG IDCODE register also steps with this register.

IOH supports the CRID feature where this register's value can be changed by BIOS.

| Device: 19, 21 Function: 0 | | | |
|--|------|---------|--|
| Device: 16, 17 Function: 0, 1 | | | |
| Device: 20 Function: 0-3 | | | |
| Offset: 08h | | | |
| Bit | Attr | Default | Description |
| 7:4 | RO | 0 | Major Revision Steppings which require all masks to be regenerated. 1: A stepping 2: B stepping |
| 3:0 | RO | 0 | Minor Revision Incremented for each stepping which does not modify all masks. Reset for each major revision. 0: x0 stepping 1: x1 stepping 2: x2 stepping |



21.4.2.6 CCR: Class Code Register

This register contains the Class Code for the device.

| Device: 19 Function: 0 | | | |
|--|------|--|--|
| Device: 16, 17 Function: 0, 1 | | | |
| Device: 20 Function: 0-3 | | | |
| Offset: 09h | | | |
| Bit | Attr | Default | Description |
| 23:16 | RO | Dev: def 13: 06h 14: 06h 15: 11h else: 08h | BaseClass: Base Class Provides the PCIe base class type. Most common registers will default to 08h. (Base system peripherals.) DF* functions related to PCIe and ESI will default to 06h (bridge devices) Performance monitoring (Dev #15) will default to 11h, indicating a "Signal Acquisition Device". ESI, PCIe, and CCRs are defined in their own register sections. |
| 15:8 | RO | Dev: def 15: 01h 19, 20: 00h else: 80h | SubClass: Sub-Class PCI Express/ESI ports, DMA device are covered in their own sections. For I/OxAPIC device (dev#19), this field is always fixed at 00h to indicate interrupt controller. |
| 7:0 | RO | Dev: def 19: 20h else: 00h | RLProgInt: Register-Level Programming Interface This field is hardwired to 20h for I/OxAPIC and is set to 00h for all other devices. |

21.4.2.7 CLS: Cacheline Size Register

| Device: 19 Function: 0 | | | |
|--|------|---------|---|
| Device: 16, 17 Function: 0, 1 | | | |
| Device: 20 Function: 0-3 | | | |
| Offset: 0Ch | | | |
| Bit | Attr | Default | Description |
| 7:0 | RW | 0 | Cacheline Size This register is set as RW for compatibility reasons only. Cacheline size for IOH is always 64B. IOH hardware ignore this setting. |



21.4.2.8 HDR: Header Type Register

This register identifies the header layout of the configuration space.

| Device: 19 Function: 0 Device: 16, 17 Function: 0, 1 Device: 20 Function: 0-3 Offset: 0Eh | | | |
|---|------|--|---|
| Bit | Attr | Default | Description |
| 7 | RO | Dev: def 16: 1 17: 1 19: 1 20: 1 | Multi-function Device: This bit is set for Devices 16, 17, 19 and 20. |
| 6:0 | RO | 00h | Configuration Layout This field identifies the format of the configuration header layout. For devices defined in this section, this is type 0. (Type 1 devices are defined in their own sections) |

21.4.2.9 SVID: Subsystem Vendor ID

Subsystem vendor ID.

| Device: 16, 17 Function: 0, 1 Device: 20 Function: 0-3 Offset: 2Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RWO | 0h | Subsystem Vendor ID Assigned by PCI-SIG for the subsystem vendor |

21.4.2.10 SID: Subsystem Device ID

Subsystem device ID.

| Device: 16, 17 Function: 0, 1 Device: 20 Function: 0-3 Offset: 2Eh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RWO | 00h | Subsystem Device ID Assigned by the subsystem vendor to uniquely identify the subsystem |



21.4.2.11 CAPPTR: Capability Pointer

The CAPPTR is used to point to a linked list of additional capabilities implemented by the device. It provides the offset to the first set of capabilities registers located in the PCI compatible space from 40h.

| Device: 19 Function: 0 | | | |
|--|------|---|---|
| Device: 16, 17 Function: 0, 1 | | | |
| Device: 20 Function: 0-3 | | | |
| Offset: 34h | | | |
| Bit | Attr | Default | Description |
| 7:0 | RO | Dev_fun: Def 16_0: 50h 16_1: 00h 17_0: 50h 17-1: 00h 19_0: 6Ch 20_3: 00h else: 40h | Capability Pointer Points to the first capability structure for the device. |

21.4.2.12 INTL: Interrupt Line Register

The Interrupt Line register is used to communicate interrupt line routing information between initialization code and the device driver.

| Device: 19 Function: 0 | | | |
|--|------|---------|--|
| Device: 16, 17 Function: 0, 1 | | | |
| Device: 20 Function: 0-3 | | | |
| Offset: 3Ch | | | |
| Bit | Attr | Default | Description |
| 7:0 | RO | 0 | Interrupt Line This bit is RW for devices that can generate a legacy INTx message and is needed only for compatibility purposes. |



21.4.2.13 INTP: Interrupt Pin Register

Indicates what INTx message a device generates. This register has no meaning for the IOH devices covered by this section.

| Device: 19 Function: 0 Device: 16, 17 Function: 0, 1 Device: 20 Function: 0-3 Offset: 3Dh | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RO | 0 | Interrupt Pin Only DMA and PCIe are capable of generating INTx interrupt (see INTPIN register in the respective sections). These bits have no meaning for the IOH devices covered by this section and are hard coded to '0'. |

21.4.3 Register Definitions - Extended Config Space

The registers in this section are common for devices/functions with extended configuration space. These registers allow software to access the extended space while running under shrink wrapped OS's. The only exceptions are that the PCI Express ports and DMA registers, which may have additional characteristics are being defined in their own respective sections.

21.4.3.1 CAPID: PCI Express Capability List Register

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 3.0 configuration space.

| Device: 20 Function: 0-2 Offset: 40h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RO | 10h | Capability ID Provides the PCI Express capability ID assigned by PCI-SIG. |

21.4.3.2 NXTPTR: PCI Express Next Capability List Register

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 3.0 configuration space.

| Device: 20 Function: 0-2 Offset: 41h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RO | 0 | Next Ptr This field is set to the PCI PM capability. |



21.4.3.3 EXPCAP: PCI Express Capabilities Register

The PCI Express Capabilities register identifies the PCI Express device type and associated capabilities.

| Device: 20 Function: 0-2 Offset: 42h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:14 | RV | 0h | Reserved |
| 13:9 | RO | 00h | Interrupt Message Number Applies (that is, meaningful) only to the root ports and does not apply to the DMA register devices. This field indicates the interrupt message number that is generated for PM/HP/BW-change events. When there are more than one MSI interrupt Number, this register field is required to contain the offset between the base Message Data and the MSI Message that is generated when the associated status bits in this capability register are set. IOH assigns the first vector for PM/HP/BW-change events and so this field is set to 0. |
| 8 | RO | 0 | Slot Implemented Applies only to the root ports and does not apply to the DMA device. 1: indicates that the PCI Express link associated with the port is connected to a slot. 0: indicates no slot is connected to this port. This register bit is of type "write once" and is controlled by BIOS/special initialization firmware. |
| 7:4 | RO | 1001 | Device/Port Type This field identifies the type of device. It is set to 0100 for all the Express ports and 1001 for the DMA register device. |
| 3:0 | RO | 2h | Capability Version This field identifies the version of the PCI Express capability structure. Set to 2h for PCI Express and DMA devices for compliance with the extended base registers. |

21.4.3.4 DEVCAP: PCI Express Device Capabilities Register

The PCI Express Device Capabilities register identifies device specific information for the device.

| Device: 20 Function: 0-2 Offset: 44h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:28 | RV | 0h | Reserved |
| 27:26 | RO | 0h | Captured Slot Power Limit Scale Does not apply to root ports or integrated devices |
| 25:18 | RO | 0h | Captured Slot Power Limit Value Does not apply to root ports or integrated devices |
| 17:16 | RV | 0h | Reserved |
| 15 | RO | 1h | Role Based Error Reporting: IOH is 1.1 compliant and so supports this feature |
| 14 | RO | 0h | Power Indicator Present on Device Does not apply to root ports or integrated devices |
| 13 | RO | 0h | Attention Indicator Present Does not apply to root ports or integrated devices |



| Device: 20 Function: 0-2 Offset: 44h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 12 | RO | 0 | Attention Button Present Does not apply to root ports or integrated devices |
| 11:9 | RO | 000 | Endpoint L1 Acceptable Latency Does not apply to IOH |
| 8:6 | RO | 000 | Endpoint L0s Acceptable Latency Does not apply to IOH |
| 5 | RO | 0 | Extended Tag Field Supported IOH devices support only 5-bit tag field. |
| 4:3 | RO | 0h | Phantom Functions Supported IOH does not support phantom functions. |
| 2:0 | RO | 000 | Max Payload Size Supported IOH supports 256B payloads on Express port and 128B on the remainder of the devices. |

21.4.3.5 DEVCON: PCI Express Device Control Register

The PCI Express Device Control register controls PCI Express specific capabilities parameters associated with the device.

| Device: 20 Function: 0-2 Offset: 48h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15 | RV | 0h | Reserved |
| 14:12 | RO | 000 | Max_Read_Request_Size Express/ESI/DMA ports in IOH do not generate requests greater than 128B and this field is ignored. |
| 11 | RO | 0 | Enable No Snoop Not applicable to root ports since they never set the 'No Snoop' bit for transactions they originate (not forwarded from peer) to PCI Express. For DMA, when this bit is clear, all DMA transactions must be snooped. When set, DMA transactions to main memory can utilize No Snoop optimization under the guidance of the device driver. This bit has no impact on forwarding of NoSnoop attribute on peer requests. |
| 10 | RO | 0 | Auxiliary Power Management Enable Not applicable to IOH |
| 9 | RO | 0 | Phantom Functions Enable Not applicable to IOH since it never uses phantom functions as a requester. |
| 8 | RO | 0h | Extended Tag Field Enable This bit enables the PCI Express port/ESI to use an 8-bit Tag field as a requester. |



| Device: 20 Function: 0-2 Offset: 48h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:5 | RO | 000 | Max Payload Size This field is set by configuration software for the maximum TLP payload size for the PCI Express port. As a receiver, the IOH must handle TLPs as large as the set value. As a requester (that is, for requests where IOH's own RequesterID is used), it must not generate TLPs exceeding the set value. Permissible values that can be programmed are indicated by the Max_Payload_Size_Supported in the Device Capabilities register: 000: 128B max payload size 001: 256B max payload size (applies only to standard PCI Express ports and other devices alias to 128B) others: alias to 128B |
| 4 | RO | 0 | Enable Relaxed Ordering Not applicable to root ports since they never set relaxed ordering bit as a requester (this does not include tx forwarded from peer devices). For DMA, when this bit is clear, all DMA transactions must follow strict ordering. When set, DMA transactions are allowed to be relaxed ordered under the guidance of the device driver. This bit has no impact on forwarding of relaxed ordering attribute on peer requests. |
| 3 | RO | 0 | Unsupported Request Reporting Enable Applies only to the PCI Express/ESI ports. This bit controls the reporting of unsupported requests that IOH itself detects on requests its receives from a PCI Express/ESI port. 0: Reporting of unsupported requests is disabled 1: Reporting of unsupported requests is enabled. <i>Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for complete details of how this bit is used in conjunction with other bits to UR errors.</i> |
| 2 | RO | 0 | Fatal Error Reporting Enable Applies only to the PCI Express/ESI ports. Controls the reporting of fatal errors that IOH detects on the PCI Express/ESI interface. 0: Reporting of Fatal error detected by device is disabled 1: Reporting of Fatal error detected by device is enabled <i>Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for complete details of how this bit is used in conjunction with other bits to report errors.</i> For the PCI Express/ESI ports, this bit is not used to control the reporting of other internal component uncorrectable fatal errors (at the port unit) in any way. |
| 1 | RO | 0 | Non Fatal Error Reporting Enable Applies only to the PCI Express/ESI ports. Controls the reporting of non-fatal errors that IOH detects on the PCI Express/ESI interface or any non-fatal errors that DMA detect 0: Reporting of Non Fatal error detected by device is disabled 1: Reporting of Non Fatal error detected by device is enabled <i>Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for complete details of how this bit is used in conjunction with other bits to report errors.</i> For the PCI Express/ESI ports, this bit is not used to control the reporting of other internal component uncorrectable non-fatal errors (at the port unit) in any way. |



| Device: 20 Function: 0-2 Offset: 48h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 0 | RO | 0 | Correctable Error Reporting Enable Applies only to the PCI Express/ESI ports. Controls the reporting of correctable errors that IOH detects on the PCI Express/ESI interface 0: Reporting of link Correctable error detected by the port is disabled 1: Reporting of link Correctable error detected by port is enabled Refer to <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s</i> for complete details of how this bit is used in conjunction with other bits to report errors. For the PCI Express/ESI ports, this bit is not used to control the reporting of other internal component correctable errors (at the port unit) in any way. |

21.4.3.6 DEVSTS: PCI Express Device Status Register

The PCI Express Device Status register provides information about PCI Express device specific parameters associated with the device.

| Device: 20 Function: 0-2 Offset: 4Ah | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:6 | RV | 0 | <i>Reserved.</i> |
| 5 | RO | 0 | Transactions Pending Does not apply to root/ESI ports, I/OxAPIC bit hardwired to 0 for these devices. 1: indicates that the DMA device has outstanding Non-Posted Request which it has issued either towards main memory or a peer PCI Express port, which have not been completed. 0: DMA reports this bit cleared only when all Completions for any outstanding Non-Posted Requests it owns have been received. |
| 4 | RO | 0 | AUX Power Detected Does not apply to IOH |
| 3 | RO | 0 | Unsupported Request Detected This bit applies only to the root/ESI ports and does not apply to DMA, I/OxAPIC devices hardwire this bit to 0. This bit indicates that the root port detected an Unsupported Request. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control Register. 1: Unsupported Request detected at the device/port. These unsupported requests are NP requests inbound that the root port received and it detected them as unsupported requests (for example, address decoding failures that the root port detected on a packet, receiving inbound lock reads, BME bit is clear, and so on). Note that this bit is not set on peer2peer completions with UR status that are forwarded by the root port to the PCIe link. 0: No unsupported request detected by the root port |
| 2 | RO | 0 | Fatal Error Detected This bit applies only to the root/ESI ports and does not apply to DMA, I/OxAPIC devices hardwire this bit to 0. This bit indicates that a fatal (uncorrectable) error is detected by the device. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. 1: Fatal errors detected 0: No Fatal errors detected |



| Device: 20 Function: 0-2 Offset: 4Ah | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 1 | RO | 0 | Non Fatal Error Detected This bit applies only to the root/ESI ports and does not apply to DMA, I/OxAPIC devices hardwire this bit to 0. This bit gets set if a non-fatal uncorrectable error is detected by the device. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. 1: Non Fatal errors detected 0: No non-Fatal Errors detected |
| 0 | RO | 0 | Correctable Error Detected This bit applies only to the root/ESI ports and does not apply to DMA, I/OxAPIC devices hardwire this bit to 0. This bit gets set if a correctable error is detected by the device. Errors are logged in this register regardless of whether error reporting is enabled or not in the PCI Express Device Control register. 1: correctable errors detected 0: No correctable errors detected |

21.4.3.7 LNKCAP: PCI Express Link Capabilities Register

The Link Capabilities register identifies the PCI Express specific link capabilities.

| Device: 20 Function: 0-2 Offset: 4Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:24 | RO | 0 | Port Number This field indicates the PCI Express port number for the link and is initialized by software/BIOS. |
| 23:22 | RV | 0h | <i>Reserved.</i> |
| 21 | RO | 1 | Link Bandwidth Notification Capability A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. |
| 20 | RO | 1 | Data Link Layer Link Active Reporting Capable IOH supports reporting status of the data link layer so software knows when it can enumerate a device on the link or otherwise know the status of the link. |
| 19 | RO | 1 | Surprise Down Error Reporting Capable IOH supports reporting a surprise down error condition |
| 18 | RO | 0 | Clock Power Management Does not apply to IOH. |
| 17:15 | RO | 7h | L1 Exit Latency IOH does not support L1 ASPM |
| 14:12 | RO | 7h | LOs Exit Latency |
| 11:10 | RO | 01 | Active State Link PM Support Only LOs is supported |



| Device: 20 Function: 0-2 Offset: 4Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 9:4 | RO | 0 | Maximum Link Width This field indicates the maximum width of the given PCI Express Link attached to the port. 000001: x1 000010: x2 ¹ 000100: x4 001000: x8 010000: x16 Others - <i>Reserved</i> This is left as a RWO register for bios to update based on the platform usage of the links. |
| 3:0 | RO | 0 | Link Speeds Supported IOH supports both 2.5 Gbps and 5Gbps speeds if Gen2_OFF fuse is OFF else it supports only Gen1 This register is RWO when Gen2_OFF so that BIOS can change the supported speeds field to be 0001b (Gen1 only) if the board routing is not capable of Gen2 (even though IOH silicon itself is capable of Gen2) This bit is RWO if Gen2_OFF fuse is OFF and is RO if Gen2_OFF fuse is ON. |

Notes:

1. There are restrictions with routing x2 lanes from IOH to a slot. See [Section 5.2](#) for details.

21.4.3.8 LNKCON: PCI Express Link Control Register

The PCI Express Link Control register controls the PCI Express Link specific parameters.

| Device: 20 Function: 0-2 Offset: 50h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:12 | RV | 0 | Reserved |
| 11 | RO | 0 | Link Autonomous Bandwidth Interrupt Enable When set to 1b this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been set. |
| 10 | RO | 0 | Link Bandwidth Management Interrupt Enable When set to 1b this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been set. |
| 9 | RO | 0 | Hardware Autonomous Width Disable IOH never changes a configured link width for reasons other than reliability. |
| 8 | RO | 0 | Enable Clock Power Management N/A to IOH |
| 7 | RO | 0 | Extended Synch This bit when set forces the transmission of additional ordered sets when exiting L0s and when in recovery. See PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for details. |
| 6 | RO | 0 | Common Clock Configuration IOH does nothing with this bit |



| Device: 20 Function: 0-2 Offset: 50h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 5 | RO | 0 | Retrain Link A write of 1 to this bit initiates link retraining in the given PCI Express port by directing the LTSSM to the recovery state if the current state is [L0, L0s or L1]. If the current state is anything other than L0, L0s, L1 then a write to this bit does nothing. This bit always returns 0 when read. If the Target Link Speed field has been set to a non-zero value different than the current operating speed, then the LTSSM will attempt to negotiate to the target link speed. It is permitted to write 1b to this bit while simultaneously writing modified values to other fields in this register. When this is done, all modified values that affect link retraining must be applied in the subsequent retraining. |
| 4 | RO | 0 | Link Disable This field controls whether the link associated with the PCI Express port is enabled or disabled. When this bit is a 1, a previously configured link (a link that has gone past the polling state) would return to the "disabled" state as defined in the <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*'s</i> . When this bit is clear, an LTSSM in the "disabled" state goes back to the detect state. 0: Enables the link associated with the PCI Express port 1: Disables the link associated with the PCI Express port |
| 3 | RO | 0 | Read Completion Boundary Set to zero to indicate IOH could return read completions at 64B boundaries |
| 2 | RV | 0 | <i>Reserved.</i> |
| 1:0 | RO | 00 | Active State Link PM Control When 01b or 11b, L0s on transmitter is enabled, otherwise it is disabled. |

21.4.3.9 LNKSTS: PCI Express Link Status Register

The PCI Express Link Status register provides information on the status of the PCI Express Link such as negotiated width, training, and so on.

| Device: 20 Function: 0-2 Offset: 52h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15 | RO | 0 | Link Autonomous Bandwidth Status This bit is set to 1b by hardware to indicate that hardware has autonomously changed link speed or width, without the port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable link operation. IOH sets this bit when it receives eight consecutive TS1 or TS2 ordered sets with the Autonomous Change bit set. Note that if the status bit is set by hardware in the same clock software clears the status bit, the status bit should remain set and if MSI is enabled, the hardware should trigger a new MSI. |
| 14 | RO | 0 | Link Bandwidth Management Status This bit is set to 1b by hardware to indicate that either of the following has occurred without the port transitioning through DL_Down status: a) A link retraining initiated by a write of 1b to the Retrain Link bit has completed b) Hardware has autonomously changed link speed or width to attempt to correct unreliable link operation Note that if the status bit is set by hardware in the same clock software clears the status bit, the status bit should remain set and if MSI is enabled, the hardware should trigger a new MSI. |



| Device: 20 Function: 0-2 Offset: 52h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 13 | RO | 0 | Data Link Layer Link Active Set to 1b when the Data Link Control and Management State Machine is in the DL_Active state, 0b otherwise. On a downstream port or upstream port, when this bit is 0b, the transaction layer associated with the link will abort all transactions that would otherwise be routed to that link. |
| 12 | RO | 1 | Slot Clock Configuration This bit indicates whether IOH receives clock from the same xtal that also provides clock to the device on the other end of the link. 1: indicates that same xtal provides clocks to devices on both ends of the link 0: indicates that different xtals provide clocks to devices on both ends of the link |
| 11 | RO | 0 | Link Training This field indicates the status of an ongoing link training session in the PCI Express port 0: LTSSM has exited the recovery/configuration state 1: LTSSM is in recovery/configuration state or the Retrain Link was set but training has not yet begun. The IOH hardware clears this bit once LTSSM has exited the recovery/configuration state. Refer to <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s</i> for details of which states within the LTSSM would set this bit and which states would clear this bit. |
| 10 | RV | 0 | Reserved |
| 9:4 | RO | 0 | Negotiated Link Width This field indicates the negotiated width of the given PCI Express link after training is completed. Only x1, x2, x4 and x8 link width negotiations are possible in IOH. A value of 0x01 in this field corresponds to a link width of x1, 0x02 indicates a link width of x2 and so on, with a value of 0x8 for a link width of x8. The value in this field is reserved and could show any value when the link is not up. Software determines if the link is up or not by reading bit 13 of this register. |
| 3:0 | RO | 0 | Current Link Speed This field indicates the negotiated Link speed of the given PCI Express Link. 0001 - 2.5 Gbps 0010 - 5Gbps (IOH will never set this value when Gen2_OFF fuse is blown) Others - <i>Reserved</i> The value in this field is not defined and could show any value, when the link is not up. Software determines if the link is up or not by reading bit 13 of this register. |



21.4.3.10 SLTCAP: PCI Express Slot Capabilities Register

The Slot Capabilities register identifies the PCI Express specific slot capabilities. These registers must be ignored by software on the ESI links.

| Device: 20 Function: 0-2 Offset: 54h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:19 | RO | 0 | Physical Slot Number This field indicates the physical slot number of the slot connected to the PCI Express port and is initialized by bios. |
| 18 | RO | 0 | Command Complete Not Capable: IOH is capable of command complete interrupt. |
| 17 | RO | 0 | Electromechanical Interlock Present This bit when set indicates that an Electromechanical Interlock is implemented on the chassis for this slot and that lock is controlled by bit 11 in Slot Control register. Bios note: this capability is not set if the Electromechanical Interlock control is connected to main slot power control. |
| 16:15 | RO | 0 | Slot Power Limit Scale This field specifies the scale used for the Slot Power Limit Value and is initialized by bios. IOH uses this field when it sends a Set_Slot_Power_Limit message on PCI Express. Range of Values: 00: 1.0x 01: 0.1x 10: 0.01x 11: 0.001x |
| 14:7 | RO | 0 | Slot Power Limit Value This field specifies the upper limit on power supplied by slot in conjunction with the Slot Power Limit Scale value defined previously $Power\ limit\ (in\ Watts) = SPSL \times SPLV$. This field is initialized by bios. IOH uses this field when it sends a Set_Slot_Power_Limit message on PCI Express. Design note: IOH can chose to send the Set_Slot_Power_Limit message on the link at first link up condition without regards to whether this register and the Slot Power Limit Scale register are programmed yet by bios. IOH must then be designed to discard a received Set_Slot_Power_Limit message without an error. |
| 6 | RO | 0 | Hot-plug Capable This field defines hot-plug support capabilities for the PCI Express port. 0: indicates that this slot is not capable of supporting Hot-plug operations. 1: indicates that this slot is capable of supporting hot-plug operations This bit is programmed by BIOS based on the system design. This bit must be programmed by bios to be consistent with the VPP enable bit for the port. |



| Device: 20 Function: 0-2 Offset: 54h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 5 | RO | 0 | <p>Hot-Plug Surprise This field indicates that a device in this slot may be removed from the system without prior notification (like for instance a PCI Express cable). 0: indicates that hot-plug surprise is not supported 1: indicates that hot-plug surprise is supported</p> <p>Note that if platform implemented cable solution (either direct or via a SIOM with repeater), on a port, then this could be set. BIOS programs this field with a 0 for CEM/SIOM FFs.</p> <p>This bit is used by IOH hardware to determine if a transition from DL_active to DL_Inactive is to be treated as a surprise down error or not. If a port is associated with a hotpluggable slot and the hotplug surprise bit is set, then any transition to DL_Inactive is not considered an error. Refer to <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s</i> for further details.</p> |
| 4 | RO | 0 | <p>Power Indicator Present This bit indicates that a Power Indicator is implemented for this slot and is electrically controlled by the chassis. 0: indicates that a Power Indicator that is electrically controlled by the chassis is not present 1: indicates that Power Indicator that is electrically controlled by the chassis is present</p> <p>BIOS programs this field with a 1 for CEM/SIOM FFs and a 0 for Express cable.</p> |
| 3 | RO | 0 | <p>Attention Indicator Present This bit indicates that an Attention Indicator is implemented for this slot and is electrically controlled by the chassis 0: indicates that an Attention Indicator that is electrically controlled by the chassis is not present 1: indicates that an Attention Indicator that is electrically controlled by the chassis is present</p> <p>BIOS programs this field with a 1 for CEM/SIOM FFs.</p> |
| 2 | RO | 0 | <p>MRL Sensor Present This bit indicates that an MRL Sensor is implemented on the chassis for this slot. 0: indicates that an MRL Sensor is not present 1: indicates that an MRL Sensor is present</p> <p>BIOS programs this field with a 0 for SIOM/Express cable and with either 0 or 1 for CEM depending on system design.</p> |
| 1 | RO | 0 | <p>Power Controller Present This bit indicates that a software controllable power controller is implemented on the chassis for this slot. 0: indicates that a software controllable power controller is not present 1: indicates that a software controllable power controller is present</p> <p>BIOS programs this field with a 1 for CEM/SIOM FFs and a 0 for Express cable.</p> |
| 0 | RO | 0 | <p>Attention Button Present This bit indicates that the Attention Button event signal is routed (from slot or on-board in the chassis) to the IOH's hotplug controller. 0: indicates that an Attention Button signal is routed to IOH 1: indicates that an Attention Button is not routed to IOH</p> <p>BIOS programs this field with a 1 for CEM/SIOM FFs.</p> |



21.4.3.11 SLTCON: PCI Express Slot Control Register

The Slot Control register identifies the PCI Express specific slot control parameters for operations such as Hot-Plug and Power Management.

| Device: 20 Function: 0-2 Offset: 58h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:13 | RV | 0h | <i>Reserved.</i> |
| 12 | RO | 0 | Data Link Layer State Changed Enable When set to 1, this field enables software notification when Data Link Layer Link Active field is changed |
| 11 | RO | 0 | Electromechanical Interlock Control When software writes either a 1 to this bit, IOH pulses the EMIL pin per <i>PCI Express Server/Workstation Module Electromechanical Spec Rev 0.5a</i> . Write of 0 has no effect. This bit always returns a 0 when read. If electromechanical lock is not implemented, then either a write of 1 or 0 to this register has no effect. |
| 10 | RO | 1 | Power Controller Control If a power controller is implemented, when written sets the power state of the slot per the defined encodings. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not executed yet at the VPP, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. 0: Power On 1: Power Off |
| 9:8 | RO | 3h | Power Indicator Control If a Power Indicator is implemented, writes to this register set the Power Indicator to the written state. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not executed yet at the VPP, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. 00: <i>Reserved.</i> 01: On 10: Blink (IOH drives 1.5 Hz square wave for Chassis mounted LEDs) 11: Off When this register is written, the event is signaled via the virtual pins of the IOH over a dedicated SMBus port. IOH does not generated the Power_Indicator_On/Off/Blink messages on PCI Express when this field is written to by software. |
| 7:6 | RO | 3h | Attention Indicator Control If an Attention Indicator is implemented, writes to this register set the Attention Indicator to the written state. Reads of this field reflect the value from the latest write, even if the corresponding hot-plug command is not executed yet at the VPP, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. 00: <i>Reserved.</i> 01: On 10: Blink (The IOH drives 1.5 Hz square wave) 11: Off When this register is written, the event is signaled via the virtual pins of the IOH over a dedicated SMBus port. IOH does not generated the Attention_Indicator_On/Off/Blink messages on PCI Express when this field is written to by software. |



| Device: 20 Function: 0-2 Offset: 58h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 5 | RO | 0 | Hot-plug Interrupt Enable When set to 1b, this bit enables generation of Hot-Plug MSI interrupt (and not wake event) on enabled Hot-Plug events, provided ACPI mode for hotplug is disabled. 0: disables interrupt generation on Hot-plug events 1: enables interrupt generation on Hot-plug events |
| 4 | RO | 0 | Command Completed Interrupt Enable This field enables the generation of Hot-plug interrupts (and not wake event) when a command is completed by the Hot-plug controller connected to the PCI Express port 0: disables hot-plug interrupts on a command completion by a hot-plug Controller 1: Enables hot-plug interrupts on a command completion by a hot-plug Controller |
| 3 | RO | 0 | Presence Detect Changed Enable This bit enables the generation of hot-plug interrupts or wake messages via a presence detect changed event. 0: disables generation of hot-plug interrupts or wake messages when a presence detect changed event happens. 1- Enables generation of hot-plug interrupts or wake messages when a presence detect changed event happens. |
| 2 | RO | 0 | MRL Sensor Changed Enable This bit enables the generation of hot-plug interrupts or wake messages via a MRL Sensor changed event. 0: disables generation of hot-plug interrupts or wake messages when an MRL Sensor changed event happens. 1: Enables generation of hot-plug interrupts or wake messages when an MRL Sensor changed event happens. |
| 1 | RO | 0 | Power Fault Detected Enable This bit enables the generation of hot-plug interrupts or wake messages via a power fault event. 0: disables generation of hot-plug interrupts or wake messages when a power fault event happens. 1: Enables generation of hot-plug interrupts or wake messages when a power fault event happens. |
| 0 | RO | 0 | Attention Button Pressed Enable This bit enables the generation of hot-plug interrupts or wake messages via an attention button pressed event. 0: disables generation of hot-plug interrupts or wake messages when the attention button is pressed. 1: Enables generation of hot-plug interrupts or wake messages when the attention button is pressed. |



21.4.3.12 SLTSTS: PCI Express Slot Status Register

The PCI Express Slot Status register defines important status information for operations such as Hot-Plug and Power Management.

| Device: 20 Function: 0-2 Offset: 5Ah | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:9 | RV | 0h | <i>Reserved.</i> |
| 8 | RO | 0 | Data Link Layer State Changed This bit is set (if it is not already set) when the state of the Data Link Layer Link Active bit in the Link Status register changes. Software must read Data Link Layer Active field to determine the link state before initiating configuration cycles to the hot plugged device. |
| 7 | RO | 0 | Electromechanical Latch Status When read this register returns the current state of the Electromechanical Interlock (the EMILS pin) which has the defined encodings as: 0b Electromechanical Interlock Disengaged 1b Electromechanical Interlock Engaged |
| 6 | RO | 0 | Presence Detect State For ports with slots (where the Slot Implemented bit of the PCI Express Capabilities Registers is 1b), this field is the logical OR of the Presence Detect status determined via an in-band mechanism and sideband Present Detect pins. Refer to how PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for how the inband presence detect mechanism works (certain states in the LTSSM constitute "card present" and others don't). 0: Card/Module/Cable slot empty or Cable Slot occupied but not powered 1: Card/module Present in slot (powered or unpowered) or cable present and powered on other end For ports with no slots, IOH hardwires this bit to 1b. Note: OS could get confused when it sees an empty PCI Express root port that is, "no slots + no presence", since this is now disallowed in the spec. So bios must hide all unused root ports devices in IOH config space, via the DEVHIDE register in Intel QuickPath Interconnect CSR space. |
| 5 | RO | 0 | MRL Sensor State This bit reports the status of an MRL sensor if it is implemented. 0: MRL Closed 1: MRL Open |
| 4 | RO | 0 | Command Completed This bit is set by the IOH when the hot-plug command has completed and the hot-plug controller is ready to accept a subsequent command. It is subsequently cleared by software after the field has been read and processed. This bit provides no guarantee that the action corresponding to the command is complete. |



| Device: 20 Function: 0-2 Offset: 5Ah | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 3 | RO | 0 | Presence Detect Changed This bit is set by the IOH when a Presence Detect Changed event is detected. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support out-of-band presence detect. |
| 2 | RO | 0 | MRL Sensor Changed This bit is set by the IOH when an MRL Sensor Changed event is detected. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support MRL. |
| 1 | RO | 0 | Power Fault Detected This bit is set by the IOH when a power fault event is detected by the power controller. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support power fault detection. |
| 0 | RO | 0 | Attention Button Pressed This bit is set by the IOH when the attention button is pressed. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support attention button. IOH silently discards the Attention_Button_Pressed message if received from PCI Express link without updating this bit. |

21.4.3.13 ROOTCON: PCI Express Root Control Register

The PCI Express Root Control register specifies parameters specific to the root complex port.

| Device: 20 Function: 0-2 Offset: 5Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:5 | RV | 0h | <i>Reserved.</i> |
| 4 | RO | 0 | CRS software visibility Enable This bit, when set, enables the Root Port to return Configuration Request Retry Status (CRS) Completion Status to software. |
| 3 | RO | 0 | PME Interrupt Enable (Applies only to devices 0-8. This bit is a don't care for device 8) This field controls the generation of MSI interrupts for PME messages. 1: Enables interrupt generation upon receipt of a PME message 0: Disables interrupt generation for PME messages. |



| Device: 20 Function: 0-2 Offset: 5Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 2 | RO | 0 | <p>System Error on Fatal Error Enable</p> <p>This field enables notifying the internal core error logic of occurrence of an uncorrectable fatal error at the port or below its hierarchy. The internal core error logic of IOH then decides if/how to escalate the error further (pins/ message etc). Refer to Section 16.2 for details of how/which system notification is generated for a PCI Express/ESI fatal error.</p> <p>1: indicates that a internal core error logic notification should be generated if a fatal error (ERR_FATAL) is reported by any of the devices in the hierarchy associated with and including this port.</p> <p>0: No internal core error logic notification should be generated on a fatal error (ERR_FATAL) reported by any of the devices in the hierarchy associated with and including this port.</p> <p>Note that generation of system notification on a PCI Express/ESI fatal error is orthogonal to generation of an MSI interrupt for the same error. Both a system error and MSI can be generated on a fatal error or software can chose one of the two.</p> <p>Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for details of how this bit is used in conjunction with other error control bits to generate core logic notification of error events in a PCI Express/ESI port.</p> |
| 1 | RO | 0 | <p>System Error on Non-Fatal Error Enable</p> <p>This field enables notifying the internal core error logic of occurrence of an uncorrectable non-fatal error at the port or below its hierarchy. The internal core error logic of IOH then decides if/how to escalate the error further (pins/ message etc). Refer to Section 16 for details of how/which system notification is generated for a PCI Express/ESI non-fatal error.</p> <p>1: indicates that a internal core error logic notification should be generated if a non-fatal error (ERR_NONFATAL) is reported by any of the devices in the hierarchy associated with and including this port.</p> <p>0: No internal core error logic notification should be generated on a non-fatal error (ERR_NONFATAL) reported by any of the devices in the hierarchy associated with and including this port.</p> <p>Note that generation of system notification on a PCI Express/ESI non-fatal error is orthogonal to generation of an MSI interrupt for the same error. Both a system error and MSI can be generated on a non-fatal error or software can chose one of the two.</p> <p>Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for details of how this bit is used in conjunction with other error control bits to generate core logic notification of error events in a PCI Express/ESI port.</p> |



| Device: 20 Function: 0-2 Offset: 5Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 0 | RO | 0 | <p>System Error on Correctable Error Enable</p> <p>This field controls notifying the internal core error logic of the occurrence of a correctable error in the device or below its hierarchy. The internal core error logic of IOH then decides if/how to escalate the error further (pins/message etc). Refer to Section 16.2 for details of how/which system notification is generated for a PCI Express correctable error.</p> <p>1: indicates that an internal core error logic notification should be generated if a correctable error (ERR_COR) is reported by any of the devices in the hierarchy associated with and including this port.</p> <p>0: No internal core error logic notification should be generated on a correctable error (ERR_COR) reported by any of the devices in the hierarchy associated with and including this port.</p> <p>Note that generation of system notification on a PCI Express correctable error is orthogonal to generation of an MSI interrupt for the same error. Both a system error and MSI can be generated on a correctable error or software can chose one of the two.</p> <p>Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for details of how this bit is used in conjunction with other error control bits to generate core logic notification of error events in a PCI Express/ESI port.</p> |

21.4.3.14 ROOTCAP: PCI Express Root Capabilities Register

The PCI Express Root Status register specifies parameters specific to the root complex port.

| Device: 20 Function: 0-2 Offset: 5Eh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:1 | RV | 0h | <i>Reserved.</i> |
| 0 | RO | 1 | <p>CRS Software Visibility</p> <p>This bit, when set, indicates that the Root Port is capable of returning Configuration Request Retry Status (CRS) Completion Status to software. IOH supports this capability.</p> |



21.4.3.15 ROOTSTS: PCI Express Root Status Register

The PCI Express Root Status register specifies parameters specific to the root complex port.

| Device: 20 Function: 0-2 Offset: 60h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:18 | RV | 0h | <i>Reserved.</i> |
| 17 | RO | 0 | PME Pending This field indicates that another PME is pending when the PME Status bit is set. When the PME Status bit is cleared by software; the pending PME is delivered by hardware by setting the PME Status bit again and updating the Requestor ID appropriately. The PME pending bit is cleared by hardware if no more PMEs are pending. |
| 16 | RO | 0 | PME Status This field indicates a PM_PME message (either from the link or internally from within that root port) was received at the port. 1: PME was asserted by a requester as indicated by the PMEREQID field This bit is cleared by software by writing a '1'. Note that the root port itself could be the source of a PME event when a hotplug event is observed when the port is in D3hot state. |
| 15:0 | RO | 0 | PME Requester ID This field indicates the PCI requester ID of the last PME requester. If the root port itself was the source of the (virtual) PME message, then a RequesterID of IOHBUSNO:DevNo:0 is logged in this field. |

21.4.3.16 DEVCAP2: PCI Express Device Capabilities 2 Register

The PCI Express Device Capabilities register identifies device specific information for the device.

| Device: 20 Function: 0-2 Offset: 64h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:6 | RV | 0 | Reserved |
| 5 | RO | 0 | Alternative RID Interpretation (ARI) Capable This bit is hardwired to 0b indicating no support for this capability. |
| 4 | RO | 0 | Completion Timeout Disable Support IOH does not support disabling completion timeout |
| 3:0 | RO | 0000 | Completion Timeout Values Supported This field indicates device support for the optional Completion Timeout programmability mechanism. 0000b: Completions Timeout programming not supported. |



21.4.3.17 DEVCON2: PCI Express Device Control 2 Register

The PCI Express Device Control register controls PCI Express specific capabilities parameters associated with the device.

| Device: 20 Function: 0-2 Offset: 68h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:6 | RV | 0 | Reserved |
| 5 | RO | 0 | Alternative RID Interpretation (ARI) Enable When set to 1b, ARI is enabled in Root Port. |
| 4 | RO | 0 | Completion Timeout Disable When set to 1b, this bit disables the Completion Timeout Mechanism for all NP tx that IOH issues on the PCIe/DMI link and in the case of CBDMA, for all NP tx that DMA issues upstream. When set to 0b, completion timeout is enabled. Software can change this field while there is active traffic in the root port. |
| 3:0 | RO | 0000b | Completion Timeout Value on NP Tx that IOH Issues on PCIe/DMI: In devices that support Completion Timeout Programmability, this field allows system software to modify the Completion Timeout range. The following encodings and corresponding timeout ranges are defined: 0000b: 2ms 0001b: Reserved (IOH aliases to 0000b) 0010b: Reserved (IOH aliases to 0000b) 0101b: 4ms 0110b: 10ms 1001b: 40ms 1010b: 210ms 1101b: 800ms 1110b: 2s to 6.5s Note: These values can deviate +/-10% When the OS selects the 2s - 6.5s range, CTOCTRL further controls the timeout value within that range. For all other ranges selected by OS, the timeout value within that range is fixed in the IOH hardware. Software can change this field while there is active traffic in the root port. This value is also used to control PME_TO_ACK timeout. This field sets the timeout value for receiving the PME_TO_ACK message after a PME_TURN_OFF message has been transmitted. The PME_TO_ACK timeout has meaning only if Bit 6 of MISCCTRLSTS is set to 1b. |

21.4.3.18 DEVSTS2: PCI Express Device Status 2 Register

The PCI Express Device Status register provides information about PCI Express device specific parameters associated with the device.

| Device: 20 Function: 0-2 Offset: 6Ah | | | |
|---|------|---------|------------------|
| Bit | Attr | Default | Description |
| 15:0 | RV | 0h | <i>Reserved.</i> |



21.4.3.19 LNKCAP2: PCI Express Link Capabilities 2 Register

The Link Capabilities register identifies the PCI Express specific link capabilities.

| Device: 20 Function: 0-2 Offset: 6Ch | | | |
|--|------|---------|-----------------|
| Bit | Attr | Default | Description |
| 31:0 | RV | 0 | <i>Reserved</i> |

21.4.3.20 LNKCON2: PCI Express Link Control 2 Register

The PCI Express Link Control register controls the PCI Express Link specific parameters.

| Device: 20 Function: 0-2 Offset: 70h | | | |
|--|------|---------|-----------------|
| Bit | Attr | Default | Description |
| 15:0 | RV | 0 | <i>Reserved</i> |

21.4.3.21 LNKSTS2: PCI Express Link Status 2 Register

The PCI Express Link Status register provides information on the status of the PCI Express Link such as negotiated width, training, and so on.

| Device: 20 Function: 0-2 Offset: 72h | | | |
|--|------|---------|-----------------|
| Bit | Attr | Default | Description |
| 15:0 | RV | 0 | <i>Reserved</i> |

21.4.3.22 SLTCAP2: PCI Express Slot Capabilities 2 Register

The Slot Capabilities register identifies the PCI Express specific slot capabilities. These registers must be ignored by software on the ESI links.

| Device: 20 Function: 0-2 Offset: 74h | | | |
|--|------|---------|-----------------|
| Bit | Attr | Default | Description |
| 31:0 | RV | 0 | <i>Reserved</i> |



21.4.3.23 SLTCON2: PCI Express Slot Control 2 Register

The Slot Control register identifies the PCI Express specific slot control parameters for operations such as Hot-plug and Power Management.

| Device: 20 Function: 0-2 Offset: 78h | | | |
|---|------|---------|------------------|
| Bit | Attr | Default | Description |
| 15:0 | RV | 0 | <i>Reserved.</i> |

21.4.3.24 SLTSTS2: PCI Express Slot Status 2 Register

The PCI Express Slot Status register defines important status information for operations such as Hot-plug and Power Management.

| Device: 20 Function: 0-2 Offset: 7Ah | | | |
|---|------|---------|------------------|
| Bit | Attr | Default | Description |
| 15:0 | RV | 0 | <i>Reserved.</i> |



21.5 IOxAPIC Controller

Table 21-4. IOH Device 19 I/OxAPIC Configuration Map - Offset 0x00-0xFF

| | | | | | | | |
|----------------|--|--------|-----|-----|-------------|---------|-----|
| DID | | VID | | 00h | | RDINDEX | 80h |
| PCISTS | | PCICMD | | 04h | | | 84h |
| CCR | | | RID | 08h | | | 88h |
| HDR | | | CLS | 0Ch | | | 8Ch |
| MBAR | | | | 10h | RDWINDOW | | 90h |
| | | | | 14h | | | 94h |
| | | | | 18h | | | 98h |
| | | | | 1Ch | | | 9Ch |
| | | | | 20h | IOAPICTETPC | | A0h |
| | | | | 24h | | | A4h |
| | | | | 28h | | | A8h |
| SID | | SVID | | 2Ch | | | ACH |
| | | | | 30h | | | B0h |
| | | | | 34h | CAPPTR | | B4h |
| | | | | 38h | | | B8h |
| | | | | 3Ch | | | BCh |
| | | | | 40h | ABAR | | C0h |
| | | | | 44h | | | C4h |
| | | | | 48h | | | C8h |
| | | | | 4Ch | | | CCh |
| | | | | 50h | | | D0h |
| | | | | 54h | | | D4h |
| | | | | 58h | | | D8h |
| | | | | 5Ch | | | DCh |
| | | | | 60h | | | E0h |
| | | | | 64h | | | E4h |
| | | | | 68h | | | E8h |
| PMCAP PMCSR | | | | 6Ch | | | ECh |
| | | | | 70h | | | F0h |
| | | | | 74h | | | F4h |
| | | | | 78h | | | F8h |
| | | | | 7Ch | | | FCh |



21.5.1 PCICMD: PCI Command Register (Dev #19)

This register defines the PCI 3.0 compatible command register values applicable to PCI Express space.

| Device: 19 Function: 0 Offset: 04h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:11 | RV | 0 | Reserved (by PCI SIG) |
| 10 | RO | 0 | Interrupt Disable Controls the ability of DMA to generate legacy INTx interrupt (when legacy INTx mode is enabled). This bit does not affect the ability of the Express port to route interrupt messages received at the PCI Express port. 1: Legacy Interrupt message generation is disabled 0: Legacy Interrupt message generation is enabled If this bit transitions from 1->0 when a previous Assert_INTx message was sent but no corresponding Deassert_INTx message sent yet, a Deassert_INTx message is sent on this bit transition. |
| 9 | RO | 0 | Fast Back-to-Back Enable Not applicable to PCI Express and is hardwired to 0 |
| 8 | RO | 0 | SERR Enable For PCI Express/ESI ports, this field enables notifying the internal core error logic of occurrence of an uncorrectable error (fatal or non-fatal) at the port. The internal core error logic of IOH then decides if/how to escalate the error further (pins/message etc.). This bit also controls the propagation of PCI Express ERR_FATAL and ERR_NONFATAL messages received from the port to the internal IOH core error logic. 1: Fatal and Non-fatal error generation and Fatal and Non-fatal error message forwarding is enabled 0: Fatal and Non-fatal error generation and Fatal and Non-fatal error message forwarding is disabled Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for details of how this bit is used in conjunction with other control bits in the Root Control register for forwarding errors detected on the PCI Express interface to the system core error logic. This bit has no impact on error reporting from the other devices - DMA, I/OxAPIC, Perf Mon and PCI Express DF* registers. |
| 7 | RO | 0 | IDSEL Stepping/Wait Cycle Control Not applicable to internal IOH devices. Hardwired to 0. |
| 6 | RO | 0 | Parity Error Response For PCI Express/ESI ports, IOH ignores this bit and always does ECC/parity checking and signaling for data/address of transactions both to and from IOH. This bit though affects the setting of bit 8 in the PCISTS (see bit 8 in Section 21.4.2.4) register. This bit has no impact on error reporting from the other devices - DMA, I/OxAPIC, Perf Mon and PCI Express DF* registers. |
| 5 | RO | 0 | VGA palette snoop Enable Not applicable to internal IOH devices. Hardwired to 0. |
| 4 | RO | 0 | Memory Write and Invalidate Enable Not applicable to internal IOH devices. Hardwired to 0. |
| 3 | RO | 0 | Special Cycle Enable Not applicable to PCI Express. Hardwired to 0. |



| Device: 19 Function: 0 Offset: 04h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 2 | RW | 0 | Bus Master Enable Controls the ability of the PCI Express/ESI port in generating/forwarding memory (including MSI writes) or I/O transactions (and not messages) or configuration transactions from the secondary side to the primary side. For DMA and I/OxAPIC, this bit enables them to generate memory write/MSI and memory read transactions (read applies only to DMA). 1: Enables the PCI Express/ESI port, I/OxAPIC or DMA to generate/forward memory, config or I/O read/write requests. 0: The Bus Master is disabled. When this bit is 0, IOH root ports will treat upstream PCI Express memory writes/reads, IO writes/reads, and configuration reads and writes as unsupported requests (and follow the rules for handling unsupported requests). This behavior is also true towards transactions that are already pending in the IOH root port's internal queues when the BME bit is turned off. I/OxAPIC and DMA cannot generate any memory transactions when this bit is 0. |
| 1 | RW | 0 | Memory Space Enable 1: Enables a PCI Express/ESI port's memory range registers, internal I/OxAPIC's MBAR register (ABAR range decode is not enabled by this bit) or CB DMA device's memory BARs to be decoded as valid target addresses for transactions from primary side. 0: Disables a PCI Express/ESI port's memory range registers (excluding the IOxAPIC range registers), internal I/OxAPIC's MBAR register (but not ABAR register) or DMA device's memory BARs to be decoded as valid target addresses for transactions from primary side. Note that if a PCI Express/ESI port's MSE bit is clear, that port can still be target of any memory transaction if subtractive decoding is enabled on that port. |
| 0 | RO | 0 | IO Space Enable Applies only to PCI Express/ESI ports 1: Enables the I/O address range, defined in the IOBASE and IOLIM registers of the PCI-to-PCI bridge header, for target decode from primary side 0: Disables the I/O address range, defined in the IOBASE and IOLIM registers of the PCI-to-PCI bridge header, for target decode from primary side Note that if a PCI Express/ESI port's IOSE bit is clear, that port can still be target of an I/O transaction if subtractive decoding is enabled on that port. |



21.5.2 PCIISTS: PCI Status Register (Dev #19)

The PCI Status register is a 16-bit status register that reports the occurrence of various events associated with the primary side of the “virtual” PCI-PCI bridge embedded in PCI Express ports and also primary side of the other devices on the internal IOH bus.

| Register: PCIISTS Device: 19 Function: 0 Offset: 06h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15 | RO | 0 | <p>Detected Parity Error</p> <p>This bit is set by a device when it receives a packet on the primary side with an uncorrectable data error (i.e. a packet with poison bit set or an uncorrectable data ECC error was detected at the XP-DP interface when ECC checking is done) or an uncorrectable address/control parity error. The setting of this bit is regardless of the Parity Error Response bit (PERRE) in the PCICMD register.</p> <p>This bit is RO for these devices.</p> |
| 14 | RO | 0 | <p>Signaled System Error</p> <p>1: The device reported fatal/non-fatal (and not correctable) errors it detected on its PCI Express interface. Software clears this bit by writing a '1' to it. For Express ports, this bit is also set (when SERR enable bit is set) when a FATAL/NON-FATAL message is forwarded from the Express link. Note that IOH internal 'core' errors (like parity error in the internal queues) are not reported via this bit.</p> <p>0: The device did not report a fatal/non-fatal error</p> |
| 13 | RO | 0 | <p>Received Master Abort</p> <p>This bit is set when a device experiences a master abort condition on a transaction it mastered on the primary interface (IOH internal bus). Note that certain errors might be detected right at the PCI Express interface and those transactions might not 'propagate' to the primary interface before the error is detected (e.g. accesses to memory above TOCM in cases where the PCIe interface logic itself might have visibility into TOCM). Such errors do not cause this bit to be set, and are reported via the PCI Express interface error bits (secondary status register). Conditions that cause bit 13 to be set, include:</p> <ul style="list-style-type: none"> • Device receives a completion on the primary interface (internal bus of IOH) with Unsupported Request or master abort completion Status. This includes UR status received on the primary side of a PCI Express port on peer-to-peer completions also. • Device accesses to holes in the main memory address region that are detected by the Intel QuickPath Interconnect source address decoder. • Other master abort conditions detected on the IOH internal bus. |
| 12 | RO | 0 | <p>Received Target Abort</p> <p>This bit is set when a device experiences a completer abort condition on a transaction it mastered on the primary interface (IOH internal bus). Note that certain errors might be detected right at the PCI Express interface and those transactions might not 'propagate' to the primary interface before the error is detected (e.g. accesses to memory above VTCSRBASE). Such errors do not cause this bit to be set, and are reported via the PCI Express interface error bits (secondary status register). Conditions that cause bit 12 to be set, include:</p> <ul style="list-style-type: none"> • Device receives a completion on the primary interface (internal bus of IOH) with completer abort completion Status. This includes CA status received on the primary side of a PCI Express port on peer-to-peer completions also. • Accesses to Intel QuickPath Interconnect that return a failed completion status • Other master abort conditions detected on the IOH internal bus. |
| 11 | RWC | 0 | <p>Signaled Target Abort</p> <p>This bit is set when a device signals a completer abort completion status on the primary side (internal bus of IOH). This condition includes a PCI Express port forwarding a completer abort status received on a completion from the secondary side and passed to the primary side on a peer2peer completion.</p> |



| Register: PCISTS Device: 19 Function: 0 Offset: 06h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 10:9 | RO | 0h | DEVSEL# Timing Not applicable to PCI Express. Hardwired to 0. |
| 8 | RO | 0 | Master Data Parity Error This bit is set by a device if the Parity Error Response bit in the PCI Command register is set and it receives a completion with poisoned data from the primary side or if it forwards a packet with data (including MSI writes) to the primary side with poison. |
| 7 | RO | 0 | Fast Back-to-Back Not applicable to PCI Express. Hardwired to 0. |
| 6 | RV | 0 | <i>Reserved</i> |
| 5 | RO | 0 | 66MHz capable Not applicable to PCI Express. Hardwired to 0. |
| 4 | RO | 1h | Capabilities List This bit indicates the presence of a capabilities list structure |
| 3 | RO | 0 | INTx Status Indicates that a legacy INTx interrupt condition is pending internally in the CB DMA device. This bit has meaning only in the legacy interrupt mode. This bit is always 0 when MSI-X (see) has been selected for DMA interrupts. Note that the setting of the INTx status bit is independent of the INTx enable bit in the PCI command register i.e. this bit is set anytime the DMA engine is setup by its driver to generate any interrupt and the condition that triggers the interrupt has occurred, regardless of whether a legacy interrupt message was signaled to the ICH or not. Note that the INTx enable bit has to be set in the PCICMD register for DMA to generate a INTx message to the ICH. This bit is not applicable to PCI Express and ESI ports and this bit does not get set for interrupts forwarded from a PCI Express port to the ICH from downstream devices. This bit also does not apply to Perf Mon, I/OxAPIC and DF* register devices. |
| 2:0 | RV | 0h | <i>Reserved</i> |

21.5.3 MBAR: IOxAPIC Base Address Register

| Register: MBAR Device: 19 Function: 0 Offset: 10h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 63:32 | RV | 0h | <i>Reserved</i> |
| 31:12 | RW | 0h | BAR: this marks the 4KB aligned 32-bit base address for memory-mapped registers of I/OxAPIC |
| 11:4 | RV | 0h | <i>Reserved</i> |
| 3 | RO | 0 | Prefetchable The IOxAPIC registers are not prefetchable. |
| 2:1 | RO | 00 | Type The IOAPIC registers can only be placed below 4G system address space. |
| 0 | RO | 0 | Memory Space This Base Address Register indicates memory space. |



21.5.4 ABAR: I/OxAPIC Alternate BAR

| Register: ABAR Device: 19 Function: 0 Offset: 40h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15 | RW | 0 | ABAR Enable: When set, the range FECX_YZ00 to FECX_YZFF is enabled as an alternate access method to the IOxAPIC registers and these addresses are claimed by the IOH's internal I/OxAPIC regardless of the setting the MSE bit in the I/OxAPIC config space. Bits 'XYZ' are defined below. |
| 14:12 | RV | 0 | Reserved |
| 11:8 | RW | 0 | XBAD: Base Address [19:16] These bits determine the high order bits of the I/O APIC address map. When a memory address is recognized by the IOH which matches FECX_YZ00-to-FECX_YZFF, the IOH will respond to the cycle and access the internal I/O APIC. |
| 7:4 | RW | 0h | YBAD: Base Address [15:12] These bits determine the low order bits of the I/O APIC address map. When a memory address is recognized by the IOH which matches FECX_YZ00-to-FECX_YZFF, the IOH will respond to the cycle and access the internal I/O APIC. |
| 3:0 | RW | 0h | ZBAD: Base Address [11:8] These bits determine the low order bits of the I/O APIC address map. When a memory address is recognized by the IOH which matches FECX_YZ00-to-FECX_YZFF, the IOH will respond to the cycle and access the internal I/O APIC. |

21.5.5 PMCAP: Power Management Capabilities Register

The PM Capabilities Register defines the capability ID, next pointer and other power management related support. The following PM registers /capabilities are added for software compliance.

| Register: PMCAP Device: 19 Function: 0 Offset: 6Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:27 | RO | 11001b | PME Support Bits 31, 30 and 27 must be set to '1' for PCI-PCI bridge structures representing ports on root complexes. |
| 26 | RO | 0 | D2 Support IOH does not support power management state D2. |
| 25 | RO | 0 | D1 Support IOH does not support power management state D1. |
| 24:22 | RO | 0h | AUX Current |
| 21 | RO | 0 | Device Specific Initialization |
| 20 | RV | 0 | Reserved |



| Register: PMCAP Device: 19 Function: 0 Offset: 6Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 19 | RO | 0 | PME Clock This field is hardwired to 0h as it does not apply to PCI Express. |
| 18:16 | RWO | 011 | Version This field is set to 3h (PM 1.2 compliant) as version number. Bit is RWO to make the version 2h incase legacy OS'es have any issues. |
| 15:8 | RO | 00h | Next Capability Pointer This is the last capability in the chain and hence set to 0. |
| 7:0 | RO | 01h | Capability ID Provides the PM capability ID assigned by PCI-SIG. |

21.5.6 PMCSR: Power Management Control and Status Register

This register provides status and control information for PM events in the PCI Express port of the IOH.

| Register: PMCSR Device: 19 Function: 0 Offset: 70h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:24 | RO | 00h | Data Not relevant for IOH |
| 23 | RO | 0h | Bus Power/Clock Control Enable This field is hardwired to 0h as it does not apply to PCI Express. |
| 22 | RO | 0h | B2/B3 Support This field is hardwired to 0h as it does not apply to PCI Express. |
| 21:16 | RV | 0h | <i>Reserved</i> |
| 15 | RO | 0h | PME Status Applies only to root ports This PME Status is a sticky bit. This bit is set, independent of the PMEEN bit defined below, on an enabled PCI Express hotplug event provided the root port was in D3hot state. Software clears this bit by writing a '1' when it has been completed. Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for further details on wake event generation at a root port. |
| 14:13 | RO | 0h | Data Scale Not relevant for IOH |
| 12:9 | RO | 0h | Data Select Not relevant for IOH |
| 8 | RO | 0h | PME Enable Applies only to root ports. This field is a sticky bit and when set, enables PMEs generated internally on a PCI Express hotplug event to set the appropriate bits in the ROOTSTS register (which can then trigger an MSI or cause a _PMEGPE event). |



| Register: PMCSR Device: 19 Function: 0 Offset: 70h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:4 | RV | 0h | <i>Reserved</i> |
| 3 | RO | 0 | Indicates IOH does not reset its registers when transitioning from D3hot to D0. |
| 2 | RV | 0h | <i>Reserved2</i> |
| 1:0 | RW | 0h | <p>Power State</p> <p>This 2-bit field is used to determine the current power state of the function and to set a new power state as well.</p> <p>00: D0 01: D1 (not supported by IOH) 10: D2 (not supported by IOH) 11: D3_hot</p> <p>If Software tries to write 01 or 10 to this field, the power state does not change from the existing power state (which is either D0 or D3hot) and nor do these bits 1:0 change value.</p> <p>All devices will</p> <ul style="list-style-type: none"> a) respond to only Type 0 configuration transactions targeted at the device's configuration space, when in D3hot state b) root port will not forward Type 1 or Type 0 transactions to the downstream PCIe link c) will not respond to memory/IO transactions (i.e. D3hot state is equivalent to MSE/IOSE bits being clear), with one exception noted below, as target d) will not generate any memory/IO/configuration transactions as initiator on the primary bus. <p>Exception to c) is that root ports will continue to decode and forward memory transactions that target the IOAPIC address range, even when the root port is in D3hot state.</p> <p>Inbound memory/IO/configuration transactions that happen when the device is in D3hot state are aborted and root ports return a UR response on PCIe. Messages/completions will still pass through in either direction without being aborted.</p> |

21.5.7 RDINDEX: Alternate Index to read Indirect I/OxAPIC Registers

| Register: RDINDEX Device: 19 Function: 0 Offset: 80h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RW | 0h | <p>Index: When bmc/jtag wants to read the indirect RTE registers of I/OxAPIC, this register is used to point to the index of the indirect register, as defined in the I/OxAPIC indirect memory space. Software writes to this register and then does a read of the RDWINDOW register to read the contents at that index.</p> <p>Note h/w does not preclude software from accessing this register over Intel QPI but that is not what this register is defined for.</p> |



21.5.8 RDWINDOW: Alternate Window to read Indirect I/O/APIC Registers

| Register: RDWINDOW Device: 19 Function: 0 Offset: 90h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:0 | RO | 0h | Window: When SMBUS/JTAG reads this register, the data contained in the indirect register pointed to by the RDINDEX register is returned on the read. |

21.5.9 IOAPICTETPC: IOAPIC Table Entry Target Programmable Control

| Register: IOAPICTETPC Device: 19 Function: 0 Offset: A0h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:13 | RV | 00000h | Reserved |
| 12 | RW | 1h | SRC17INTA: 0 – src/int is connected to IOAPIC table entry 5 1 – src/int is connected to IOAPIC table entry 21 |
| 11 | RW | 1h | SRC16INTB: 0 – src/int is connected to IOAPIC table entry 3 1 – src/int is connected to IOAPIC table entry 22 |
| 10 | RW | 1h | SRC13INTA: 0 – src/int is connected to IOAPIC table entry 2 1 – src/int is connected to IOAPIC table entry 21 |
| 9 | RW | 1h | SRC14INTD: 0 – src/int is connected to IOAPIC table entry 1 1 – src/int is connected to IOAPIC table entry 22 |
| 8 | RW | 1h | SRC10INTD: 0 – src/int is connected to IOAPIC table entry 16 1 – src/int is connected to IOAPIC table entry 21 |
| 7 | RW | 1h | SRC10INTC: 0 – src/int is connected to IOAPIC table entry 18 1 – src/int is connected to IOAPIC table entry 22 |
| 6 | RW | 0h | SRC10INTB: 0 – src/int is connected to IOAPIC table entry 7 1 – src/int is connected to IOAPIC table entry 17 |
| 5 | RW | 1h | SRC9INTC: 0 – src/int is connected to IOAPIC table entry 16 1 – src/int is connected to IOAPIC table entry 23 |
| 4 | RW | 0h | SRC6INTB: 0 – src/int is connected to IOAPIC table entry 17 1 – src/int is connected to IOAPIC table entry 23 |



| Register: IOAPICTETPC Device: 19 Function: 0 Offset: A0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 3 | RW | 1h | SRC5INTD: 0 – src/int is connected to IOAPIC table entry 18 1 – src/int is connected to IOAPIC table entry 23 |
| 2 | RW | 0h | SRC3INTD: 0 – src/int is connected to IOAPIC table entry 5 1 – src/int is connected to IOAPIC table entry 11 |
| 1 | RW | 0h | SRC3INTC: 0 – src/int is connected to IOAPIC table entry 3 1 – src/int is connected to IOAPIC table entry 10 |
| 0 | RW | 0h | SRC3INTB: 0 – src/int is connected to IOAPIC table entry 1 1 – src/int is connected to IOAPIC table entry 12 |

21.5.10 MBAR: IOxAPIC Base Address Register

| Register: MBAR Device: 19 Function: 0 Offset: 10h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 63:32 | RV | 0h | Reserved |
| 31:12 | RW | 0h | BAR: This marks the 4KB aligned 32-bit base address for memory-mapped registers of IOxAPIC |
| 11:4 | RV | 0h | Reserved |
| 3 | RO | 0 | Prefetchable The IOxAPIC registers are not prefetchable. |
| 2:1 | RO | 00 | Type The IOAPIC registers can only be placed below 4G system address space. |
| 0 | RO | 0 | Memory Space This Base Address Register indicates memory space. |



21.5.11 ABAR: I/OxAPIC Alternate BAR

| Register: ABAR Device: 19 Function: 0 Offset: 40h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15 | RW | 0 | ABAR Enable: When set, the range FECX_YZ00 to FECX_YZFF is enabled as an alternate access method to the IOxAPIC registers and these addresses are claimed by the IOH's internal I/OxAPIC regardless of the setting the MSE bit in the I/OxAPIC config space. Bits 'XYZ' are defined below. |
| 14:12 | RV | 0h | Reserved |
| 11:8 | RW | 0h | Base Address [19:16] (XBAD): These bits determine the high order bits of the I/O APIC address map. When a memory address is recognized by the IOH which matches FECX_YZ00-to-FECX_YZFF, the IOH will respond to the cycle and access the internal I/O APIC. |
| 7:4 | RW | 0h | Base Address [15:12] (YBAD): These bits determine the low order bits of the I/O APIC address map. When a memory address is recognized by the IOH which matches FECX_YZ00-to-FECX_YZFF, the IOH will respond to the cycle and access the internal I/O APIC. |
| 3:0 | RW | 0h | Base Address [11:8] (ZBAD): These bits determine the low order bits of the I/O APIC address map. When a memory address is recognized by the IOH which matches FECX_YZ00-to-FECX_YZFF, the IOH will respond to the cycle and access the internal I/O APIC. |

21.5.12 PMCAP: Power Management Capabilities Register

The PM Capabilities Register defines the capability ID, next pointer and other power management related support. The following PM registers /capabilities are added for software compliance.

| Register: PMCAP Device: 19 Function: 0 Offset: 6Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:27 | RO | 11001b | PME Support Bits 31, 30 and 27 must be set to '1' for PCI-PCI bridge structures representing ports on root complexes. |
| 26 | RO | 0 | D2 Support IOH does not support power management state D2. |
| 25 | RO | 0 | D1 Support IOH does not support power management state D1. |
| 24:22 | RO | 0h | AUX Current |
| 21 | RO | 0 | Device Specific Initialization |
| 20 | RV | 0 | <i>Reserved.</i> |



| Register: PMCAP Device: 19 Function: 0 Offset: 6Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 19 | RO | 0 | PME Clock This field is hardwired to 0h as it does not apply to PCI Express. |
| 18:16 | RWO | 011 | Version This field is set to 3h (PM 1.2 compliant) as version number. Bit is RWO to make the version 2h incase legacy OS'es have any issues. |
| 15:8 | RO | 00h | Next Capability Pointer This is the last capability in the chain and hence set to 0. |
| 7:0 | RO | 01h | Capability ID Provides the PM capability ID assigned by PCI-SIG. |

21.5.13 PMCSR: Power Management Control and Status Register

This register provides status and control information for PM events in the PCI Express port of the IOH.

| Register: PMCSR Device: 19 Function: 0 Offset: 70h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:24 | RO | 00h | Data Not relevant for IOH |
| 23 | RO | 0h | Bus Power/Clock Control Enable This field is hardwired to 0h as it does not apply to PCI Express. |
| 22 | RO | 0h | B2/B3 Support This field is hardwired to 0h as it does not apply to PCI Express. |
| 21:16 | RV | 0h | <i>Reserved.</i> |
| 15 | RO | 0h | PME Status Applies only to root ports This PME Status is a sticky bit. This bit is set, independent of the PMEEN bit defined below, on an enabled PCI Express hotplug event provided the root port was in D3hot state. Software clears this bit by writing a '1' when it has been completed. Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for further details on wake event generation at a root port. |
| 14:13 | RO | 0h | Data Scale Not relevant for IOH |
| 12:9 | RO | 0h | Data Select Not relevant for IOH |
| 8 | RO | 0h | PME Enable Applies only to root ports. This field is a sticky bit and when set, enables PMEs generated internally on a PCI Express hotplug event to set the appropriate bits in the ROOTSTS register (which can then trigger an MSI or cause a _PMEGPE event). |



| Register: PMCSR Device: 19 Function: 0 Offset: 70h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:4 | RV | 0h | <i>Reserved.</i> |
| 3 | RO | 0 | Indicates IOH does not reset its registers when transitioning from D3hot to D0. |
| 2 | RV | 0h | <i>Reserved.</i> |
| 1:0 | RW | 0h | <p>Power State This 2-bit field is used to determine the current power state of the function and to set a new power state as well.</p> <p>00: D0 01: D1 (not supported by IOH) 10: D2 (not supported by IOH) 11: D3_hot</p> <p>If Software tries to write 01 or 10 to this field, the power state does not change from the existing power state (which is either D0 or D3hot) and nor do these bits1:0 change value.</p> <p>All devices will</p> <ul style="list-style-type: none"> a) respond to only Type 0 configuration transactions targeted at the device's configuration space, when in D3hot state b) root port will not forward Type 1 or Type 0 transactions to the downstream PCIe link c) will not respond to memory/IO transactions (that is, D3hot state is equivalent to MSE/IOSE bits being clear), with one exception noted below, as target d) will not generate any memory/IO/configuration transactions as initiator on the primary bus. <p>Exception to c) is that root ports will continue to decode and forward memory transactions that target the IOAPIC address range, even when the root port is in D3hot state.</p> <p>Inbound memory/IO/configuration transactions that happen when the device is in D3hot state are aborted and root ports return a UR response on PCIe. Messages/completions will still pass through in either direction without being aborted.</p> |

21.5.14 RDINDEX: Alternate Index to read Indirect I/OxAPIC Registers

| Register: RDINDEX Device: 19 Function: 0 Offset: 80h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RW | 0h | <p>Index: When bmc/jtag wants to read the indirect RTE registers of I/OxAPIC, this register is used to point to the index of the indirect register, as defined in the I/OxAPIC indirect memory space. Software writes to this register and then does a read of the RDWINDOW register to read the contents at that index.</p> <p>Note hardware does not preclude software from accessing this register over Intel QuickPath Interconnect but that is not what this register is defined for.</p> |



21.5.15 RDWINDOW: Alternate Window to read Indirect I/O/APIC Registers

| Register: RDWINDOW Device: 19 Function: 0 Offset: 90h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:0 | RO | 0h | Window: When SMBUS/JTAG reads this register, the data contained in the indirect register pointed to by the RDINDEX register is returned on the read. |

21.5.16 IOAPICTETPC: IOxAPIC Table Entry Target Programmable Control

| Register: IOAPICTETPC Device: 19 Function: 0 Offset: A0h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:13 | RV | 00000h | Reserved |
| 12 | RW | 1h | SRC17INTA: 0 – src/int is connected to IOAPIC table entry 5 1 – src/int is connected to IOAPIC table entry 21 |
| 11 | RW | 1h | SRC16INTB: 0 – src/int is connected to IOAPIC table entry 3 1 – src/int is connected to IOAPIC table entry 22 |
| 10 | RW | 1h | SRC13INTA: 0 – src/int is connected to IOAPIC table entry 2 1 – src/int is connected to IOAPIC table entry 21 |
| 9 | RW | 1h | SRC14INTA: 0 – src/int is connected to IOAPIC table entry 1 1 – src/int is connected to IOAPIC table entry 22 |
| 8 | RW | 1h | SRC10INTD: 0 – src/int is connected to IOAPIC table entry 16 1 – src/int is connected to IOAPIC table entry 21 |
| 7 | RW | 1h | SRC10INTC: 0 – src/int is connected to IOAPIC table entry 18 1 – src/int is connected to IOAPIC table entry 22 |
| 6 | RW | 0h | SRC10INTB: 0 – src/int is connected to IOAPIC table entry 7 1 – src/int is connected to IOAPIC table entry 17 |
| 5 | RW | 1h | SRC9INTC: 0 – src/int is connected to IOAPIC table entry 16 1 – src/int is connected to IOAPIC table entry 23 |
| 4 | RW | 0h | SRC6INTB: 0 – src/int is connected to IOAPIC table entry 17 1 – src/int is connected to IOAPIC table entry 23 |



| Register: IOAPICTETPC Device: 19 Function: 0 Offset: A0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 3 | RW | 1h | SRC5INTD: 0 – src/int is connected to IOAPIC table entry 18 1 – src/int is connected to IOAPIC table entry 23 |
| 2 | RW | 0h | SRC3INTD: 0 – src/int is connected to IOAPIC table entry 5 1 – src/int is connected to IOAPIC table entry 11 |
| 1 | RW | 0h | SRC3INTC: 0 – src/int is connected to IOAPIC table entry 3 1 – src/int is connected to IOAPIC table entry 10 |
| 0 | RW | 0h | SRC3INTB: 0 – src/int is connected to IOAPIC table entry 1 1 – src/int is connected to IOAPIC table entry 12 |

21.5.17 I/OxAPIC Memory Mapped Registers

I/OxAPIC has a direct memory mapped space. An index/data register pair is located within the directed memory mapped region and is used to access the redirection table entries. provides the direct memory mapped registers of the I/OxAPIC. The offsets shown in the table are from the base address in either ABAR or MBAR or both. Accesses to addresses beyond 40h return all 0s.

Note that only addresses up to offset 0xFF can be accessed via the ABAR register whereas offsets up to 0xFFF can be accessed via MBAR. Only aligned DWORD reads and write are allowed towards the I/OxAPIC memory space. Any other accesses will result in an error.



Table 21-5. I/OxAPIC Direct Memory Mapped Registers

| Register | Byte Offset |
|----------|-------------|
| Index | 00h |
| | 04h |
| | 08h |
| | 0Ch |
| Window | 10h |
| | 14h |
| | 18h |
| | 1Ch |
| PAR | 20h |
| | 24h |
| | 28h |
| | 2Ch |
| | 30h |
| | 34h |
| | 38h |
| | 3Ch |
| EOI | 40h |
| | ... |
| | FF |
| | ... |
| | FFF |

21.5.18 Index Register

The Index Register will select which indirect register appears in the window register to be manipulated by software. Software will program this register to select the desired APIC internal register.

| Register: Index BAR: XBAR Offset: 00h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RW | 0 | Index (IDX): Indirect register to access. |



21.5.19 Window Register

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 be accessed in byte quantities.

| Register: Window BAR: MBAR Offset: 10h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:0 | RW | 0 | Window (WND) : Data to be written to the indirect register on writes, and location of read data from the indirect register on reads. |

21.5.20 PAR Register

| Register: PAR BAR: MBAR Offset: 20h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RO | 0 | Assertion (PAR) : IOH does not allow writes to the PAR to cause MSI interrupts. |

21.5.21 EOI Register

| Register: EOI BAR: MBAR Offset: 40h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RW | 0 | EOI : The EOI register is present to provide a mechanism to efficiently convert level interrupts to edge triggered MSI interrupts. When a write is issued to this register, the I/O(x)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 that if multiple I/O Redirection entries, for any reason, assign the same vector, each of those entries will have the Remote_IRR bit reset to '0'. This will cause the corresponding I/OxAPIC entries to resample their level interrupt inputs and if they are still asserted, cause more MSI interrupt(s) (if unmasked) which will again set the Remote_IRR bit. |



Table 21-6. I/OxAPIC Indexed Registers (Redirection Table Entries)

| Indexed Register | Index |
|------------------|-------|
| APICID | 00h |
| Version | 01h |
| ARBID | 02h |
| BCFG | 03h |
| | ... |
| | ... |
| RTL0 | 10h |
| RTH0 | 11h |
| RTL1 | 12h |
| RTH1 | 13h |
| | ... |
| | ... |
| | ... |
| | ... |
| RTL23 | 3Eh |
| RTH23 | 3Fh |
| | 40h |
| | ... |
| | FFh |

21.5.22 APICID

This register uniquely identifies an APIC in the system. This register is not used by OS'es anymore and is still implemented in hardware because of FUD.

| Register: APICID BAR: MBAR Offset: 10h IA: 00h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:28 | RV | 0 | Reserved |
| 27:24 | RW | 0 | APICID: Allows for up to 16 unique APIC IDs in the system. |
| 23:0 | RV | 0 | Reserved |

21.5.23 Version

This register uniquely identifies an APIC in the system. This register is not used by OS'es anymore and is still implemented in hardware because of FUD.

| Register: Version BAR: MBAR Offset: 10h IA: 01h | | | |
|--|------|---------|-------------|
| Bit | Attr | Default | Description |
| 31:24 | RV | 0 | Reserved |



| Register: Version BAR: MBAR Offset: 10h IA: 01h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 23:16 | RO | 17h | Maximum Redirection Entries (MAX): This is the entry number of the highest entry in the redirection table. It is equal to the number of interrupt inputs minus one. This field is hardwired to 17h to indicate 24 interrupts. |
| 15 | RO | 0 | IRQ Assertion Register Supported (PRQ): This bit is set to 0 to indicate that this version of the I/OxAPIC does not implement the IRQ Assertion register and does not allow PCI devices to write to it to cause interrupts. |
| 14:8 | RV | 0 | Reserved |
| 7:0 | RO | 20h | Version (VS): This identifies the implementation version. This field is hardwired to 20h indicate this is an I/OxAPIC. |

21.5.24 ARBID

This is a legacy register carried over from days of serial bus interrupt delivery. This register has no meaning in IOH. It just tracks the APICID register for compatibility reasons.

| Register: ARBUID BAR: MBAR Offset: 10h IA: 02h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:28 | RV | 0 | Reserved |
| 27:24 | RO | 0 | Arbitration ID: Just tracks the APICID register. |
| 23:0 | RV | 0 | Reserved |

21.5.25 BCFG

| Register: BCFG BAR: MBAR Offset: 10h IA: 03h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:1 | RV | 0 | Reserved |
| 0 | RW | 1 | Boot Configuration: This bit is a default1 to indicate FSB delivery mode. A value of 0 has no effect. Its left as RW for software compatibility reasons. |

21.5.26 RTL[0:23]: Redirection Table Low DWORD

The information in this register along with Redirection Table High DWORD register is used to construct the MSI interrupt. There is one of these pairs of registers for every interrupt. The first interrupt has the redirection registers at offset 10h. The second interrupt at 12h, third at 14h, and so on, until the final interrupt (interrupt 23) at 3Eh.



| Register: RTL[0:23] BAR: MBAR Offset: 10h IA: 10h- 3Eh by 2 | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:18 | RV | 0 | Reserved |
| 17 | RW | 0 | Disable Flushing: This bit has no meaning in IOH. This bit is R/W for software compatibility reasons only |
| 16 | RW | 1 | <p>Mask (MSK):</p> <p>When cleared, an edge assertion or level (depending on bit 15 in this register) on the corresponding interrupt input results in delivery of an MSI interrupt using the contents of the corresponding redirection table high/low entry.</p> <p>When set, an edge or level on the corresponding interrupt input does not cause MSI Interrupts and no MSI interrupts are held pending as well (that is, if an edge interrupt asserted when the mask bit is set, no MSI interrupt is sent and the hardware does not remember the event to cause an MSI later when the mask is cleared). When set, assertion/deassertion of the corresponding interrupt input causes Assert/Deassert_INTx messages to be sent to the legacy ICH, provided the 'Disable PCI INTx Routing to ICH' bit is clear. If the latter is set, Assert/Deassert_INTx messages are not sent to the legacy ICH.</p> <p>When mask bit goes from 1 to 0 for an entry and the entry is programmed for level input, the input is sampled and if asserted, an MSI is sent. Also, if an Assert_INTx message was previously sent to the legacy ICH/internal-coalescing logic on behalf of the entry, when the mask bit is clear, then a Deassert_INTx event is scheduled on behalf of the entry (whether this event results in a Deassert_INTx message to the legacy ICH depends on whether there were other outstanding Deassert_INTx messages from other sources). When the mask bit goes from 0 to 1, and the corresponding interrupt input is already asserted, an Assert_INTx event is scheduled on behalf of the entry. Note though that if the interrupt is deasserted when the bit transitions from 0 to 1, a Deassert_INTx is not scheduled on behalf of the entry.</p> |
| 15 | RW | 0 | Trigger Mode (TM): This field indicates the type of signal on the interrupt input that triggers an interrupt. 0 indicates edge sensitive, 1 indicates level sensitive. |
| 14 | RO | 0 | Remote IRR (RIRR): This bit is used for level triggered interrupts; its meaning is undefined for edge triggered interrupts. For level triggered interrupts, this bit is set when an MSI interrupt has been issued by the I/OxAPIC into the system fabric (noting that if BME bit is clear or when the mask bit is set, no new MSI interrupts cannot be generated and this bit cannot transition from 0 to 1 in those conditions). It is reset (if set) when an EOI message is received from a local APIC with the appropriate vector number, at which time the level interrupt input corresponding to the entry is resampled causing one more MSI interrupt (if other enable bits are set) and causing this bit to be set again. |
| 13 | RW | 0 | <p>Interrupt Input Pin Polarity (IP): 0=active high; 1=active low.</p> <p>Strictly, speaking this bit has no meaning in IOH since the Assert/Deassert_INTx messages are level in-sensitive. But the core I/OxAPIC logic that is reused from PXH might be built to use this bit to determine the correct polarity. Most OS'es today support only active low interrupt inputs for PCI devices. Given that, the OS is expected to program a 1 into this register and so the "internal" virtual wire signals in the IOH need to be active low, that is, 0=asserted and 1=deasserted.</p> |
| 12 | RO | 0 | Delivery Status: When trigger mode is set to level and the entry is <u>unmasked</u> , this bit indicates the state of the level interrupt, that is, 1b if interrupt is asserted else 0b. When the trigger mode is set to level but the entry is <u>masked</u> , this bit is always 0b. This bit is always 0b when trigger mode is set to edge. |
| 11 | RW | 0 | <p>Destination Mode (DSTM):</p> <p>0 – Physical 1 – Logical</p> |



| Register: RTL[0:23] BAR: MBAR Offset: 10h IA: 10h- 3Eh by 2 | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 10:8 | RW | 0 | Delivery Mode (DELM): This field specifies how the APICs listed in the destination field should act upon reception of the interrupt. Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode. The encodings are: 000 – Fixed: Trigger Mode can be edge or level. Examine TM bit to determine. 001 – Lowest Priority: Trigger Mode can be edge or level. Examine TM bit to determine. 010 – SMI/PMI: Trigger mode is always edge and TM bit is ignored. 011 – Reserved 100 – NMI. Trigger mode is always edge and TM bit is ignored. 101 – INIT. Trigger mode is always edge and TM bit is ignored. 110 – Reserved 111 – ExtINT. Trigger mode is always edge and TM bit is ignored. |
| 7:0 | RW | 0 | Vector (VCT): This field contains the interrupt vector for this interrupt |

21.5.27 RTH[0:23]: Redirection Table High DWORD

| Register: RTH[0:23] BAR: MBAR Offset: 10h IA: 11h - 3h by 2 | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:24 | RW | 0h | Destination ID (DID): They are bits [19:12] of the MSI address. |
| 23:16 | RW | 0h | Extended Destination ID (EDID): These bits become bits [11:4] of the MSI address. |
| 15:00 | RV | 0h | Reserved |



21.6 Intel® VT, Address Mapping, System Management, Device Hide, Misc

The offsets shown in Table 21-4 are offsets from the base of the CSR region. Any change to these registers under that event can only happen during an Intel QuickPath Interconnect quiescence flow. Any exceptions will be called out when appropriate.

Table 21-7. Core Registers (Dev 20, Function 0) - Offset 0x00-0xFF

| | | | | |
|---------|---------------------|-----|---------------------|-----|
| DID | VID | 00h | | 80h |
| PCISTS | PCICMD | 04h | | 84h |
| CCR | RID | 08h | GENPROTRANGE0.BASE | 88h |
| HDR | CLS | 0Ch | | 8Ch |
| | | 10h | GENPROTRANGE0.LIMIT | 90h |
| | | 14h | | 94h |
| | | 18h | IOHMISCCTRL | 98h |
| | | 1Ch | IOHMISCSS | 9Ch |
| | | 20h | | A0h |
| | | 24h | | A4h |
| | | 28h | TSEGCTRL | A8h |
| SID | SVID | 2Ch | | ACH |
| | | 30h | GENPROTRANGE1.BASE | B0h |
| | CAPPTR ¹ | 34h | | B4h |
| | | 38h | GENPROTRANGE1.LIMIT | B8h |
| | INTP | 3Ch | | BCh |
| | INTL | | | |
| EXPCAPS | EXPNPTR | 40h | GENPROTRANGE2.BASE | C0h |
| | EXPCAPID | | | |
| | DEVCAP | 44h | | C4h |
| DEVSTS | DEVCTRL | 48h | GENPROTRANGE2.LIMIT | C8h |
| | | 4Ch | | CCh |
| | | 50h | TOLM | D0h |
| | | 54h | TOHM | D4h |
| | | 58h | | D8h |
| | | 5Ch | NCMEM.BASE | DCh |
| | | 60h | | E0h |
| | | 64h | NCMEM.LIMIT | E4h |
| | | 68h | | E8h |
| | | 6Ch | | ECh |
| | | 70h | DEVHIDE 1 | F0h |
| | | 74h | | F4h |
| | | 78h | DEVHIDE 2 | F8h |
| | | 7Ch | | FCh |

Notes:

1. CAPPTR points to the first capability block



Table 21-8. Core Registers (Dev 20, Function 0)

| | | | | | | | | |
|--------------------------------|--------------|-------------------------|------------------|------|------------|-----------|-------------|------|
| Reserved for PCIe header space | | | | 100h | VTBAR | | 180h | |
| | | | | 104h | | VTGENCTRL | 184h | |
| | IOHBUSN O | LIO.LIMIT | LIO.BASE | 108h | | | 188h | |
| LMMIOL.LIMIT | | LMMIOL.BASE | | 10Ch | VTGENCTRL2 | | 18Ch | |
| LMMIOH.LIMIT | | LMMIOH.BASE | | 110h | VTSTS | | 190h | |
| LMMIOH.BASEU | | | | 114h | | | 194h | |
| LMMIOH.LIMITU | | | | 118h | | | 198h | |
| | | LCFGBUS.L IMIT | LCFGBUS.B ASE | 11Ch | | | 19Ch | |
| | | GIO.LIMIT | GIO.BASE | 120h | | | 1A0h | |
| GMMIOL.LIMIT | | GMMIOL.BASE | | 124h | | | 1A4h | |
| GMMIOH.LIMIT | | GMMIOH.BASE | | 128h | | | VTUNCERRSTS | 1A8h |
| GMMIOH.BASEU | | | | 12Ch | | | VTUNCERRMSK | 1ACh |
| GMMIOH.LIMITU | | | | 130h | | | VTUNCERRSEV | 1B0h |
| | | GCFGBUS.L IMIT | GCFGBUS. BASE | 134h | | | VTUNCERRPTR | 1B4h |
| | | | | 138h | | | 1B8h | |
| | | | | 13Ch | 1BCh | | | |
| | | | | 140h | 1C0h | | | |
| | | | | 144h | 1C4h | | | |
| DUALIOAPIC.ABAR LIMIT | | DUALIOAPIC.ABAR BASE | | 148h | | 1C8h | | |
| | | | | 14Ch | | 1CCh | | |
| | | | | 150h | | 1D0h | | |
| | | | | 154h | | 1D4h | | |
| | | | | 158h | | 1D8h | | |
| | | | | 15Ch | | 1DCh | | |
| | | | | 160h | | 1E0h | | |
| | | | | 164h | | 1E4h | | |
| | | | | 168h | | 1E8h | | |
| | | | | 16Ch | | 1ECh | | |
| | | | | 170h | | 1F0h | | |
| 174h | | 1F4h | | | | | | |
| 178h | | 1F8h | | | | | | |
| 17Ch | | 1FCh | | | | | | |



21.6.1 GENPROTRANGE0.BASE: Generic Protected Memory Range 0 Base Address Register

| Register: GENPROTRANGE0.BASE Device: 20 Function: 0 Offset: 88h | | | |
|--|------|--------------|---|
| Bit | Attr | Default | Description |
| 63:51 | RO | 0 | Correspond to address A[63:51] of the protected range and is always 0. |
| 50:16 | RWLB | 7_FFFF_FFFFh | <p>Base address [50:16] of generic memory address range that needs to be protected from inbound dma accesses. The protected memory range can be anywhere in the memory space addressable by the processor. Addresses that fall in this range i.e. GenProtRange.Base[63:16] <= Address [63:16] <= GenProtRange.Limit [63:16], are completely aborted by IOH.</p> <p>Setting the Protected range base address greater than the limit address disables the protected memory region.</p> <p>Note that this range is orthogonal to VT-d spec defined protected address range.</p> <p>This register is programmed once at boot time and does not change after that, including any quiesce flows.</p> |
| 15:0 | RV | 0 | Reserved |

21.6.2 GENPROTRANGE0.LIMIT: Generic Protected Memory Range 0 Limit Address Register

| Register: GENPROTRANGE0.LIMIT Device: 20 Function: 0 Offset: 90h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 63:51 | RO | 0 | Correspond to address A[63:51] of the protected range and is always 0. |
| 50:16 | RWLB | 0 | <p>Limit address [50:16] of generic memory address range that needs to be protected from inbound dma accesses. The protected memory range can be anywhere in the memory space addressable by the processor. Addresses that fall in this range i.e. GenProtRange.Base[63:16] <= Address [63:16] <= GenProtRange.Limit [63:16], are completely aborted by IOH.</p> <p>Setting the Protected range base address greater than the limit address disables the protected memory region.</p> <p>Note that this range is orthogonal to VT-d spec defined protected address range.</p> <p>This register is programmed once at boot time and does not change after that, including any quiesce flows.</p> |
| 15:0 | RV | 0 | Reserved |



21.6.3 IOHMISCCTRL: IOH MISC Control Register

| Register: IOHMISCCTRL Device: 20 Function: 0 Offset: 98h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:14 | RV | 0 | Reserved |
| 13 | RW | 0 | CPUCSR_IB_Abort: This bit controls if inbound access to CPUCSR range is aborted. 0 - IB access to CPUCSR range is enabled, that is, allowed. 1 - IB access to CPUCSR range is disabled, that is, disallowed. |
| 12 | RW | 0 | Lock Thawing Mode. Mode controls how inbound queues in the south agents (PCIe, ESI) thaw when they are target of a locked read. 0 - Thaw only posted requests 1 - Thaw posted and non-posted requests. |
| 11:10 | RW | strap | SUBDECEN Indicates the port that provides the subtractive decode path for inbound and outbound decode. 00 - ESI 01 - Reserved 10 - Reserved 11 - Intel QuickPath Interconnect When this points to ESI, all address ranges in the peer-to-peer config space of the port are ignored for address decode purposes. When this field is 00, then the IOH is the legacy IOH. The strap referred to in the default field is the LEGACYIOH strap. |
| 9 | RV | 0 | Reserved |
| 8 | RW | 0 | TOCMVALID: This bit is set by software after it has initialized the TOCM register with the right value. IOH decoder uses this bit to determine if bits from 32 to TOCM are to be decoded towards privileged CSR space. |
| 7:3 | RW | 01001 | TOCM Indicates the top of Intel QuickPath Interconnect physical addressability limit. 00000-00100: Reserved 00101: 2 ³⁷ 00110: 2 ³⁸ ... 10011: 2 ⁵¹ 10100 -11111: Reserve IOH uses this to abort all inbound transactions that cross this limit. |
| 2 | RW | 0 | EN1K This bit when set, enables 1K granularity for I/O space decode in each of the virtual P2P bridges corresponding to root ports, and ESI ports. |
| 1:0 | RV | 0 | Reserved |



21.6.4 IOHMISCSS: IOH MISC Status

| Register: IOHMISCSS Device: 20 Function: 0 Offset: 9Ch | | | |
|---|------|--------------------------------|--|
| Bit | Attr | Default | Description |
| 31:30 | RV | 0 | <i>Reserved</i> |
| 29 | RO | Strap: NODEID2 | NodeID2 - Node ID 2 |
| 28 | RO | Strap: DualIOH_QP IPTSEL | Intel QuickPath Interconnect/NodeID3 - For dual node configurations, indicates which Intel QuickPath Interconnect port is connected to the other IOH. For all configurations, this strap indicates NID3 value. |
| 27 | RV | 0 | <i>Reserved</i> |
| 26 | RO | strap: TESTHI[4] | Must be pulled high. |
| 25 | RO | strap: DUALIOH | DUALIOH - Indicates dual IOH configuration |
| 24:18 | RV | 0 | <i>Reserved</i> |
| 17 | RO | strap: LEGACYIOH | Legacy IOH. |
| 16 | RO | strap: TESTHI1 | Firmware Agent. Legacy IOHs are advertised as a firmware agent. |
| 15:10 | RO | strap: PCEWIDTH[5: :0] | PCIe Link width select |
| 9:8 | RO | strap:DDRFR EQ[3:2] | DDR Frequency selection |
| 7:6 | RV | 0 | <i>Reserved</i> |
| 5 | RO | strap: PESBLCSEL | PESBLCSEL |
| 4 | RO | strap: QPISBLCSEL | QPISBLCSEL |
| 3 | RV | 0 | <i>Reserved</i> |
| 2 | RO | strap: SMBUSID | SMBUSID |
| 1:0 | RO | strap: QPIFREQSEL [1:0] | QPIFREQSEL[1:0] |



21.6.5 IOH System Management Registers

21.6.5.1 TSEGCTRL: TSeg Control Register

The location of the TSeg region, size, and enable/disable control.

| Register: TSEGCTRL Device: 20 Function: 0 Offset: A8h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:20 | RWO | FE0h | TBA: TSeg Base Address Indicates the base address which is aligned to a 1MB boundary. Bits [31:20] corresponds to A[31:20] address bits. |
| 19:4 | RV | 0 | <i>Reserved</i> |
| 3:1 | RWO | 100 | TSEG_SIZE: Size of TSeg 000: 512KB 001: 1 MB 010: 2 MB 011: 4MB 100: 8 MB 101-111: <i>Reserved</i> |
| 0 | RWO | 1 | TSEG_EN: TSeg Enabling Control 0: Disabling the TSeg in IOH. 1: Enabling the TSeg in IOH for IB access check. |

21.6.5.2 GENPROTRANGE.BASE1: Generic Protected Memory Range 1 Base Address Register

| Register: GENPROTRANGE.BASE1 Device: 20 Function: 0 Offset: B0h | | | |
|--|------|--------------|--|
| Bit | Attr | Default | Description |
| 63:51 | RO | 0 | Correspond to address A[63:51] of the protected range and is always 0. |
| 50:16 | RW | 7_FFFF-FFFFh | Base address [50:16] of generic memory address range. The protected memory range can be anywhere in the memory space addressable by the processor. Addresses that fall in this range, that is, GenProtRange.Base[63:16] <= Address [63:16] <= GenProtRange.Limit [63:19], are completely aborted by IOH. Setting the Protected range base address greater than the limit address disables the protected memory region. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 15:0 | RV | 0 | <i>Reserved</i> |



21.6.5.3 GENPROTRANGE1.LIMIT: Generic Protected Memory Range 1 Limit Address Register

| Register: GENPROTRANGE1.LIMIT Device: 20 Function: 0 Offset: B8h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 63:51 | RO | 0 | Correspond to address A[63:51] of the protected range and is always 0. |
| 50:16 | RWLB | 0 | <p>Limit address [50:16] of generic memory address. The protected memory range can be anywhere in the memory space addressable by the processor. Addresses that fall in this range, that is, GenProtRange.Base[63:16] <= Address [63:16] <= GenProtRange.Limit [63:19], are completely aborted by IOH.</p> <p>Setting the Protected range base address greater than the limit address disables the protected memory region.</p> <p>This register is programmed once at boot time and does not change after that, including any quiesce flows.</p> |
| 15:0 | RV | 0 | Reserved |

21.6.5.4 GENPROTRANGE2.BASE: Generic Protected Memory Range 2 Base Address Register

| Register: GENPROTRANGE2.BASE Device: 20 Function: 0 Offset: C0h | | | |
|--|------|---------------|---|
| Bit | Attr | Default | Description |
| 63:51 | RO | 0 | Correspond to address A[63:51] of the protected range and is always 0. |
| 50:16 | RWLB | 7_FFFF_FFF Fh | <p>Base address [50:19] of generic memory address range. The protected memory range can be anywhere in the memory space addressable by the processor. Addresses that fall in this range, that is, GenProtRange.Base[63:16] <= Address [63:16] <= GenProtRange.Limit [63:19], are completely aborted by IOH.</p> <p>Setting the Protected range base address greater than the limit address disables the protected memory region.</p> <p>Note that this range is orthogonal to Intel VT-d spec defined protected address range.</p> <p>This register is programmed once at boot time and does not change after that, including any quiesce flows.</p> |
| 15:0 | RV | 0 | Reserved |



21.6.5.5 GENPROTRANGE2.LIMIT: Generic Protected Memory Range 2 Limit Address Register

| Register: GENPROTRANGE.LIMIT Device: 20 Function: 0 Offset: C8h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 63:51 | RO | 0 | Correspond to address A[63:51] of the protected range and is always 0. |
| 50:16 | RWLB | 0 | Limit address [50:16] of generic memory address range. The protected memory range can be anywhere in the memory space addressable by the processor. Addresses that fall in this range, that is, GenProtRange.Base[63:16] <= Address [63:16] <= GenProtRange.Limit [63:19], are completely aborted by IOH. Setting the Protected range base address greater than the limit address disables the protected memory region. Note that this range is orthogonal to Intel VT-d spec defined protected address range. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 15:0 | RV | 0 | Reserved |

21.6.5.6 TOLM: Top of Low Memory

Top of low memory. Note that bottom of low memory is assumed to be 0.

| Register: TOLM Device: 20 Function: 0 Offset: D0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:26 | RWLB | 0 | TOLM address Indicates the top of low dram memory which is aligned to a 64MB boundary. A 32 bit transaction that satisfies '0 <= A[31:26] <= TOLM[31:26]' is a transaction towards main memory. |
| 25:0 | RV | 0 | Reserved |

21.6.5.7 TOHM: Top of High Memory

Top of high memory. Note that bottom of high memory is fixed at 4 GB.

| Register: TOHM Device: 20 Function: 0 Offset: D4h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 63:26 | RWLB | 0 | TOHM address Indicates the limit of an aligned 64 MB granular region that decodes > 4 GB addresses towards system dram memory. A 64-bit transaction that satisfies '4G <= A[63:26] <= TOHM[63:26]' is a transaction towards main memory. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 25:0 | RV | 0 | Reserved |



21.6.5.8 NCMEM.BASE: NCMEM Base

Base address of Intel® QuickPath Interconnect non-coherent memory.

| Register: NCMEM.BASE Device: 20 Function: 0 Offset: DCh | | | |
|--|------|-------------------|--|
| Bit | Attr | Default | Description |
| 63:26 | RWLB | 3F_FFFF_FFF Fh | Non Coherent memory base address Describes the base address of a 64MB aligned dram memory region on Intel QuickPath Interconnect that is non-coherent. Address bits [63:26] of an inbound address if it satisfies 'NcMem.Base[63:26] <= A[63:26] <= NcMem.Limit[63:26]' is considered to be towards the non-coherent Intel QuickPath Interconnect memory region. Its expected that the range indicated by the Non-coherent memory base and limit registers is a subset of either the low dram or high dram memory regions as described via the corresponding base and limit registers. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 25:0 | RV | 0 | Reserved |

21.6.5.9 NCMEM.LIMIT: NCMEM Limit

Limit of Intel QuickPath Interconnect non-coherent memory.

| Register: NCMEM.LIMIT Device: 20 Function: 0 Offset: E4h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 63:26 | RW | 0 | Non Coherent memory limit address Describes the limit address of a 64MB aligned dram memory region on Intel QuickPath Interconnect that is non-coherent. Address bits [63:26] of an inbound address if it satisfies 'NcMem.Base[63:26] <= A[63:26] <= NcMem.Limit[63:26]' is considered to be towards the non-coherent Intel® QuickPath Interconnect memory region. Its expected that the range indicated by the Non-coherent memory base and limit registers is a subset of either the low dram or high dram memory regions as described via the corresponding base and limit registers. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 25:0 | RV | 0 | Reserved |



21.6.5.10 DEVHIDE1: Device Hide 1 Register

This register provides a method to hide the PCI config space of devices inside IOH, from the host initiated configuration accesses. This register has no impact on configuration accesses from SMBUS/JTAG ports of IOH. When set, all PCI configuration accesses from Intel QPI targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed.

| Register: DEVHIDE1 Device: 20 Function: 0 Offset: F0h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31 | RWLB | 0 | Hide_Dev18_fun1: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 30 | RWLB | 0 | Hide_Dev18_fun0: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 29 | RWLB | 0 | Hide_Dev17_fun1: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 28 | RWLB | 0 | Hide_Dev17_fun0: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 27 | RWLB | 0 | Hide_Dev16_fun1: When set, all PCI configuration accesses from Intel® QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |



| Register: DEVHIDE1 Device: 20 Function: 0 Offset: F0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 26 | RWLB | 0 | Hide_Dev16_fun0: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 25 | RWLB | 0 | Hide_Dev15_fun0: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 24 | RWLB | 0 | Hide_Dev14_fun3: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 23 | RWLB | 0 | Hide_Dev14_fun2: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 22 | RWLB | 0 | Hide_Dev14_fun1: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 21 | RWLB | 0 | Hide_Dev14_fun0: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |



| Register: DEVHIDE1 Device: 20 Function: 0 Offset: F0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 20 | RWLB | 0 | Hide_Dev13_fun7: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 19 | RWLB | 0 | Hide_Dev13_fun6: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 18 | RWLB | 0 | Hide_Dev13_fun5: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 17 | RWLB | 0 | Hide_Dev13_fun4: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 16 | RWLB | 0 | Hide_Dev13_fun3: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 15 | RWLB | 0 | Hide_Dev13_fun2: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |



| Register: DEVHIDE1 Device: 20 Function: 0 Offset: F0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 14 | RWLB | 0 | Hide_Dev13_fun1: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 13 | RWLB | 0 | Hide_Dev13_fun0: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 12 | RWLB | 0 | Hide_Dev20_fun3: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 11 | RWLB | 0 | Hide_Dev14_fun4: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 10 | RWLB | 0 | Hide_Dev10: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 9 | RWLB | 0 | Hide_Dev9: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |



| Register: DEVHIDE1 Device: 20 Function: 0 Offset: F0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 8 | RWLB | 0 | Hide_Dev8: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 7 | RWLB | 0 | Hide_Dev7: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 6 | RWLB | 0 | Hide_Dev6: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 5 | RWLB | 0 | Hide_Dev5: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 4 | RWL | 0 | Hide_Dev8: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |



| Register: DEVHIDE1 Device: 20 Function: 0 Offset: F0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 3 | RWL | 0 | Hide_Dev3: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 2 | RWL | 0 | Hide_Dev2: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 1 | RWL | 0 | Hide_Dev1: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |
| 0 | RWL | 0 | Hide_Dev0: When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed. If software hides function#0 in either device 16, or 17, it needs to hide all functions within that device to comply with PCI rules. This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space. |



21.6.5.11 DEVHIDE2: Device Hide 2 Register

This register provides a method to hide the PCI config space of devices inside IOH, from the host initiated configuration accesses. This register has no impact on configuration accesses from SMBus/JTAG ports of an IOH.

| Register: DEVHIDE2 Device: 20 Function: 0 Offset: F8h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:15 | RV | 0 | Reserved |
| 14:7 | RWLB | 0 | <p>Device/Function Hide: Bit 7 corresponds to Device#22/Function#0, bit 8 corresponds to Device#22/Function#1, ..., bit 14 corresponds to Device#22/Function#7.</p> <p>When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed.</p> <p>If software hides function#0 in either device 22, it needs to hide all functions within that device to comply with PCI rules.</p> <p>This bit has no effect on SMBUS and JTAG initiated accesses to corresponding device's config space</p> |
| 6:0 | RWLB | 0 | <p>Device/Function Hide: Bit 0 corresponds to Device#18/Function#2, bit 1 corresponds to Device#18/Function#3, Bit 2 corresponds to Device#19/Function#0, bit 3 corresponds to Device#20/Function#0, Bit 4 corresponds to Device#20/Function#1, bit 5 corresponds to Device#20/Function#2, bit 6 corresponds to Device#21/Function#0.</p> <p>When set, all PCI configuration accesses from Intel QuickPath Interconnect targeting the corresponding device's configuration space inside IOH are master aborted. When clear, configuration accesses targeting the device's configuration space are allowed.</p> <p>This bit has no effect on smbus and JTAG initiated accesses to corresponding device's config space.</p> <p>If software hides function#0 in device20, it needs to hide all functions within that device to comply with PCI rules.</p> <p>This bit has no impact on memory transactions targeting the device (e.g. memory transactions targeting the MBAR/ABAR region of IOAPIC)</p> |

21.6.5.12 IOHBUSNO: IOH Internal Bus Number

| Register: IOHBUSNO Device: 20 Function: 0 Offset: 10Ah | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:9 | RV | 0h | RSVD |
| 8 | RW | 0h | <p>Valid</p> <p>1: This IOH claims PCI config access to its internal devices (device/function) defined in Table 21-1 with the Bus number defined in bits[7:0] of this register only.</p> <p>0: This IOH claims PCI config access to its internal devices (device/function) defined in Table 21-1 with ANY Bus number, regardless of bits[7:0] of this register.</p> |



| Register: IOHBUSNO Device: 20 Function: 0 Offset: 10Ah | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RW | 00h | Internal bus number of IOH Is used to compare against the bus no in the Intel QuickPath Interconnect config tx and decide if the access is to the IOH internal devices or it goes out to a bus hierarchy below the IOH's internal bus. This register is programmed once at boot time and does not change after that. |

21.6.5.13 LIO.BASE: Local I/O Base Register

Provides the I/O range consumed by the hierarchy below an Intel QuickPath Interconnect port.

| Register: LIO.BASE Device: 20 Function: 0 Offset: 108h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:4 | RW | 0h | Local I/O Base Address Corresponds to A[15:12] of the I/O addresses of the local hierarchy below Intel® QuickPath Interconnect port. An inbound I/O address that satisfies 'local I/O base[7:4] <= A[15:12] <= local I/O limit[7:4]' is treated as a local peer-to-peer transaction that does not cross an Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in I/O cycle decoding. Setting LIO.BASE greater than LIO.LIMIT disables local IO peer-to-peer. This register is programmed once at boot time and does not change after that. |
| 3:0 | RV | 0h | Reserved |

21.6.5.14 LIO.LIMIT: Local I/O Limit Register

| Register: LIO.LIMIT Device: 20 Function: 0 Offset: 109h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:4 | RW | 0h | Local I/O Limit Address Corresponds to A[15:12] of the I/O addresses of the local hierarchy below an Intel QuickPath Interconnect port. An inbound I/O address that satisfies 'local I/O base[7:4] <= A[15:12] <= local I/O limit[7:4]' is treated as a local peer-to-peer transaction that does not cross an Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the I/O cycle decoding. Setting LIO.BASE greater than LIO.LIMIT disables local IO peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 3:0 | RV | 0h | Reserved |



21.6.5.15 LMMIOL.BASE: Local MMIOL Base

| Register: LMMIOL.BASE Device: 20 Function: 0 Offset: 10Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:8 | RW | 0h | Local MMIOL Base Address Corresponds to A[31:24] of MMIOL base address. An inbound memory address that satisfies 'local MMIOL base[15:8] <= A[31:24] <= local MMIOL limit[15:8]' is treated as a local peer-to-peer transaction that do not cross an Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting LMMIOL.BASE greater than LMMIOL.LIMIT disables local MMIOL peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 7:0 | RV | 0h | Reserved |

21.6.5.16 LMMIOL.LIMIT: Local MMIOL Limit

| Register: LMMIOL.LIMIT Device: 20 Function: 0 Offset: 10Eh | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:8 | RW | 0h | Local MMIOL Limit Address Corresponds to A[31:24] of MMIOL limit. An inbound memory address that satisfies 'local MMIOL base[15:8] <= A[31:24] <= local MMIOL limit[15:8]' is treated as a local peer-to-peer transaction that does not cross an Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting LMMIOL.BASE greater than LMMIOL.LIMIT disables local MMIOL peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 7:0 | RV | 0h | Reserved |

21.6.5.17 LMMIOH.BASE: Local MMIOH Base

| Register: LMMIOH.BASE Device: 20 Function: 0 Offset: 110h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:10 | RW | 0h | Local MMIOH Base Address Corresponds to A[31:26] of MMIOH base. An inbound memory address that satisfies 'local MMIOH base upper[31:0] local MMIOH base[15:10] <= A[63:26] <= local MMIOH limit upper[31:0] local MMIOH limit[15:10]' is treated as a local peer-to-peer transaction that does not cross an Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting LMMIOH.BASEU::LMMIOH.BASE greater than LMMIOH.LIMITU::LMMIOH.LIMIT disables local MMIOH peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |



| Register: LMMIOH.BASE Device: 20 Function: 0 Offset: 110h | | | |
|--|------|---------|-------------|
| Bit | Attr | Default | Description |
| 9:0 | RV | 0h | Reserved |

21.6.5.18 LMMIOH.LIMIT: Local MMIOH Limit

| Register: LMMIOH.LIMIT Device: 20 Function: 0 Offset: 112h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:10 | RW | 0h | Local MMIOH Limit Address Corresponds to A[31:26] of MMIOH limit. An inbound memory address that satisfies 'local MMIOH base upper[31:0] local MMIOH base[15:10] <= A[63:26] <= local MMIOH limit upper[31:0] local MMIOH limit[15:10]' is treated as local a peer-to-peer transactions that does not cross an Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting LMMIOH.BASEU LMMIOH.BASE greater than LMMIOH.LIMITU::LMMIOH.LIMIT disables local MMIOH peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 8:0 | RV | 0h | Reserved |

21.6.5.19 LMMIOH.BASEU: Local MMIOH Base Upper

| Register: LMMIOH.BASEU Device: 20 Function: 0 Offset: 114h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:19 | RO | 0h | Correspond to address A[63:51] of the local MMIOH range and is always 0. |
| 18:0 | RW | 0h | Local MMIOH Base Upper Address Corresponds to A[50:32] of MMIOH base. An inbound memory address that satisfies 'local MMIOH base upper[31:0]::local MMIOH base[15:10] <= A[63:26] <= local MMIOH limit upper[31:0]::local MMIOH limit[15:10]' is treated as a local peer-to-peer transaction that does not cross an Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting LMMIOH.BASEU::LMMIOH.BASE greater than LMMIOH.LIMITU::LMMIOH.LIMIT disables local MMIOH peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |



21.6.5.20 LMMIOH.LIMITU: Local MMIOH Limit Upper

| Register: LMMIOH.LIMITU Device: 20 Function: 0 Offset: 118h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:19 | RO | 0h | Correspond to address A[63:51] of the local MMIOH range and is always 0. |
| 18:0 | RW | 0h | Local MMIOH Limit Upper Address Corresponds to A[50:32] of MMIOH limit. An inbound memory address that satisfies 'local MMIOH base upper[31:0]::local MMIOH base[15:10] <= A[63:26] <= local MMIOH limit upper[31:0]::local MMIOH limit[15:10]' is treated as local a peer-to-peer transactions that does not cross an Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting LMMIOH.BASEU::LMMIOH.BASE greater than LMMIOH.LIMITU::LMMIOH.LIMIT disables local MMIOH peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |

21.6.5.21 LCFGBUS.BASE: Local Configuration Bus Number Base Register

| Register: LCFGBUS.BASE Device: 20 Function: 0 Offset: 11Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RW | 0h | Local Configuration Bus Number Base Corresponds to base bus number of bus number range allocated to the hierarchy below the Intel QuickPath Interconnect link. An inbound or outbound configuration tx falls within the local bus number range if 'Local Bus Number Base [7:0] <= Bus Number[7:0] <= Local Bus Number Limit [7:0]' and such transactions are treated as local peer-to-peer transactions that do not cross an Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the configuration cycle decoding. Setting LCFGBUS.BASE greater than LCFGBUS.LIMIT disables local peer-to-peer configuration cycles. This register is programmed once at boot time and does not change after that, including any quiesce flows. |



21.6.5.22 LCFGBUS.LIMIT: Local Configuration Bus Number Limit Register

| Register: LCFGBUS.LIMIT Device: 20 Function: 0 Offset: 11Dh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RW | 0h | Local Configuration Bus Number Limit Corresponds to Limit bus number of bus number range allocated to the hierarchy below the Intel QuickPath Interconnect link. An inbound configuration falls within the local bus number range if 'Local Bus Number Base [7:0] <= Bus Number[7:0] <= Local Bus Number Limit [7:0]' and such transactions are treated as local peer-to-peer transactions that do not cross an Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the configuration cycle decoding. Setting LCFGBUS.BASE greater than LCFGBUS.LIMIT disables local peer-to-peer configuration cycles. This register is programmed once at boot time and does not change after that, including any quiesce flows. This register is programmed once at boot time and does not change after that, including any quiesce flows. |

21.6.5.23 GIO.BASE: Global I/O Base Register

| Register: GIO.BASE Device: 20 Function: 0 Offset: 120h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:4 | RW | 0h | Global I/O Base Address Corresponds to A[15:12] of the I/O addresses of the entire I/O region. An inbound or outbound I/O address that satisfies 'global I/O base[7:4] <= A[15:12] <= global I/O limit[7:4]' but is outside of the local I/O address range is treated as remote peer I/O over Intel QuickPath Interconnect. Refer to Chapter 7 for more details of how this register is used in the I/O cycle decoding. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 3:0 | RV | 0h | Reserved |

21.6.5.24 GIO.LIMIT: Global I/O Limit Register

| Register: GIO.LIMIT Device: 20 Function: 0 Offset: 121h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:4 | RW | Fh | Global I/O Limit Address Corresponds to A[15:12] of the I/O addresses of the entire I/O region. An inbound or outbound I/O address that satisfies 'global I/O base[7:4] <= A[15:12] <= global I/O limit[7:4]' but is outside of the local I/O address range is treated as remote peer I/O over Intel QuickPath Interconnect. Refer to Chapter 7 for more details of how this register is used in the I/O cycle decoding. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 3:0 | RV | 0h | Reserved |



21.6.5.25 GMMIOL.BASE: Global MMIOL Base

| Register: GMMIOL.BASE Device: 20 Function: 0 Offset: 124h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:8 | RW | 0h | Global MMIOL Base Address Corresponds to A[31:24] of global MMIOL base. An inbound or outbound memory address that satisfies 'global MMIOL base[15:8] <= A[31:24] <= global MMIOL limit[15:8]' but is outside of the local MMIOL range is treated as a remote peer memory transaction over Intel QuickPath Interconnect. Refer to Chapter 7 for more details of how this register is used in MMIO decoding. Setting GMMIOL.BASE greater than GMMIOL.LIMIT disables global MMIOL peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 7:0 | RV | 0h | Reserved |

21.6.5.26 GMMIOL.LIMIT: Global MMIOL Limit

| Register: GMMIOL.LIMIT Device: 20 Function: 0 Offset: 126h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:8 | RW | 0h | Global MMIOL Limit Address Corresponds to A[31:24] of global MMIOL limit. An inbound or outbound memory address that satisfies 'global MMIOL base[15:8] <= A[31:24] <= global MMIOL limit[15:8]' but is outside of the local MMIOL range is treated as a remote peer-to-peer transaction over Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting GMMIOL.BASE greater than GMMIOL.LIMIT disables global MMIOL peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 7:0 | RV | 0h | Reserved |



21.6.5.27 GMMIOH.BASE: Global MMIOH Base

| Register: GMMIOH.BASE Device: 20 Function: 0 Offset: 128h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:10 | RW | 0h | Global MMIOH Base Address Corresponds to A[31:26] of global MMIOH base. An inbound or outbound memory address that satisfies 'global MMIOH base upper[31:0]::global MMIOH base[15:10] <= A[63:26] <= global MMIOH limit upper[31:0]::global MMIOH limit[15:10]' but is outside of the local MMIOH range is treated as a remote peer-to-peer transaction over Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting GMMIOH.BASEU::GMMIOH.BASE greater than GMMIOH.LIMITU::GMMIOH.LIMIT disables global MMIOH peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 9:0 | RV | 0h | Reserved |

21.6.5.28 GMMIOH.LIMIT: Global MMIOH Limit

| Register: GMMIOH.LIMIT Device: 20 Function: 0 Offset: 12Ah | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:10 | RW | 0h | Global MMIOH Limit Address Corresponds to A[31:26] of global MMIOH limit. An inbound or outbound memory address that satisfies 'global MMIOH base upper[31:0]::global MMIOH base[15:10] <= A[63:26] <= global MMIOH limit upper[31:0]::global MMIOH limit[15:10]' but is outside of the local MMIOH range is treated as a remote peer-to-peer transaction over Intel [®] QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting GMMIOH.BASEU::GMMIOH.BASE greater than GMMIOH.LIMITU::GMMIOH.LIMIT disables global MMIOH peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 9:0 | RV | 0h | Reserved |

21.6.5.29 GMMIOH.BASEU: Global MMIOH Base Upper

| Register: GMMIOH.BASEU Device: 20 Function: 0 Offset: 12Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:19 | RO | 0h | Correspond to address A[63:51] of the global MMIOH range and is always 0. |



| Register: GMMIOH.BASEU Device: 20 Function: 0 Offset: 12Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 18:0 | RW | 0h | Global MMIOH Base Upper Address Corresponds to A[50:32] of global MMIOH base. An inbound or outbound memory address that satisfies 'global MMIOH base upper[31:0]::global MMIOH base[15:10] <= A[63:26] <= global MMIOH limit upper[31:0]::global MMIOH limit[15:10]' but is outside of the local MMIOH range is treated as a remote peer-to-peer transaction over Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting GMMIOH.BASEU::GMMIOH.BASE greater than GMMIOH.LIMITU::GMMIOH.LIMIT disables global MMIOH peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |

21.6.5.30 GMMIOH.LIMITU: Global MMIOH Limit Upper

| Register: GMMIOH.LIMITU Device: 20 Function: 0 Offset: 130h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:19 | RO | 0h | Correspond to address A[63:51] of the global MMIOH range and is always 0. |
| 18:0 | RW | 0h | Global MMIOH Limit Upper Address Corresponds to A[51:32] of global MMIOH limit. An inbound or outbound memory address that satisfies 'global MMIOH base upper[31:0]::global MMIOH base[15:10] <= A[63:26] <= global MMIOH limit upper[31:0]::global MMIOH limit[15:10]' but is outside of the local MMIOH range is treated as a remote peer-to-peer transaction over Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the MMIO decoding. Setting GMMIOH.BASEU::GMMIOH.BASE greater than GMMIOH.LIMITU::GMMIOH.LIMIT disables global MMIOH peer-to-peer. This register is programmed once at boot time and does not change after that, including any quiesce flows. |

21.6.5.31 GCFGBUS.BASE: Global Configuration Bus Number Base Register

| Register: GCFGBUS.BASE Device: 20 Function: 0 Offset: 134h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RW | 0h | Global Configuration Bus Number Base Corresponds to base bus number of bus number range that spans all IOHs in a partition. An inbound or outbound configuration tx that satisfies 'Global Bus Number Base [7:0] <= Bus Number[7:0] <= Global Bus Number Limit [7:0]' but is outside of the local bus number range is treated as a remote peer-to-peer transaction over Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the configuration cycle decoding. |



21.6.5.32 GCFGBUS.LIMIT: Global Configuration Bus Number Limit Register

| Register: GCFGBUS.LIMIT Device: 20 Function: 0 Offset: 135h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RW | FFh | Global Configuration Bus Number Limit Corresponds to limit bus number of bus number range allocated across all IOHs in the partition. An inbound or outbound configuration that satisfies 'Global Bus Number Base [7:0] <= Bus Number[7:0] <= Global Bus Number Limit [7:0]' but is outside of the low bus number range is treated as a remote peer-to-peer transaction over Intel QuickPath Interconnect link. Refer to Chapter 7 for more details of how this register is used in the configuration cycle decoding. This register is programmed once at boot time and does not change after that, including any quiesce flows. |

21.6.5.33 VTBAR: Base Address Register for Intel VT-d Chipset Registers

| Register: VTBAR Device: 20 Function: 0 Offset: 180h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:13 | RWL | 0 | Intel VT-d Chipset Base Address: Provides an aligned 8K base address for IOH registers relating to Intel VT-d. All inbound accesses to this region are completer aborted by the IOH. This register is programmed once at boot time and does not change after that, including any quiesce flows. |
| 12:1 | RV | 0 | Reserved |
| 0 | RWL | 0 | Intel VT-d Chipset Base Address Enable: Enables the VTBAR register. This bit is RO if "Intel VT-d enable fuse" is OFF |

21.6.5.34 VTGENCTRL: Intel VT-d General Control Register

| Register: VTGENCTRL Device: 20 Function: 0 Offset: 184h | | | |
|--|------|---------|-------------|
| Bit | Attr | Default | Description |
| 15:11 | RV | 0 | Reserved |
| 10:8 | RV | 111 | Reserved |



| Register: VTGENCTRL Device: 20 Function: 0 Offset: 184h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:4 | RWL | 0011 | <p>Non-Isoch HPA_LIMIT: Represents the host processor addressing limit</p> <p>0000: 2³⁶ (that is, bits 35:0) 0001: 2³⁷ (that is, bits 36:0) 0010: 2³⁸ 0011: 2³⁹ 0100: 2⁴⁰ ... 1111: 2⁵¹ (that is, bits 50:0)</p> <p>When Intel VT-d translation is enabled on an Intel VT-d engine (non-isoch), all host addresses (during page walks) that go beyond the limit specified in this register will be aborted by IOH. Note that pass-through and 'translated' ATS accesses carry the host-address directly in the access and are subject to this check as well.</p> <p>When Intel VT-d translation is enabled or disabled on a Intel VT-d engine (isoch or non-isoch), all host addresses (during page walks) that go beyond the limit specified in this register will be aborted by the IOH. Note that pass-through and 'translated' ATS accesses carry the host-address directly in the access and are subject to this check as well.</p> <p>Note for Error logging due to HPA limits check violations: When Intel VT-d translation is enabled, HPA limit check violations from the following requests will not be logged in the error register.</p> <ol style="list-style-type: none"> Translated request(AT=10) Pass-through untranslated request. <p>When Intel VT-d translation is disabled, HPA limit violations from untranslated request will be logged in the IOHERRST register when the HPA limit is set to 2³⁶. HPA limit check violation with other HPA limit settings will not be logged.</p> |
| 3:0 | RWL | 8h | <p>Non-Isoch GPA_LIMIT: Represents the guest virtual addressing limit for the non-Isoch Intel VT-d engine.</p> <p>0000: 2⁴⁰ (that is, bits 39:0) 0001: 2⁴¹ (that is, bits 40:0) .. 0111: 2⁴⁷ 1000: 2⁴⁸ 1001-1111: Reserved</p> <p>When Intel VT-d translation is enabled, all incoming guest addresses from PCI Express, associated with the non-isoch Intel VT-d engine, that go beyond the limit specified in this register will be aborted by IOH and a UR response returned. This register is not used when translation is not enabled. Note that 'translated' and 'pass-through' addresses are in the 'host-addressing' domain and NOT 'guest-addressing' domain and hence GPA_LIMIT checking on those accesses are bypassed and instead HPA_LIMIT checking applies.</p> |



21.6.5.35 VTGENCTRL2: Intel VT-d General Control 2 Register

| Register: VTGENCTRL2 Device: 20 Function: 0 Offset: 18Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:11 | RV | 0 | Reserved |
| 10:7 | RWL | Fh | LRU Timer |
| 6:5 | RWL | 0 | Prefetch Control: Queued invalidation, interrupt table read, context table reads and root table reads NEVER have prefetch/snarf/reuse capability. This is a general rule. Beyond that the Prefetch Control bits control additional behavior as shown below. Prefetch Control: This field controls which VT-d reads are to be considered for prefetch/snarf/reuse in the CSI buffers. 00 -> Prefetch/snarf/reuse is turned off. i.e. CSI cluster never reuses the VT-d read data. 01 -> Prefetch/snarf/reuse is enabled for all leaf/non-leaf VT-d page walk reads. 10 -> Reserved. 11 -> Prefetch/snarf/resue is enabled on all leaf (not non-leaf) VT-d page walk reads regardless of the setting of the CC.ALH bit. |
| 4:0 | RV | 0 | Reserved |

21.6.5.36 VTSTS: Intel VT-d Status Register

| Register: VTSTS Device: 20 Function: 0 Offset: 190h | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:2 | RV | 0 | Reserved |
| 1 | RW1CS | 0 | Interrupt transaction seen on VC1/VCp |
| 0 | RW1CS | 0 | ATS command detected toward ESI port |

21.6.5.37 VTUNCERRSTS - VT Uncorrectable Error Status Register

| Register: VTUNCERRSTS Device: 20 Function: 0 Offset: 1A8h | | | |
|--|--------|---------|--|
| Bit | Attr | Default | Description |
| 31 | RW1CST | 0 | Intel VT-d spec defined errors: When set, this bit is set when a Intel VT-d spec defined error has been detected (and logged in the Intel VT-d fault registers) |
| 30:9 | RV | 0 | Reserved |
| 8 | RW1CST | 0 | Protected memory region space violated status |
| 7 | RV | 0 | Reserved |
| 6 | RW1CST | 0 | Unsuccessful status received in Intel QPI read completion status |
| 5 | RW1CST | 0 | TLB1 parity error status |



| Register: VTUNCERRSTS Device: 20 Function: 0 Offset: 1A8h | | | |
|--|--------|---------|---|
| Bit | Attr | Default | Description |
| 4 | RW1CST | 0 | TLB0 parity error status |
| 3 | RW1CST | 0 | Data parity error while doing a L3 lookup status |
| 2 | RW1CST | 0 | Data parity error while doing a L2 lookup status |
| 1 | RW1CST | 0 | Data parity error while doing a L1 lookup status |
| 0 | RW1CST | 0 | Intel VT-Data parity error while doing a context cache look up status |

21.6.5.38 VTUNCERRMSK - VT Uncorrectable Error Mask Register

| Register: VTUNCERRMSK Device: 20 Function: 0 Offset: 1ACh | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31 | RWS | 0 | Mask reporting VT-d defined errors to IOH core logic |
| 30:9 | RV | 0 | Reserved |
| 8 | RWS | 0 | Protected memory region space violated mask |
| 7 | RV | 0 | Reserved |
| 6 | RWS | 0 | Unsuccessful status received in Intel QPI read completion mask |
| 5 | RWS | 0 | TLB1 parity error mask |
| 4 | RWS | 0 | TLB0 parity error mask |
| 3 | RWS | 0 | Data parity error while doing a L3 lookup mask |
| 2 | RWS | 0 | Data parity error while doing a L2 lookup mask |
| 1 | RWS | 0 | Data parity error while doing a L1 lookup mask |
| 0 | RWS | 0 | Data parity error while doing a context cache look up mask |

21.6.5.39 VTUNCERRSEV - VT Uncorrectable Error Severity Register

| Register: VTUNCERRSEV Device: 20 Function: 0 Offset: 1B0h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31 | RWS | 0 | Intel VT-d spec defined error severity: When set, this bit escalates reporting of VT-d spec defined errors, as FATAL errors. When clear, those errors are escalated as Nonfatal errors. |
| 30:9 | RV | 0 | Reserved |
| 8 | RWS | 1 | Protected memory region space violated severity |
| 7 | RV | 0 | Reserved |
| 6 | RWS | 0 | Unsuccessful status received in Intel QuickPath Interconnect read completion severity |
| 5 | RWS | 1 | TLB1 parity error severity |



| Register: VTUNCERSEV Device: 20 Function: 0 Offset: 1B0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 4 | RWS | 1 | TLB0 parity error severity |
| 3 | RWS | 1 | Data parity error while doing a L3 lookup severity |
| 2 | RWS | 1 | Data parity error while doing a L2 lookup severity |
| 1 | RWS | 1 | Data parity error while doing a L1 lookup severity |
| 0 | RWS | 1 | Data parity error while doing a context cache look up severity |

21.6.5.40 VTUNCERRPTR - VT Uncorrectable Error Pointer Register

| Register: VTUNCERRPTR Device: 20 Function: 0 Offset: 1B4h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:5 | RV | 0 | Reserved |
| 4:0 | ROS | 0 | Intel VT Uncorrectable First Error Pointer This field points to which of the unmasked uncorrectable errors happened first. This field is only valid when the corresponding error is unmasked and the status bit is set and this field is rearmed to load again when the status bit indicated to by this pointer is cleared by software from 1 to 0. Value of 0x0 corresponds to bit 0 in VTUNCERRSTS register, value of 0x1 corresponds to bit 1 etc. |



21.6.6 Semaphore and Scratch Pad Registers (Dev20, Function 1)

Table 21-9. Semaphore and Scratch pad Registers (Dev 20, Function 1)

| | | | | | | |
|----------------------------|--------|-------|--------|---------------------|--------|--------|
| DID | VID | 000h | SR[1] | 080h | | |
| PCISTS | PCICMD | 004h | SR[2] | 084h | | |
| CCR | RID | 008h | SR[3] | 088h | | |
| HDR | CLS | 00Ch | SR[4] | 08Ch | | |
| | | 010h | SR[5] | 090h | | |
| | | 014h | SR[6] | 094h | | |
| | | 018h | SR[7] | 098h | | |
| | | 01Ch | SR[8] | 09Ch | | |
| | | 020h | SR[9] | 0A0h | | |
| | | 024h | SR[10] | 0A4h | | |
| | | 028h | SR[11] | 0A8h | | |
| | | SID | SVID | 02Ch | SR[12] | 0ACh |
| | | | | 030h | SR[13] | 0B0h |
| | | | | CAPPTR ¹ | 034h | SR[14] |
| 038h | SR[15] | | | 0B8h | | |
| INTP | INTL | | | 03Ch | SR[16] | 0BCh |
| EXPCAP | NXTPTR | CAPID | 040h | SR[17] | 0C0h | |
| DEVCAP | | 044h | | | | |
| DEVSTS | DEVCON | 048h | | | 0C4h | |
| RESERVED PCIe Header space | | 04Ch | | | 0C8h | |
| | | 050h | | | 0CCh | |
| | | 054h | | | 0D0h | |
| | | 058h | | | 0D4h | |
| | | 05Ch | | | 0D8h | |
| | | 060h | | | 0DCh | |
| | | 064h | | | 0E0h | |
| | | 068h | | | 0E4h | |
| 06Ch | 0E8h | | | | | |
| 070h | 0ECh | | | | | |
| 074h | 0F0h | | | | | |
| 078h | 0F4h | | | | | |
| 07Ch | 0F8h | | | | | |
| SR[0] | | 07Ch | CWR[8] | 0FCh | | |

Notes:

1. CAPPTR points to the first capability block



| | | | | |
|----------------------------|-------|--------|------|------|
| RESERVED PCIe Header space | 100h | IR[16] | 180h | |
| CWR[9] | 104h | IR[17] | 184h | |
| CWR[10] | 108h | | 188h | |
| CWR[11] | 10Ch | | 18Ch | |
| CWR[12] | 110h | | 190h | |
| CWR[13] | 114h | | 194h | |
| CWR[14] | 118h | | 198h | |
| CWR[15] | 11Ch | | 19Ch | |
| CWR[16] | 120h | | 1A0h | |
| CWR[17] | 124h | | 1A4h | |
| | 128h | | 1A8h | |
| | 12Ch | | 1ACh | |
| | 130h | | 1B0h | |
| | 134h | | 1B4h | |
| | 138h | | 1B8h | |
| | 13Ch | | 1BCh | |
| | IR[0] | | 140h | 1C0h |
| | IR[1] | | 144h | 1C4h |
| | IR[2] | | 148h | 1C8h |
| | IR[3] | | 14Ch | 1CCh |
| IR[4] | 150h | | 1D0h | |
| IR[5] | 154h | | 1D4h | |
| IR[6] | 158h | 1D8h | | |
| IR[7] | 15Ch | 1DCh | | |
| IR[8] | 160h | 1E0h | | |
| IR[9] | 164h | 1E4h | | |
| IR[10] | 168h | 1E8h | | |
| IR[11] | 16Ch | 1ECh | | |
| IR[12] | 170h | 1F0h | | |
| IR[13] | 174h | 1F4h | | |
| IR[14] | 178h | 1F8h | | |
| IR[15] | 17Ch | 1FCh | | |

21.6.6.1 SR[0:3]: Scratch Pad Register 0-3 (Sticky)

| Register: SR[4:7] Device: 20 Function: 1 Offset: 07Ch-088h | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLBS | 0h | Scratch Pad – Sticky Sticky scratch pad registers for firmware utilization |



21.6.6.2 SR[4:7]: Scratch Pad Register 4-7 (Sticky)

| Register: SR[4:7] Device: 20 Function: 1 Offset: 08Ch-098h | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWSLB | 0h | Scratch Pad – Sticky Sticky scratch pad registers for firmware utilization |

21.6.6.3 SR[8:11]: Scratch Pad Register 8-11 (Non-Sticky)

| Register: SR[8:11] Device: 20 Function: 1 Offset: 09Ch-0A8h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLB | 0h | Scratch Pad – Non-Sticky Non-sticky scratch pad registers for firmware utilization |

21.6.6.4 SR[12:15]: Scratch Pad Register 12-15 (Non-Sticky)

| Register: SR[12:15] Device: 20 Function: 1 Offset: 0ACh-0B8h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLB | 0h | Scratch Pad – Non-Sticky Non-sticky scratch pad registers for firmware utilization |

21.6.6.5 SR[16:17]: Scratch Pad Register 16-17 (Non-Sticky)

| Register: SR[16:17] Device: 20 Function: 1 Offset: 0BCh-0C0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLB | 0h | Scratch Pad – Non-Sticky Non-sticky scratch pad registers for firmware utilization |



21.6.6.6 CWR[0:3]: Conditional Write Registers 0-3

| Register: CWR[0:3] Device: 20 Function: 1 Offset: 0DCh-0E8h | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLBS | 0 | Conditional Write These registers are physically mapped to scratch pad registers. A read from CWR[n] reads SR[n]. A write to CWR[n] writes SR[n] if SR[n][0] = 0 before the write, and has no effect otherwise. The registers provide firmware with synchronization variables (semaphores) that are overloaded onto the same physical registers as SR. |

21.6.6.7 CWR[4:7]: Conditional Write Registers 4-7

| Register: CWR[4:7] Device: 20 Function: 1 Offset: 0ECh-0F8h | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLBS | 0 | Conditional Write These registers are physically mapped to scratch pad registers. A read from CWR[n] reads SR[n]. A write to CWR[n] writes SR[n] if SR[n][0] = 0 before the write, and has no effect otherwise. The registers provide firmware with synchronization variables (semaphores) that are overloaded onto the same physical registers as SR. |

21.6.6.8 CWR[8:11]: Conditional Write Registers 8-11

| Register: CWR[8:11] Device: 20 Function: 1 Offset: 0FCh-10Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLB | 0 | Conditional Write These registers are physically mapped to scratch pad registers. A read from CWR[n] reads SR[n]. A write to CWR[n] writes SR[n] if SR[n][0] = 0 before the write, and has no effect otherwise. The registers provide firmware with synchronization variables (semaphores) that are overloaded onto the same physical registers as SR. |



21.6.6.9 CWR[12:15]: Conditional Write Registers 12-15

| Register: CWR[12:15] Device: 20 Function: 1 Offset: 110h-11Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLB | 0 | Conditional Write These registers are physically mapped to scratch pad registers. A read from CWR[n] reads SR[n]. A write to CWR[n] writes SR[n] if SR[n][0] = 0 before the write, and has no effect otherwise. The registers provide firmware with synchronization variables (semaphores) that are overloaded onto the same physical registers as SR. |

21.6.6.10 CWR[16:17]: Conditional Write Registers 16-17

| Register: CWR[16:17] Device: 20 Function: 1 Offset: 120h-124h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLB | 0h | Conditional Write These registers are physically mapped to scratch pad registers. A read from CWR[n] reads SR[n]. A write to CWR[n] writes SR[n] if SR[n][0] = 0 before the write, and has no effect otherwise. The registers provide firmware with synchronization variables (semaphores) that are overloaded onto the same physical registers as SR. |

21.6.6.11 IR[0:3]: Increment Registers 0-3

| Register: IR[0:3] Device: 20 Function: 1 Offset: 140h-14Ch | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLBS | 0 | Increment These registers are physically mapped to scratch pad registers. A read from IR[n] reads SR[n] and then increments SR[n]. A write to IR[n] increments SR[n] while the write data is unused. Increments within SR[n] for reads and writes roll over to zero. The read or write and the increment side effect are atomic with respect to other accesses. The registers provide firmware with synchronization variables (semaphores) that are overloaded onto the same physical registers as SR. |



21.6.6.12 IR[4:7]: Increment Registers 4-7

| Register: IR[4:7] Device: 20 Function: 1 Offset: 150h-15Ch | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLBS | 0 | Increment These registers are physically mapped to scratch pad registers. A read from IR[n] reads SR[n] and then increments SR[n]. A write to IR[n] increments SR[n] while the write data is unused. Increments within SR[n] for reads and writes roll over to zero. The read or write and the increment side effect are atomic with respect to other accesses. The registers provide firmware with synchronization variables (semaphores) that are overloaded onto the same physical registers as SR. |

21.6.6.13 IR[8:11]: Increment Registers 8-11

| Register: IR[8:11] Device: 20 Function: 1 Offset: 160h-16Ch by 4 | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLB | 0 | Increment These registers are physically mapped to scratch pad registers. A read from IR[n] reads SR[n] and then increments SR[n]. A write to IR[n] increments SR[n] while the write data is unused. Increments within SR[n] for reads and writes roll over to zero. The read or write and the increment side effect are atomic with respect to other accesses. The registers provide firmware with synchronization variables (semaphores) that are overloaded onto the same physical registers as SR. |

21.6.6.14 IR[12:15]: Increment Registers 12-15

| Register: IR[12:15] Device: 20 Function: 1 Offset: 170h-17Ch by 4 | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLB | 0 | Increment These registers are physically mapped to scratch pad registers. A read from IR[n] reads SR[n] and then increments SR[n]. A write to IR[n] increments SR[n] while the write data is unused. Increments within SR[n] for reads and writes roll over to zero. The read or write and the increment side effect are atomic with respect to other accesses. The registers provide firmware with synchronization variables (semaphores) that are overloaded onto the same physical registers as SR. |



21.6.6.15 IR[16:17]: Increment Registers 16-17

| Register: IR[16:17] Device: 20 Function: 1 Offset: 180h-184h by 4 | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RWLB | 0h | Increment These registers are physically mapped to scratch pad registers. A read from IR[n] reads SR[n] and then increments SR[n]. A write to IR[n] increments SR[n] while the write data is unused. Increments within SR[n] for reads and writes roll over to zero. The read or write and the increment side effect are atomic with respect to other accesses. The registers provide firmware with synchronization variables (semaphores) that are overloaded onto the same physical registers as SR. |



21.6.7 IOH System/Control Status Registers

Table 21-10. IOH Control/Status & Global Error Register Map (Dev 20, Function 2, Page 1 of 4)

| | | | | |
|----------------------------|--------|------|-----------|------|
| DID | VID | 000h | QPIERRSV | 080h |
| PCISTS | PCICMD | 004h | | 084h |
| CCR | RID | 008h | QPIPERRSV | 088h |
| HDR | CLS | 00Ch | IOHERRSV | 08Ch |
| | | 010h | | 090h |
| | | 014h | PCIERRSV | 094h |
| | | 018h | THRERRSV | 098h |
| | | 01Ch | SYSMAP | 09Ch |
| | | 020h | VIRAL | 0A0h |
| | | 024h | ERRPINCTL | 0A4h |
| SID | SVID | 028h | ERRPINST | 0A8h |
| | | 02Ch | ERRPINDAT | 0ACh |
| | | 030h | | 0B0h |
| | | 034h | VPPCTL | 0B4h |
| | | 038h | VPPSTS | 0B8h |
| | INTP | 03Ch | | 0BCh |
| EXPCAP | NXTPTR | 040h | PRSTRDY | 0C0h |
| | CAPID | 044h | GENMCA | 0C4h |
| DEVCAP | | 048h | GENVIRAL | 0C8h |
| DEVSTS | DEVCON | 04Ch | SYRE | 0CCh |
| RESERVED PCIe Header space | | 050h | FREQ | 0D0h |
| | | 054h | | 0D4h |
| | | 058h | | 0D8h |
| | | 05Ch | | 0DCh |
| | | 060h | | 0E0h |
| | | 064h | | 0E4h |
| | | 068h | CAPTIM | 0E8h |
| | | 06Ch | | 0ECh |
| | | 070h | | 0F0h |
| | | 074h | | 0F4h |
| 078h | | 0F8h | | |
| | | 07Ch | EOI_CTRL | 0FCh |

Notes:

1. CAPPTR points to the first capability block



Table 21-11. IOH Control/Status & Global Error Register Map (Dev 20, Function 2, Page 2 of 4)

| | | | |
|----------------------------|------|------------|------|
| RESERVED PCIe Header space | 100h | | 180h |
| | 104h | | 184h |
| | 108h | | 188h |
| | 10Ch | | 18Ch |
| | 110h | | 190h |
| | 114h | | 194h |
| | 118h | | 198h |
| | 11Ch | | 19Ch |
| | 120h | | 1A0h |
| | 124h | | 1A4h |
| | 128h | | 1A8h |
| | 12Ch | | 1ACh |
| | 130h | | 1B0h |
| | 134h | | 1B4h |
| | 138h | | 1B8h |
| | 13Ch | | 1BCh |
| | 140h | GNERRST | 1C0h |
| | 144h | GFERRST | 1C4h |
| | 148h | GERRCTL | 1C8h |
| | 14Ch | GSYSST | 1CCh |
| | 150h | GSYSCTL | 1D0h |
| | 154h | GTIME | 1D4h |
| | 158h | | 1D8h |
| | 15Ch | GFFERRST | 1DCh |
| | 160h | | 1E0h |
| | 164h | GFFERRTIME | 1E4h |
| | 168h | GFNERRST | 1E8h |
| | 16Ch | GNFERRST | 1ECh |
| | 170h | | 1F0h |
| | 174h | GNFERRTIME | 1F4h |
| | 178h | GNNERRST | 1F8h |
| | 17Ch | | 1FCh |



Table 21-12. IOH Local Error Map #1 (Dev 20, Function 2, Page 3 of 4)

| | | | |
|----------------|------|--|------|
| QPIOERRST | 200h | | 280h |
| QPIOERRCTL | 204h | | 284h |
| QPIOFFERRST | 208h | | 288h |
| QPIOFNERRST | 20Ch | | 28Ch |
| QPIONFERRST | 210h | | 290h |
| QPIONNERRST | 214h | | 294h |
| QPIOERRCNTSEL | 218h | | 298h |
| QPIOERRCNT | 21Ch | | 29Ch |
| | 220h | | 2A0h |
| | 224h | | 2A4h |
| | 228h | | 2A8h |
| | 22Ch | | 2ACh |
| QPIPOERRST | 230h | | 2B0h |
| QPIPOERRCTL | 234h | | 2B4h |
| QPIPOFFERRST | 238h | | 2B8h |
| QPIPOFNERRST | 23Ch | | 2BCh |
| QPIPOFFERRHD | 240h | | 2C0h |
| | 244h | | 2C4h |
| | 248h | | 2C8h |
| | 24Ch | | 2CCh |
| QPIPONFERRST | 250h | | 2D0h |
| QPIPONNERRST | 254h | | 2D4h |
| QPIPONFERRHD | 258h | | 2D8h |
| | 25Ch | | 2DCh |
| | 260h | | 2E0h |
| | 264h | | 2E4h |
| QPIPOERRCNTSEL | 268h | | 2E8h |
| QPIPOERRCNT | 26Ch | | 2ECh |
| | 270h | | 2F0h |
| | 274h | | 2F4h |
| | 278h | | 2F8h |
| | 27Ch | | 2FCh |



Table 21-13. IOH Local Error Map #2 (Dev 20, Function 2, Page 4 of 4)

| | | | |
|--------------|------|--|------|
| IOHERRST | 300h | | 380h |
| IOHERRCTL | 304h | | 384h |
| IOHFFERRST | 308h | | 388h |
| IOHFFERRHD | 30Ch | | 38Ch |
| | 310h | | 390h |
| | 314h | | 394h |
| | 318h | | 398h |
| IOHFNERRST | 31Ch | | 39Ch |
| IOHNFERRST | 320h | | 3A0h |
| IOHNFERRHD | 324h | | 3A4h |
| | 328h | | 3A8h |
| | 32Ch | | 3ACh |
| | 330h | | 3B0h |
| IOHNNERRST | 334h | | 3B4h |
| | 338h | | 3B8h |
| IOHERRCNTSEL | 33Ch | | 3BCh |
| IOHERRCNT | 340h | | 3C0h |
| | 344h | | 3C4h |
| | 348h | | 3C8h |
| | 34Ch | | 3CCh |
| | 350h | | 3D0h |
| | 354h | | 3D4h |
| | 358h | | 3D8h |
| | 35Ch | | 3DCh |
| THRERRST | 360h | | 3E0h |
| THRERRCTL | 364h | | 3E4h |
| THRFFERRST | 368h | | 3E8h |
| THRFNERRST | 36Ch | | 3ECh |
| THRNFERRST | 370h | | 3F0h |
| THRNNERRST | 374h | | 3F4h |
| THRERRCNTSEL | 378h | | 3F8h |
| THRERRCNT | 37Ch | | 3FCh |



21.6.7.1 QPIERRSV: Intel QuickPath Interconnect Link/Physical Error Severity Register

This register associates the detected Intel QPI Link and Physical Layer errors to an error severity level. An individual error is reported with the corresponding severity in this register. Software can program the error severity to one of the three severities supported by IOH. This register is sticky and can only be reset by PWRGOOD. The default error severity mapping is defined in [Table 16-2](#).

| Register: QPIERRSV Device: 20 Function: 2 Offset: 080h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:26 | RV | 0 | <i>Reserved</i> |
| 25:24 | RWS | 10 | D4 - Intel QuickPath Interconnect Link Internal Parity Error This field determines the error severity for the corresponding Severity error. Two-bit encoding as follows: 00: Error Severity Level 0 (Correctable) 01: Error Severity Level 1 (Recoverable) 10: Error Severity Level 2 (Fatal) 11: Reserved |
| 23:22 | RWS | 10 | D3 - Intel QPI Link Layer Control Error |
| 21:20 | RWS | 10 | C0 - Intel QPI Link Layer detected CRC error - unsuccessful link level retry - entered LLR abort state See QPI[1:0]FERRFLIT0 and QPI[1:0]FERRFLIT1 for flit info. |
| 19:18 | RWS | 00 | B1 - Intel QPI Link Layer detected CRC error - successful link level retry after PHY reinit |
| 17:16 | RWS | 00 | B0 - Intel QPI Link Layer CRC - successful link level retry |
| 15:14 | RWS | 10 | DG - Intel QPI Link Layer Detected unsupported/undefined msgclass/opcode/vn packet Error |
| 13:12 | RWS | 00 | B5 - Potential Spurious CRC error on L0s/L1 Exit |
| 11:10 | RWS | 00 | B6 - Intel QPI Link Layer CRC error |
| 9:8 | RWS | 10 | D2 - Intel QPI Physical Layer Initialization Failure |
| 7:6 | RWS | 10 | D1 - Intel QPI Physical Layer Detected Latency Buffer Rollover |
| 5:4 | RWS | 10 | D0 - Intel QuickPath Interconnect Physical Layer Detected Drift Buffer Alarm |
| 3:2 | RWS | 01 | C7 - Intel QPI Physical Layer Reset Successful with Reduced Width |
| 1:0 | RWS | 00 | B2 - Intel QPI Physical Layer Successful Reset at same Width |



21.6.7.2 QPIPERRSV: Intel QuickPath Interconnect Protocol Error Severity Register

This register associates the detected Intel QuickPath Interconnect Protocol and Routing layer errors to an error severity level. An individual error is reported with the corresponding severity in this register. Software can program the error severity to one of the three severities supported by IOH. This register is sticky and can only be reset by PWRGOOD. The default error severity mapping is defined in [Table 16-2](#).

| Register: QPIPERRSV Device: 20 Function: 2 Offset: 084h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 63:40 | RV | 0 | <i>Reserved</i> |
| 39:38 | RWS | 10 | DH - Intel QPI Protocol Layer Detected unsupported/undefined packet Error This field determines the error severity for the corresponding Severity error. Two-bit encoding as follows: 00: Error Severity Level 0 (Correctable) 01: Error Severity Level 1 (Recoverable) 10: Error Severity Level 2 (Fatal) 11: Reserved |
| 37:36 | RWS | 10 | DF - Illegal Inbound Request |
| 35:34 | RWS | 10 | DE - Routing Table Invalid |
| 33:32 | RV | 0 | <i>Reserved</i> |
| 31:30 | RWS | 10 | DC - Protocol SAD illegal or non-existent memory for outbound snoop |
| 29:28 | RWS | 10 | DB - Protocol Parity Error |
| 27:26 | RWS | 10 | DA - Protocol Queue/Table Overflow or Underflow |
| 25:24 | RWS | 10 | D9 - Protocol Layer Received Viral from Intel QPI |
| 23:22 | RWS | 10 | D8 - Protocol Layer Received Illegal packet field or Incorrect Target NodeID |
| 21:20 | RWS | 10 | D7 - Protocol Layer Received Unexpected Response/Completion |
| 19:18 | RWS | 10 | D6 - Protocol Layer Received Failed Response |
| 17:16 | RWS | 10 | D5 - Protocol Layer Detected Time-Out in ORB |
| 15:10 | RV | 0 | <i>Reserved</i> |
| 9:8 | RWS | 01 | C3 - CSR access crossing 32-bit boundary |
| 7:6 | RWS | 01 | C2 - Write Cache Un-correctable ECC |
| 5:4 | RWS | 01 | C1 - Protocol Layer Received Poisoned Packet. |
| 3:2 | RWS | 00 | B4 - Write Cache Correctable ECC |
| 1:0 | RWS | 00 | B3 - Intel QPI CPEI Error Status |



21.6.7.3 IOHERRSV: IOH Core Error Severity Register

This register associates the detected IOH internal core errors to an error severity level. An individual error is reported with the corresponding severity in this register. Software can program the error severity to one of the three severities supported by IOH. This register is sticky and can only be reset by PWRGOOD:

| Register: IOHERRSV Device: 20 Function: 2 Offset: 08Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:14 | RV | 0h | Reserved |
| 13:12 | RWS | 01 | C6 - FIFO Overflow/Underflow error This field determines the error severity for the corresponding Severity error. Two-bit encoding as follows: 00: Error Severity Level 0 (Correctable) 01: Error Severity Level 1 (Recoverable) 10: Error Severity Level 2 (Fatal) 11: Reserved |
| 11:10 | RWS | 01 | C5 -Completor abort address error This field determines the error severity for the corresponding Severity error. Two-bit encoding as follows: 00: Error Severity Level 0 (Correctable) 01: Error Severity Level 1 (Recoverable) 10: Error Severity Level 2 (Fatal) 11: Reserved |
| 9:8 | RWS | 01 | C4 -Master abort address error This field determines the error severity for the corresponding Severity error. Two-bit encoding as follows: 00: Error Severity Level 0 (Correctable) 01: Error Severity Level 1 (Recoverable) 10: Error Severity Level 2 (Fatal) 11: Reserved |
| 7:0 | RV | 0h | Reserved |

21.6.7.4 MIERRSV: Miscellaneous Error Severity Register

This register associates the detected IOH miscellaneous errors to an error severity level. An individual error is reported with the corresponding severity in this register. Software can program the error severity to one of the three severities supported by IOH. This register is sticky and can only be reset by PWRGOOD:

| Register: MIERRSV Device: 20 Function: 2 Offset: 090h | | | |
|--|------|---------|-------------|
| Bit | Attr | Default | Description |
| 31:6 | RV | 0h | Reserved |



| Register: MIERRSV Device: 20 Function: 2 Offset: 090h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:6 | RWS | 00 | 23 - VPP Error Severity Refer to bit [1:0] for description |
| 5:4 | RWS | 00 | 22 - Persistent JTAG Error Severity Refer to bit[1:0] for description. |
| 3:2 | RWS | 00 | 21 - Persistent SMBus Retry Status Severity Refer to bit[1:0] for description. |
| 1:0 | RWS | 00 | 20 - IOH Configuration Register Parity error Severity This field determines the error severity for the corresponding Severity error. Two-bit encoding as follows: 00: Error Severity Level 0 (Correctable) 01: Error Severity Level 1 (Recoverable) 10: Error Severity Level 2 (Fatal) 11: Reserved |

21.6.7.5 PCIERRSV: PCIe Error Severity Map Register

This register allows remapping of the PCIe errors to the IOH error severity.

| Register: PCIERRSV Device: 20 Function: 2 Offset: 094h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:6 | RV | 0 | Reserved |
| 5:4 | RWS | 10 | PCIe Fatal Error Severity Map 10: Map this PCIe error type to Error Severity 2 01: Map this PCIe error type to Error Severity 1 00: Map this PCIe error type to Error Severity 0 |
| 3:2 | RWS | 01 | PCIe Non-Fatal Error Severity Map same encoding as above |
| 1:0 | RWS | 00 | PCIe Correctable Error Severity Map same encoding as above |



21.6.7.6 THRERRSV: Thermal Error Severity Register

This register associates the detected thermal errors to an error severity level. An individual error is reported with the corresponding severity in this register. Software can program the error severity to one of the three severities supported by IOH. This register is sticky and can only be reset by PWRGOOD:

| Register: THRERRSV Device: 20 Function: 2 Offset: 098h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:16 | RV | 0h | Reserved |
| 15:12 | RWS | 0h | F3 - Throttling History (most recent valid CTHINT.THROTTLED bit) This field determines the error severity for the corresponding event. Four-bit encoding: xx00: Error Severity Level 0 (Correctable) xx01: Error Severity Level 1 (Recoverable) xx10: Error Severity Level 2 (Fatal) xx11: Reserved 00xx: Reserved 01xx: Send to the THERMALERT_N signal 10xx: Send to the THERMTRIP_N signal 11xx: Reserved |
| 11:8 | RWS | 1000 | F2 - Catastrophic Thermal Event This field determines the error severity for the corresponding event. Four-bit encoding: xx00: Error Severity Level 0 (Correctable) xx01: Error Severity Level 1 (Recoverable) xx10: Error Severity Level 2 (Fatal) xx11: Reserved 00xx: Reserved 01xx: Send to the THERMALERT_N signal 10xx: Send to the THERMTRIP_N signal 11xx: Reserved |
| 7:4 | RWS | 0h | F1 - TSMAX Updated This field determines the error severity for the corresponding event. Four-bit encoding: xx00: Error Severity Level 0 (Correctable) xx01: Error Severity Level 1 (Recoverable) xx10: Error Severity Level 2 (Fatal) xx11: Reserved 00xx: Reserved 01xx: Send to the THERMALERT_N signal 10xx: Send to the THERMTRIP_N signal 11xx: Reserved |
| 3:0 | RWS | 0100 | F0 - Thermal Alert This field determines the error severity for the corresponding event. Four-bit encoding: xx00: Error Severity Level 0 (Correctable) xx01: Error Severity Level 1 (Recoverable) xx10: Error Severity Level 2 (Fatal) xx11: Reserved 00xx: Reserved 01xx: Send to the THERMALERT_N signal 10xx: Send to the THERMTRIP_N signal 11xx: Reserved |



21.6.7.7 SYSMAP: System Error Event Map Register

This register maps the error severity detected by the IOH to the system events. When an error is detected by the IOH, its corresponding error severity determines which system event to generate according to this register.

| Register: SYSMAP Device: 20 Function: 2 Offset: 09Ch | | | |
|---|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:16 | RV | 0 | Reserved |
| 15 | RW1CS | 0 | ERRFRZSTS: Error Freeze Status: 0: Error Freeze was not invoked. 1: Error Freeze was invoked. Note: This register will capture the assertion of the error chip freeze signal, which is based on both an "error signal" and its respective "freeze component on error signal" enable both being asserted. |
| 14:12 | RW | 0h | ERRFRZ: Error Chip Freeze This feature is OR'd with the signal that drives the Chip Freeze Dfx signal. A chip freeze prohibits packets from entering or leaving the chip while maintaining protocol correctness. For effective use of this feature the system management software must assume that a measurable amount of time must pass before accessing the registers due to internal activity continuing for some time before coming to a halted state. 000: Do not freeze component. xx1: Freeze component on Error 0 Assertion (Correctable) x1x: Freeze component on Error 1 Assertion (Recoverable) 1xx: Freeze component on Error 2 Assertion (Fatal) After a freeze event occurs a SMBus software write with 0h will release the freeze condition. Since a reset may be required as a result of an error it may not be necessary to write a freeze release. Any combination of error select bits is possible, for example: A value of b110 enables chip freeze on any error that is not correctable. Note: When this Error Freeze signal is asserted, it has priority over the chip freeze start and stop selections. Chip freeze will remain on regardless of its start and stop, until ERRCHPFZ returns to a '0' value. |
| 11 | RV | 0 | Reserved |
| 10:8 | RWS | 000 | Severity 2 Error Map 110: Generate MCA 101: Generate CPEI 010: Generate NMI 001 Generate SMI/PMI 000: No inband message |
| 7 | RV | 0 | Reserved |
| 6:4 | RWS | 0 | Severity 1 Error Map same encoding as above |
| 3 | RV | 0 | Reserved |
| 2:0 | RWS | 0 | Severity 0 Error Map same encoding as above |



21.6.7.8 VIRAL: Viral Alert Register

This register provides the option to generate viral alert upon the detection of fatal error.

| Register: VIRAL Device: 20 Function: 2 Offset: 0A0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:3 | RV | 0 | Reserved |
| 2 | RWS | 0 | Fatal Viral Alert Enable Enable viral alert for Fatal Error. 0 - Disable Viral Alert for error severity 2. 1 - IOH goes viral when error severity 2 is set in the system event status register. |
| 1:0 | RV | 0 | Reserved |

21.6.7.9 ERRPINCTL: Error Pin Control Register

This register provides the option to configure an error pin to either as a special purpose error pin which is asserted based on the detected error severity, or as a general purpose output which is asserted based on the value in the ERRPINDAT. The assertion of the error pins can also be completely disabled by this register.

| Register: ERRPINCTL Device: 20 Function: 2 Offset: 0A4h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:6 | RV | 0 | Reserved |
| 5:4 | RWS | 0 | Error[2] Pin Assertion Control 11: Reserved. 10: Assert Error Pin when error severity 2 is set in the system event status reg. 01: Assert and Deassert Error pin according to error pin data register 00: Disable Error pin assertion |
| 3:2 | RWS | 0 | Error[1] Pin Assertion Control 11: Reserved. 10: Assert Error Pin when error severity 1 is set in the system event status reg. 01: Assert and Deassert Error pin according to error pin data register 00: Disable Error pin assertion |
| 1:0 | RWS | 0 | Error[0] Pin Assertion Control 11: Reserved. 10: Assert Error Pin when error severity 0 is set in the system event status reg. 01: Assert and Deassert Error pin according to error pin data register 00: Disable Error pin assertion |



21.6.7.10 ERRPINST: Error Pin Status Register

This register reflects the state of the error pin assertion. The status bit of the corresponding error pin is set upon the deassertion to assertion transition of the error pin. This bit is cleared by the software with writing 1 to the corresponding bit.

| Register: ERRPINST Device: 20 Function: 2 Offset: 0A8h | | | |
|---|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:3 | RV | 0 | Reserved |
| 2 | RW1CS | 0 | Error[2] Pin status This bit is set upon the transition of deassertion to assertion of the Error pin. Software write 1 to clear the status. |
| 1 | RW1CS | 0 | Error[1] Pin status |
| 0 | RW1CS | 0 | Error[0] Pin status |

21.6.7.11 ERRPINDAT: Error Pin Data Register

This register provides the data value when the error pin is configured as a general-purpose output.

| Register: ERRPINDAT Device: 20 Function: 2 Offset: 0ACh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:3 | RV | 0 | Reserved |
| 2 | RW | 0 | Error[2] Pin Data (applies when ERRPINCTL[5:4]=01; otherwise reserved) This bit acts as the general purpose output for the Error[2] pin. Error [2] pin value will follow the value programmed in Error[2] Pin Data register. This bit applies only when ERRPINCTL[5:4]=01; otherwise it is reserved. 0 - Deassert Error[2] pin 1 - Assert Error[2] pin This value only applies to the pin when ERRPINCTL[5:4]=01 |
| 1 | RW | 0 | Error[1] Pin Data (applies when ERRPINCTL[3:2]=01; otherwise reserved) |
| 0 | RW | 0 | Error[0] Pin Data (applies when ERRPINCTL[1:0]=01; otherwise reserved) |



21.6.7.12 VPPCTL: VPP Control

This register defines the control/command for PCA9555.

| Register: VPPCTL Device: 20 Function: 2 Offset: 0B0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 63:56 | RV | 0 | Reserved |
| 55 | RWS | 0 | VPP Reset Mode: 0: Power good reset will reset the VPP state machines and hard reset will cause the VPP state machine to terminate at the next 'logical' VPP stream boundary and then reset the VPP state machines 1: Both power good and hard reset will reset the VPP state machines |
| 54:44 | RWS | 0 | VPP Enable: When set, the VPP function for the corresponding root port is enabled. 54:44 - PCI[10:0] |
| 43:0 | RWS | 0 | VPP Address Assigns the VPP address of the device on the VPP interface and assigns the port address for the ports within the VPP device. There are more address bits than root ports so assignment must be spread across VPP ports. Addr Port Number Root Port [43:41] [40] PCIE[10] [39:37] [36] PCIE[9] [35:33] [32] PCIE[8] [31:29] [28] PCIE[7] [27:25] [24] PCIE[6] [23:21] [20] PCIE[5] [19:17] [16] PCIE[4] [15:13] [12] PCIE[3] [11:9] [8] PCIE[2] [7:5] [4] PCIE[1] [3:1] [0] PCIE[0] |

21.6.7.13 VPPSTS: VPP Status Register

This register defines the status from PCA9555.

| Register: VPPSTS Device: 20 Function: 2 Offset: 0B8h | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:1 | RV | 0 | Reserved |
| 0 | RW1CS | 00 | VPP Port Error Happened That is, an unexpected STOP of NACK was seen on the VPP port |



21.6.7.14 PRSTRDY: Reset Release Ready

This register is used to indicate to BMC that IOH has received CPU_RST_DONE_ACK from the ICH, and that BMC can release the reset. Note: This register applies only to the legacy IOH where an ICH is connected. For non-legacy IOH, the corresponding bit in this register is always set to 0.

| Register: PRSTRDY Device: 20 Function: 2 Offset: 0C0h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:1 | RV | 0 | Reserved |
| 0 | RW1C | 0 | Reset Release Ready This bit indicates that IOH has received CPU_RST_DONE_ACK from the ICH and that BMC can release the reset. 0 - Keep reset asserted 1 - BMC release reset |

21.6.7.15 GENMCA: Generate MCA Register

This register is used to generate MCA interrupt to CPU by firmware.

| Register: GENMCA Device: 20 Function: 2 Offset: 0C4h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:1 | RV | 0 | Reserved |
| 0 | RWS | 0 | Generate MCA When this bit is set and transition from 0 to 1, IOH dispatches a MCA interrupt defined in the QPIPMCAC register to the CPU. This bit is cleared by hardware when IOH has dispatched MCA to the Intel QuickPath Interconnect link. |

21.6.7.16 GENVIRAL: Generate Viral

This register is used to generate Viral alert to CPU by firmware and clear Viral.

| Register: GENVIRAL Device: 20 Function: 2 Offset: 0C8h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3 | RWO | 0 | Viral_Clear_disable When VIRAL_CLEAR_DISABLE = 1 – this will not allow the clearing of viral alert by setting VIRAL_CLEAR bit; - Eventually, the Viral Alert will be cleared once all the Fatal Error and VIRAL_status (RW1CS) is cleared by software. When VIRAL_CLEAR_DISABLE = 0 – this will clear the Viral alert generated in the system (Viral broadcast will be stopped) by setting the VIRAL_CLEAR bit – but still the VIRAL_STATUS will be set until software clears it. |
| 2 | RW | 0 | Viral_Clear This bit is active when VIRAL_CLEAR_DIS = 0; by setting this bit Viral alert generated in the system will be cleared. |



| Register: GENVIRAL Device: 20 Function: 2 Offset: 0C8h | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 1 | RW1CS | 0 | Viral_Status This bit is set when IOH is in viral mode. |
| 0 | RWS | 0 | Generate Viral Alert When this bit is set and transition from 0 to 1, IOH sets Intel QuickPath Interconnect cluster(s) to viral. This bit is cleared by hardware when IOH has set viral alert on Intel QPI cluster(s). |

21.6.7.17 SYRE: System Reset

This register controls IOH reset behavior. Any resets produced by a write to this register must be delayed until the configuration write is completed on the PCIe/ESI, Intel QuickPath Interconnect, SMBUS, and JTAG interfaces.

There is no “SOFT RESET” bit in this register. That function is invoked through the ESI interface. There are no Intel QPI:PCI Express gear ratio definitions in this register. The Intel QuickPath Interconnect frequencies are specified in the FREQ register. The PCI Express frequencies are automatically negotiated inband.

| Register: SYRE Device: 20 Function: 2 Offset: 0CCh | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:15 | RV | 0 | Reserved |
| 14 | RV | 0 | S5 1 - Translate ESI.GO_S3 to QPI.SpcPMReq (S5) 0 - Forward ESI.GO_S3 to QPI.SpcPMReq(S3) |
| 13:12 | RV | 0 | Reserved |
| 11 | RW | 0 | RSTMSK 0 - the IOH will perform the appropriate internal handshakes on RST_N signal transitions to progress through the hard reset. 1 - IOH ignores RST_N, unaffected by the RST_N assertion |
| 10 | RW | 0 | CPURESET 1 - IOH asserts RESETO_N Once this bit has been set to initiate RESETO_N to the CPU, it will not clear itself. To initiate the second RESETO_N, software must write '0' for manual clear then followed by a write '1'. This bit will not accept a '1' written to it while LT.SENTER.STS is set. These behaviors are described in the RESET chapter. |
| 9:1 | RV | 0 | Reserved |
| 0 | RWS | 1 | Enable CPU BIST 1- Enable CPU BIST 0- Disable CPU BIST This bit controls whether or not BIST is run in the CPU on reset. It's value will correspond to the BIST value in the POC exchanged from IOH on Intel QuickPath Interconnect. This value will only make a difference in CPU's that observe POC (like Xeon). By default BIST is disabled. If BIST is desired, then after this bit is set the CPU must be reset to cause the CPU to capture the new value. |



21.6.7.18 FREQ: Frequencies

This register defines the Intel® QuickPath Interconnect frequency. The QPIFREQSEL[1:0] straps determines the Intel® QuickPath Interconnect link:core frequency ratio. This FREQ register is read-only, and it indicates the PRESENT frequency of the links.

Note: This register is sticky. The frequency bits and can only be latched in at PWRGOOD.

| Register: FREQ Device: 20 Function: 2 Offset: 0D0h | | | |
|---|------|----------------------|---|
| Bit | Attr | Default | Description |
| 31:2 | RV | 0h | Reserved |
| 1:0 | RO | STRAP: QPIFREQSEL | QPIFREQSEL: Intel QPI High Frequency "00" = 4.800 GT/s "01" = 5.860 GT/s (Intel Xeon processor 7500 series only) "10" = 6.400 GT/s "11" = <i>Reserved</i> This is the value of the QPIFREQSEL signals sampled at PWRGOOD. |

21.6.7.19 CAPTIM: Cap Timer

This register sets the cap timer count value.

| Register: CAPTIM Device: 20 Function: 2 Offset: 0E8h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:14 | RV | 0 | Reserved |
| 13:0 | RWS | 7FFh | CAPTIM: Cap Timer Value Cap timer value. When enabled, a detected outbound ESI transaction will start the timer. The returning read data completion is held in the core until the expiration of the counter. After the transaction is released the counter is re-loaded with this count value (or cleared depending on implementation). The counter is free-running until the CAPTIMEN bit is cleared. |

21.6.7.20 EOI_CTRL: Global EOI Control Register

| Register: EOI_CTRL Device: 20 Function: 2 Offset: FCh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:1 | RV | 00h | <i>Reserved.</i> |
| 0 | RW | 0 | Drop_EOI: 0: EOI messages from Intel QuickPath Interconnect are broadcasted to root/ESI ports and IOxAPIC per the normal rules for EOI broadcast. 1: EOI messages from Intel® QPI are simply dropped and not broadcast to root/ESI ports or the integrated IOxAPIC |



21.7 Global Error Registers

Table 21-14. IOH Control/Status & Global Error Register Map (Dev 20, Function 2)

| | | | |
|----------------------------|------|------------|------|
| RESERVED PCIe Header space | 100h | | 180h |
| | 104h | | 184h |
| | 108h | | 188h |
| | 10Ch | | 18Ch |
| | 110h | | 190h |
| | 114h | | 194h |
| | 118h | | 198h |
| | 11Ch | | 19Ch |
| | 120h | | 1A0h |
| | 124h | | 1A4h |
| | 128h | | 1A8h |
| | 12Ch | | 1ACh |
| | 130h | | 1B0h |
| | 134h | | 1B4h |
| | 138h | | 1B8h |
| | 13Ch | | 1BCh |
| | 140h | GNERRST | 1C0h |
| | 144h | GFERRST | 1C4h |
| | 148h | GERRCTL | 1C8h |
| | 14Ch | GSYSST | 1CCh |
| | 150h | GSYSCTL | 1D0h |
| | 154h | GTIME | 1D4h |
| | 158h | | 1D8h |
| | 15Ch | GFFERRST | 1DCh |
| | 160h | GFFERRTIME | 1E0h |
| | 164h | | 1E4h |
| | 168h | GFNERST | 1E8h |
| | 16Ch | GNFERRST | 1ECh |
| | 170h | GNFERRTIME | 1F0h |
| | 174h | | 1F4h |
| | 178h | GNNERRST | 1F8h |
| | 17Ch | | 1FCh |



21.7.1 Global Error Registers

21.7.1.1 GNERRST: Global Non-Fatal Error Status Register

This register indicates the status of non-fatal error reported to the IOH global error logic. An individual error status bit that is set indicates that a particular local interface has detected an error.

| Register: GNERRST Device: 20 Function: 2 Offset: 1C0h | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:26 | RV | 0 | Reserved |
| 25 | RW1CS | 0 | Intel VT-d Error Status This bit indicates that IOH has detected an Intel VT-d related error. |
| 24 | RW1CS | 0 | Miscellaneous Error Status This bit indicates that IOH has detected a miscellaneous error. |
| 23 | RW1CS | 0 | IOH Core Error Status This bit indicates that IOH core has detected an error. |
| 22 | RV | 0 | Reserved |
| 21 | RW1CS | 0 | Thermal Error Status This bit indicates that IOH detected thermal error. |
| 20 | RW1CS | 0 | ESI Error Status This bit indicates that IOHESI port 0 has detected an error. |
| 19:16 | RV | 0 | Reserved |
| 15 | RW1CS | 0 | PCIe [10] Error Status PCIe port 10 has detected an error. |
| 14 | RW1CS | 0 | PCIe [9] Error Status PCIe port 9 has detected an error. |
| 13 | RW1CS | 0 | PCIe [8] Error Status PCIe port 8 has detected an error. |
| 12 | RW1CS | 0 | PCIe [7] Error Status PCIe port 7 has detected an error. |
| 11 | RW1CS | 0 | PCIe [6] Error Status PCIe port 6 has detected an error. |
| 10 | RW1CS | 0 | PCIe [5] Error Status PCIe port 5 has detected an error. |
| 9 | RW1CS | 0 | PCIe [4] Error Status PCIe port 4 has detected an error. |
| 8 | RW1CS | 0 | PCIe [3] Error Status PCIe port 3 has detected an error. |
| 7 | RW1CS | 0 | PCIe [2] Error Status PCIe port 2 has detected an error. |
| 6 | RW1CS | 0 | PCIe [1] Error Status PCIe port 1 has detected an error. |
| 5 | RW1CS | 0 | PCIe [0] Error Status PCIe port 0 has detected an error. PCIe[0] is associated with ESI. |
| 4 | RV | 0 | Reserved |



| Register: GNERRST Device: 20 Function: 2 Offset: 1C0h | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 3 | RW1CS | 0 | QPI[1] Protocol Error Status This bit indicates that the Intel QuickPath Interconnect protocol layer port 1 has detected an error |
| 2 | RW1CS | 0 | QPI[0] Protocol Error Status This bit indicates that the Intel QuickPath Interconnect protocol layer port 0 has detected an error |
| 1 | RW1CS | 0 | QPI[1] Error Status This bit indicates that QPI[1] port has detected an error |
| 0 | RW1CS | 0 | QPI[0] Error Status This bit indicates that QPI[0] port has detected an error |

21.7.1.2 GFERRST: Global Fatal Error Status Register

This register indicates the fatal error reported to the IOH global error logic. An individual error status bit that is set indicates that a particular local interface has detected an error.

| Register: GFERRST Device: 20 Function: 2 Offset: 1C4h | | | |
|--|--------|---------|---|
| Bit | Attr | Default | Description |
| 31:26 | RV | 0 | Reserved |
| 25 | RW1CS | 0 | VTd Error Status This bit indicates that IOH has detected a Intel VT-d related error. |
| 24 | RW1CS | 0 | Miscellaneous Error Status This bit indicates that IOH has detected a miscellaneous error. |
| 23 | RW1CS | 0 | IOH Core Error Status This bit indicates that IOH core has detected an error. |
| 22 | RV | 0 | Reserved |
| 21 | RW1CS | 0 | Thermal Error Status This bit indicates that IOH has detected thermal error. |
| 20 | RW1CST | 0 | ESI Error Status This bit indicates that IOHESI port 0 has detected an error. |
| 19:16 | RV | 0 | Reserved |
| 15 | RW1CS | 0 | PCIe [10] Error Status PCIe port 10 has detected an error. |
| 14 | RW1CS | 0 | PCIe [9] Error Status PCIe port 9 has detected an error. |
| 13 | RW1CS | 0 | PCIe [8] Error Status PCIe port 8 has detected an error. |
| 12 | RW1CS | 0 | PCIe [7] Error Status PCIe port 7 has detected an error. |
| 11 | RW1CS | 0 | PCIe [6] Error Status PCIe port 6 has detected an error. |
| 10 | RW1CS | 0 | PCIe [5] Error Status PCIe port 5 has detected an error. |



| Register: GFERRST Device: 20 Function: 2 Offset: 1C4h | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 9 | RW1CS | 0 | PCIe [4] Error Status PCIe port 4 has detected an error. |
| 8 | RW1CS | 0 | PCIe [3] Error Status PCIe port 3 has detected an error. |
| 7 | RW1CS | 0 | PCIe [2] Error Status PCIe port 2 has detected an error. |
| 6 | RW1CS | 0 | PCIe [1] Error Status PCIe port 1 has detected an error. |
| 5 | | | PCIe [0] Error Status PCIe port 0 has detected an error. PCIe[0] is associated with ESI. |
| 4 | RV | 0 | Reserved |
| 3 | RW1CS | 0 | QPI[1] Protocol Error Status This bit indicates that the Intel QuickPath Interconnect protocol layer port 1 has detected an error |
| 2 | RW1CS | 0 | QPI[0] Protocol Error Status This bit indicates that the Intel QuickPath Interconnect protocol layer port 0 has detected an error |
| 1 | RW1CS | 0 | QPI[1] Error Status This bit indicates that QPI[1] port has detected an error |
| 0 | RW1CS | 0 | QPI[0] Error Status This bit indicates that QPI[0] port has detected an error |

21.7.1.3 GERRCTL: Global Error Control

This register controls the reporting of errors detected by the IOH local interfaces. An individual error control bit that is set disable (masks) error reporting of the particular local interface; software may set or clear the control bit. This register is not sticky and it gets reset to default upon system reset.

Note that bit fields in this register can become reserved depending on the port configuration. For example, if the PCIe port is configured as 2X8 ports, then only the corresponding PCI-EX8 bit fields are valid; other bits are unused and reserved.

Please note by default error reporting is enabled and setting global error control register will disable (mask) errors reporting from the local interface to global error status register. If the error reporting is disabled (masked) in this register, all errors from the corresponding local interface will not set any of the global error status bits.



| Register: GERRCTL Device: 20 Function: 2 Offset: 1C8h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:26 | RV | 0 | Reserved |
| 25 | RW | 0 | VTd Error Enable This bit controls the VTd related error. 0: Error reporting enabled 1: Error reporting disabled |
| 24 | RW | 0 | Miscellaneous Error Enable This bit controls the miscellaneous error detected in the IOH. 0: Error reporting enabled 1: Error reporting disabled |
| 23 | RW | 0 | IOH Core Error Enable This bit controls the error detected in the IOH Core. 0: Error reporting enabled 1: Error reporting disabled |
| 22 | RV | 0 | Reserved |
| 21 | RW | 0 | Thermal Error Enable This bit controls the detected Thermal error. in the IOH 0: Error reporting enabled 1: Error reporting disabled |
| 20 | RW | 0 | ESI Error Enable This bit controls the error detected in the ESI Port. 0: Error reporting enabled 1: Error reporting disabled |
| 19:16 | RV | 0 | Reserved |
| 15 | RW | 0 | PCIe [10] Error Enable This bit controls the error detected in the PCIe port 10 0: Error reporting enabled 1: Error reporting disabled |
| 14 | RW | 0 | PCIe [9] Error Enable This bit controls the error detected in the PCIe port 9 0: Error reporting enabled 1: Error reporting disabled |
| 13 | RW | 0 | PCIe [8] Error Enable This bit controls the error detected in the PCIe port 8 0: Error reporting enabled 1: Error reporting disabled |
| 12 | RW | 0 | PCIe [7] Error Enable This bit controls the error detected in the PCIe port 7 0: Error reporting enabled 1: Error reporting disabled |
| 11 | RW | 0 | PCIe [6] Error Enable This bit controls the error detected in the PCIe port 6 0: Error reporting enabled 1: Error reporting disabled |
| 10 | RW | 0 | PCIe [5] Error Enable This bit controls the error detected in the PCIe port 5 0: Error reporting enabled 1: Error reporting disabled |



| Register: GERRCTL Device: 20 Function: 2 Offset: 1C8h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 9 | RW | 0 | PCIe [4] Error Enable This bit controls the error detected in the PCIe port 4 0: Error reporting enabled 1: Error reporting disabled |
| 8 | RW | 0 | PCIe [3] Error Enable This bit controls the error detected in the PCIe port 3 0: Error reporting enabled 1: Error reporting disabled |
| 7 | RW | 0 | PCIe [2] Error Enable This bit controls the error detected in the PCIe port 2 0: Error reporting enabled 1: Error reporting disabled |
| 6 | RW | 0 | PCIe [1] Error Enable This bit controls the error detected in the PCIe port 1 0: Error reporting enabled 1: Error reporting disabled |
| 5 | | | PCIe [0] Error Enable This bit controls the error detected in associated with ESI. 0: Error reporting enabled 1: Error reporting disabled |
| 4 | RV | 0 | Reserved |
| 3 | RW | 0 | QPI[1] Protocol Error Enable This bit controls the error detected in the Intel QPI Protocol Layer Port 1 0: Error reporting enabled 1: Error reporting disabled |
| 2 | RW | 0 | QPI[0] Protocol Error Enable This bit controls the error detected in the Intel QPI Protocol Layer Port 0 0: Error reporting enabled 1: Error reporting disabled |
| 1 | RW | 0 | QPI[1] Error Enable This bit controls the error detected in the Intel QPI Port 1 0: Error reporting enabled 1: Error reporting masked |
| 0 | RW | 0 | QPI[0] Error Enable This bit controls the error detected in the Intel QPI Port 0 0: Error reporting enabled 1: Error reporting disabled |



21.7.1.4 GSYSST: Global System Event Status Register

This register indicates the error severity signaled by the IOH global error logic. Setting of an individual error status bit indicates that the corresponding error severity has been detected by the IOH.

| Register: GSYSST Device: 20 Function: 2 Offset: 1CCh | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:5 | RV | 0 | Reserved |
| 4 | ROS | 0 | Severity 4 Error Status When set, IOH has detected an error of error severity 4 |
| 3 | ROS | 0 | Severity 3 Error Status When set, IOH has detected an error of error severity 3 |
| 2 | ROS | 0 | Severity 2 Error Status When set, IOH has detected an error of error severity 2 |
| 1 | ROS | 0 | Severity 1 Error Status When set, IOH has detected an error of error severity 1 |
| 0 | ROS | 0 | Severity 0 Error Status When set, IOH has detected an error of error severity 0 |

21.7.1.5 GSYSCTL: Global System Event Control Register

The system event control register controls the reporting the errors indicated by the system event status register. When cleared, the error severity does not cause the generation of the system event. When set, detection of the error severity generates system event(s) according to system event map register (SYSMAP).

| Register: GSYSCTL Device: 20 Function: 2 Offset: 01D0 | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:5 | RV | 0 | Reserved |
| 4 | RW | 0 | Severity 4 Error enable 0 - Disable system event reporting of the error severity 1 - Enable system event reporting of the error severity Note setting 0 of this bit does not prevent setting of the system event status register. |
| 3 | RW | 0 | Severity 3 Error enable 0 - Disable system event reporting of the error severity 1 - Enable system event reporting of the error severity Note setting 0 of this bit does not prevent setting of the system event status register. |
| 2 | RW | 0 | Severity 2 Error enable 0 - Disable system event reporting of the error severity 1 - Enable system event reporting of the error severity Note setting 0 of this bit does not prevent setting of the system event status register. |



| Register: GSYSCTL Device: 20 Function: 2 Offset: 01D0 | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:5 | RV | 0 | Reserved |
| 1 | RW | 0 | Severity 1 Error enable 0 - Disable system event reporting of the error severity 1 - Enable system event reporting of the error severity Note setting 0 of this bit does not prevent setting of the system event status register. |
| 0 | RW | 0 | Severity 0 Error enable 0 - Disable system event reporting of the error severity 1 - Enable system event reporting of the error severity Note setting 0 of this bit does not prevent setting of the system event status register. |

21.7.1.6 GTIME: Global Error Timer Register

Global Error Timer register is a free running 64-bit counter and will indicate the current value of the 64-bit counter. This counter is reset to 0 by PWRGOOD. Once out of PWRGOOD reset, the counter begins to run.

| Register: GTIME Device: 20 Function: 2 Offset: 1D4h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 63:0 | RWS | 0 | Error Log Time Stamp This is the 64-bit free running counter using core clock. Core clock are: - 400 Mhz if CSI freq is 6.4 GHz - 367 Mhz if CSI freq is 5.867 GHz - 300 Mhz if CSI freq is 4.8 GHz - 200 MHz if CSI freq is 3.2 GHz |

21.7.1.7 GFFERRST: Global Fatal FERR Status Register

| Register: GFFERRST Device: 20 Function: 2 Offset: 1DCh | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:26 | RV | 0 | Reserved |
| 25:0 | ROS | 0 | Global Error Status Log This filed logs the global error status register content when the first fatal error is reported. This has the same format as the global error status register (GERRST). Note: If two fatal errors occur in the same cycle, both errors will be logged. |



21.7.1.8 GFFERRTIME: Global Fatal FERR Time Stamp Register

| Register: GFFERRTIME Device: 20 Function: 2 Offset: 1E0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 63:0 | ROS | 0 | Global Error Time Stamp The time stamp register logs the 64-bit free running counter when the first error was logged. |

21.7.1.9 GFNERRST: Global Fatal NERR Status Register

| Register: GFNERRST Device: 20 Function: 2 Offset: 1E8h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:26 | RV | 0 | Reserved |
| 25:0 | ROS | 0 | Global Error Status Log This field logs the global error status register content when the next fatal error is reported. This has the same format as the global error status register (GERRST). Note: Only second error gets logged into GFNERRST (subsequent error does not get logged into GFNERRST). |

21.7.1.10 GNFERRST: Global Non-Fatal FERR Status Register

| Register: GNFERRTIME Device: 20 Function: 2 Offset: 1ECh | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:26 | RV | 0 | Reserved |
| 25:0 | ROS | 0 | Global Error Status Log This field logs the global error status register content when the first non-fatal error is reported. This has the same format as the global error status register (GERRST). Note: If two non-fatal errors occur in the same cycle, both errors will be logged. |



21.7.1.11 GNFERRTIME: Global Non-Fatal FERR Time Stamp Register

| Register: GNFERRTIME Device: 20 Function: 2 Offset: 1F0h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 63:0 | ROS | 0 | Time Stamp The time stamp register logs the 64-bit free running counter when the first non-fatal error was logged. |

21.7.1.12 GNNERRST: Global Non-Fatal NERR Status Register

| Register: GNNERRST Device: 20 Function: 2 Offset: 1F8h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:26 | RV | 0 | Reserved |
| 25:0 | ROS | 0 | Global Error Status Log This field logs the global error status register content when the subsequent non-fatal error is reported. This has the same format as the global error status register (GERRST). Note: Only second error gets logged into GNFERRST (subsequent error does not get logged into GNFERRST). |



21.8 IOH Local Error Registers

Table 21-15. IOH Local Error Map #1 (Dev 20, Function 2)

| | | | |
|----------------|------|----------------|------|
| QPIOERRST | 200h | QPI1ERRST | 280h |
| QPIOERRCTL | 204h | QPI1ERRCTL | 284h |
| QPIOFFERRST | 208h | QPI1FFERRST | 288h |
| QPIOFNERRST | 20Ch | QPI1FNERRST | 28Ch |
| QPIONFERRST | 210h | QPI1NFERRST | 290h |
| QPIONNERRST | 214h | QPI1NNERRST | 294h |
| QPIOERRCNTSEL | 218h | QPI1ERRCNTSEL | 298h |
| QPIOERRCNT | 21Ch | QPI1ERRCNT | 29Ch |
| | 220h | | 2A0h |
| | 224h | | 2A4h |
| | 228h | | 2A8h |
| | 22Ch | | 2ACh |
| QPIPOERRST | 230h | QPIP1ERRST | 2B0h |
| QPIPOERRCTL | 234h | QPIP1ERRCTL | 2B4h |
| QPIPOFFERRST | 238h | QPIP1FFERRST | 2B8h |
| QPIPOFNERRST | 23Ch | QPIP1FNERRST | 2BCh |
| QPIPOFFERRHD | 240h | QPIP1FFERRHD | 2C0h |
| | 244h | | 2C4h |
| | 248h | | 2C8h |
| | 24Ch | | 2CCh |
| QPIPONFERRST | 250h | QPIP1NFERRST | 2D0h |
| QPIPONNERRST | 254h | QPIP1NNERRST | 2D4h |
| QPIPONFERRHD | 258h | QPIP1NFERRHD | 2D8h |
| | 25Ch | | 2DCh |
| | 260h | | 2E0h |
| | 264h | | 2E4h |
| QPIPOERRCNTSEL | 268h | QPIP1ERRCNTSEL | 2E8h |
| QPIPOERRCNT | 26Ch | QPIP1ERRCNT | 2ECh |
| | 270h | | 2F0h |
| | 274h | | 2F4h |
| | 278h | | 2F8h |
| | 27Ch | | 2FCh |



Table 21-16. IOH Local Error Map #2 (Dev 20, Function 2)

| | | | |
|--------------|------|-------------|------|
| IOHERRST | 300h | MIERRST | 380h |
| IOHERRCTL | 304h | MIERRCTL | 384h |
| IOHFFERRST | 308h | MIFFERRST | 388h |
| IOHFFERRHD | 30Ch | MIFFERRHD | 38Ch |
| | 310h | | 390h |
| | 314h | | 394h |
| | 318h | | 398h |
| IOHFNERRST | 31Ch | MIFNERRST | 39Ch |
| IOHNFERRST | 320h | MINFERRST | 3A0h |
| IOHNFERRHD | 324h | MINFERRHD | 3A4h |
| | 328h | | 3A8h |
| | 32Ch | | 3ACh |
| | 330h | | 3B0h |
| IOHNNERRST | 334h | MINNERRST | 3B4h |
| | 338h | | 3B8h |
| IOHERRCNTSEL | 33Ch | MIERRCNTSEL | 3BCh |
| IOHERRCNT | 340h | MIERRCNT | 3C0h |
| | 344h | | 3C4h |
| | 348h | | 3C8h |
| | 34Ch | | 3CCh |
| | 350h | | 3D0h |
| | 354h | | 3D4h |
| | 358h | | 3D8h |
| | 35Ch | | 3DCh |
| THRERRST | 360h | | 3E0h |
| THRERRCTL | 364h | | 3E4h |
| THRFFERRST | 368h | | 3E8h |
| THRFNERRST | 36Ch | | 3ECh |
| THRNFERRST | 370h | | 3F0h |
| THRNNERRST | 374h | | 3F4h |
| THRERRCNTSEL | 378h | | 3F8h |
| THRERRCNT | 37Ch | | 3FCh |



Table 21-17. IOH Local Error Map #2 (Dev 20, Function 2, Page 4 of 4)

| | | | |
|----------------|--|----------------|--|
| QPI0FERRFLIT0 | 400h 404h 408h | QPI1FERRFLIT0 | 480h 484h 488h |
| QPI0FERRFLIT1 | 40Ch 410h 414h | QPI1FERRFLIT1 | 48Ch 490h 494h |
| | 418h 41Ch 420h 424h 428h 42Ch | | 498h 49Ch 4A0h 4A4h 4A8h 4ACh |
| QPI0PFERRFLIT0 | 430h 434h 438h | QPI1PFERRFLIT0 | 4B0h 4B4h 4B8h |
| QPI0PFERRFLIT1 | 43Ch 440h 444h | QPI1PFERRFLIT1 | 4BCh 4C0h 4C4h |
| QPI0PFERRFLIT2 | 448h 44Ch 450h | QPI1PFERRFLIT2 | 4C8h 4CCh 4D0h |
| | 454h 458h 45Ch 460h 464h 468h 46Ch 470h 474h 478h 47Ch | | 4D4h 4D8h 4DCh 4E0h 4E4h 4E8h 4ECh 4F0h 4F4h 4F8h 4FCh |



21.8.1 IOH Local Error Register

21.8.1.1 QPI[1:0]ERRST: Intel QPI Error Status Register

This register indicates the error detected by the Intel QuickPath Interconnect local interface. See [Section 16.7, “IOH Error Handling Summary”](#) for more details on error type. For details on usage of local error logging, see [Section 16.4.3.1, “Local Error Registers”](#).

| Register: QPI[1:0]ERRST Device: 20 Function: 2 Offset: 280h, 200h | | | |
|--|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:13 | RV | 0 | Reserved |
| 12 | RW1CS | 0 | D4 - Intel QPI Link Internal Parity Error |
| 11 | RW1CS | 0 | D3 - Intel QPI Link Layer Control Error |
| 10 | RW1CS | 0 | C0 - Intel QPI Link Layer detected CRC error - unsuccessful link level retry - entered LLR abort state |
| 9 | RW1CS | 0 | B1 - Intel QPI Link Layer detected CRC error - successful link level retry after PHY reinit |
| 8 | RW1CS | 0 | B0 - Intel QPI Link Layer CRC - successful link level retry. - B0 bit can be set for a successful link level retry following a physical layer reset. This bit can get set even if CRC errors are not logged. |
| 7 | RW1CS | 0 | DG - Intel QPI Link Layer Detected unsupported/undefined msgclass/opcode/vn packet Error - See QPI[1:0]FERRFLIT0 for flit info. |
| 6 | RW1CS | 0 | B5 - Potential Spurious CRC error on L0s/L1 Exit - See QPI[1:0]FERRFLIT0 for 8bit-CRC flit info, and for 16-bit-CRC GB flit info and QPI[1:0]FERRFLIT1 for 16-bit-CRC GA flit info. |
| 5 | RW1CS | 0 | B6 - Intel QPI Link Layer CRC error - See QPI[1:0]FERRFLIT0 for 8bit-CRC flit info, and for 16-bit-CRC GB flit info and QPI[1:0]FERRFLIT1 for 16-bit-CRC GA flit info. |
| 4 | RW1CS | 0 | D2 - Intel QPI Physical Layer Initialization Failure |
| 3 | RW1CS | 0 | D1 - Intel QPI Physical Layer Detected Latency Buffer Rollover |
| 2 | RW1CS | 0 | D0 - Intel QPI Physical Layer Detected Drift Buffer Alarm |
| 1 | RW1CS | 0 | C7 - Intel QPI Physical Layer Reset Successful with Width Change |
| 0 | RW1CS | 0 | B2 - Intel QPI Physical Layer Successful Reset at same Width |



21.8.1.2 QPI[1:0]ERRCTL: Intel QuickPath Interconnect Error Control Register

This register enable the error status bit setting for an Intel® QuickPath Interconnect detected error. Setting of the bit enables the setting of the corresponding error status bit in QPIERRST register. If the bit is cleared, the corresponding error status will not be set.

| Register: QPI[1:0]ERRCTL Device: 20 Function: 2 Offset: 284h, 204h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:13 | RV | 0 | <i>Reserved</i> |
| 12 | RWS | 0 | D4 - Intel QPI Link Internal Parity Error Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 11 | RWS | 0 | D3 - Intel QPI Link Layer Control Error Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 10 | RWS | 0 | C0 - Intel QPI Link Layer detected CRC error - unsuccessful link level retry - entered LLR abort state Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 9 | RWS | 0 | B1 - Intel QPI Link Layer detected CRC error - successful link level retry after PHY reinit Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 8 | RWS | 0 | B0 - Intel QPI Link Layer CRC - successful link level retry Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 7 | RWS | 0 | DG - Intel QPI Link Layer Detected unsupported/undefined msgclass/opcode/vn packet Error Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 6 | RWS | 0 | B5 - Potential Spurious CRC error on L0s/L1 Exit Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 5 | RWS | 0 | B6 - Intel QPI Link Layer CRC error Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 4 | RWS | 0 | D2 - Intel QPI Physical Layer Initialization Failure Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 3 | RWS | 0 | D1 - Intel QPI Physical Layer Detected Latency Buffer Rollover Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 2 | RWS | 0 | D0 - Intel QPI Physical Layer Detected Drift Buffer Alarm Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 1 | RWS | 0 | C7 - Intel QPI Physical Layer Reset Successful with Reduced Width Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 0 | RWS | 0 | B2 - Intel QPI Physical Layer Successful Reset at same Width Enable 0 - Disable error status logging 1 - Enable Error status logging |



21.8.1.3 Intel QuickPath Interconnect Error Log Register

This register logs the information associated with the reporting of Intel QuickPath Interconnect errors. There are two sets of error log registers of identical format: FERR logs the first occurrence of an error, and NERR logs the next occurrence of the error. An individual error is reported with the corresponding severity in this register. Software can program the error severity to one of the three severities supported by IOH. This register is sticky and can only be reset by PWRGOOD. Clearing of the QPI**ERRST is done by clearing the corresponding QPIERRST bits.

21.8.1.4 QPI[1:0]FFERRST: Intel QuickPath Interconnect Fatal FERR Status Register

The error status log indicates which error is causing the report of the first fatal error event.

| Register: QPI[1:0]FFERRST Device: 20 Function: 2 Offset: 288h, 208h | | | |
|--|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:13 | RV | 0 | Reserved |
| 12 | RW1CS | 0 | D4 - Intel QPI Link Internal Parity Error |
| 11 | RW1CS | 0 | D3 - Intel QPI Link Layer Control Error |
| 10 | RW1CS | 0 | C0 - Intel QPI Link Layer detected CRC error - unsuccessful link level retry - entered LLR abort state |
| 9 | RW1CS | 0 | B1 - Intel QPI Link Layer detected CRC error - successful link level retry after PHY reinit |
| 8 | RW1CS | 0 | B0 - Intel QPI Link Layer CRC - successful link level retry |
| 7 | RW1CS | 0 | DG - Intel QPI Link Layer Detected unsupported/undefined msgclass/opcode/vn packet Error |
| 6 | RW1CS | 0 | B5 - Potential Spurious CRC error on L0s/L1 Exit |
| 5 | RW1CS | 0 | B6 - Intel QPI Link Layer CRC error |
| 4 | RW1CS | 0 | D2 - Intel QPI Physical Layer Initialization Failure |
| 3 | RW1CS | 0 | D1 - Intel QPI Physical Layer Detected Latency Buffer Rollover |
| 2 | RW1CS | 0 | D0 - Intel QPI Physical Layer Detected Drift Buffer Alarm |
| 1 | RW1CS | 0 | C7 - Intel QPI Physical Layer Reset Successful with Reduced Width |
| 0 | RW1CS | 0 | B2 - Intel QPI Physical Layer Successful Reset at same Width |

21.8.1.5 QPI[1:0]FNERRST: Intel QuickPath Interconnect Fatal NERR Status Registers

The error status log indicates which error is causing the report of the next fatal error events.

| Register: QPI[1:0]FNERRST Device: 20 Function: 2 Offset: 28Ch, 20Ch | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:13 | RV | 0 | Reserved |
| 12 | RW1CS | 0 | D4 - Intel QPI Link Internal Parity Error |



| Register: QPI[1:0]FNERRST Device: 20 Function: 2 Offset: 28Ch, 20Ch | | | |
|--|-------|---------|---|
| Bit | Attr | Default | Description |
| 11 | RW1CS | 0 | D3 - Intel QPI Link Layer Control Error |
| 10 | RW1CS | 0 | C0 - Intel QPI Link Layer detected CRC error |
| 9 | RW1CS | 0 | B1 - Intel QPI Link Layer detected CRC error |
| 8 | RW1CS | 0 | B0 - Intel QPI Link Layer CRC - successful link level retry |
| 7 | RW1CS | 0 | DG - Intel QPI Link Layer Detected unsupported/undefined msgclass/opcode/vn packet Error |
| 6 | RW1CS | 0 | B5 - Potential Spurious CRC error on L0s/L1 Exit |
| 5 | RW1CS | 0 | B6 - Intel QPI Link Layer CRC error |
| 4 | RW1CS | 0 | D2 - Intel QPI Physical Layer Initialization Failure |
| 3 | RW1CS | 0 | D1 - Intel QPI Physical Layer Detected Latency Buffer Rollover |
| 2 | RW1CS | 0 | D0 - Intel QPI Physical Layer Detected Drift Buffer Alarm |
| 1 | RW1CS | 0 | C7 - Intel QPI Physical Layer Reset Successful with Reduced Width |
| 0 | RW1CS | 0 | B2 - Intel QPI Physical Layer Successful Reset at same Width |

21.8.1.6 QPI[1:0]NFERRST: Intel QuickPath Interconnect Non-Fatal FERR Status Registers

The error status log indicates which error is causing the report of the first non-fatal error event.

| Register: QPI[1:0]NFERRST Device: 20 Function: 2 Offset: 290h, 210h | | | |
|--|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:13 | RV | 0 | Reserved |
| 12 | RW1CS | 0 | D4 - Intel QPI Link Internal Parity Error |
| 11 | RW1CS | 0 | D3 - Intel QPI Link Layer Control Error |
| 10 | RW1CS | 0 | C0 - Intel QPI Link Layer detected CRC error |
| 9 | RW1CS | 0 | B1 - Intel QPI Link Layer detected CRC error |
| 8 | RW1CS | 0 | B0 - Intel QPI Link Layer CRC - successful link level retry |
| 7 | RW1CS | 0 | DG - Intel QPI Link Layer Detected unsupported/undefined msgclass/opcode/vn packet Error |
| 6 | RW1CS | 0 | B5 - Potential Spurious CRC error on L0s/L1 Exit |
| 5 | RW1CS | 0 | B6 - Intel QPI Link Layer CRC error |
| 4 | RW1CS | 0 | D2 - Intel QPI Physical Layer Initialization Failure |
| 3 | RW1CS | 0 | D1 - Intel QPI Physical Layer Detected Latency Buffer Rollover |
| 2 | RW1CS | 0 | D0 - Intel QPI Physical Layer Detected Drift Buffer Alarm |
| 1 | RW1CS | 0 | C7 - Intel QPI Physical Layer Reset Successful with Reduced Width |
| 0 | RW1CS | 0 | B2 - Intel QPI Physical Layer Successful Reset at same Width |



21.8.1.7 QPI [1:0]NNERRST: Intel QuickPath Interconnect Non-Fatal NERR Status Registers

The error status log indicates which error is causing the report of the next non-fatal error event.

| Register: QPI[1:0]NNERRST Device: 20 Function: 2 Offset: 294h, 214h | | | |
|--|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:13 | RV | 0 | Reserved |
| 12 | RW1CS | 0 | D4 - Intel QPI Link Internal Parity Error |
| 11 | RW1CS | 0 | D3 - Intel QPI Link Layer Control Error |
| 10 | RW1CS | 0 | C0 - Intel QPI Link Layer detected CRC error |
| 9 | RW1CS | 0 | B1 - Intel QPI Link Layer detected CRC error |
| 8 | RW1CS | 0 | B0 - Intel QPI Link Layer CRC - successful link level retry |
| 7 | RW1CS | 0 | DG - Intel QPI Link Layer Detected unsupported/undefined msgclass/opcode/vn packet Error |
| 6 | RW1CS | 0 | B5 - Potential Spurious CRC error on L0s/L1 Exit |
| 5 | RW1CS | 0 | B6 - Intel QPI Link Layer CRC error |
| 4 | RW1CS | 0 | D2 - Intel QPI Physical Layer Initialization Failure |
| 3 | RW1CS | 0 | D1 - Intel QPI Physical Layer Detected Latency Buffer Rollover |
| 2 | RW1CS | 0 | D0 - Intel QPI Physical Layer Detected Drift Buffer Alarm |
| 1 | RW1CS | 0 | C7 - Intel QPI Physical Layer Reset Successful with Reduced Width |
| 0 | RW1CS | 0 | B2 - Intel QPI Physical Layer Successful Reset at same Width |

21.8.1.8 QPI [1:0]JERRCNTSEL: Intel QuickPath Interconnect Error Counter Selection Register

Selects which errors to include in QPIERRCNT.

| Register: QPI[1:0]JERRCNTSEL Device: 20 Function: 2 Offset: 298h, 218h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:13 | RV | 0 | <i>Reserved</i> |
| 12:0 | RW | 0 | See QPIERRST for per bit description of each error 0 - Do not select this error type for error counting 1 - Select this error type for error counting |



21.8.1.9 QPI[1:0]ERRCNT: Intel QuickPath Interconnect Error Counter Register

| Register: QPI[1:0]ERRCNT Device: 20 Function: 2 Offset: 29Ch, 21Ch | | | |
|---|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:8 | RV | 0 | Reserved |
| 7 | RW1CS | | ERROVF: Error Accumulator Overflow 0: No overflow occurred 1: Error overflow. The error count may not be valid. |
| 6:0 | RW1CS | 0 | ERRCNT: Error Accumulator This counter accumulates errors that occur when the associated error type is selected in the ERRCNTSEL register. This register is cleared by writing 7Fh. Maximum counter available is 127d (7Fh). |

21.8.1.10 QPIP[1:0]ERRST: Intel QuickPath Interconnect Protocol Error Status Register

This register indicates the error detected by the Intel QuickPath Interconnect protocol layer. See Section 16.7 for more details on each error type.

| Register: QPIP[1:0]ERRST Device: 20 Function: 2 Offset: 2B0h, 230h | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:20 | RV | 0 | Reserved |
| 19 | RO | 0 | DH - QPI Protocol Layer detected unsupported / undefined packet error. Note: Shown for error code listing completeness only. Hardware and software have no affect on this bit. |
| 18 | RW1CS | 0 | DF - Illegal Inbound Request |
| 17 | RW1CS | 0 | DE - Routing Table Invalid |
| 16 | RV | 0 | Reserved |
| 15 | RW1CS | 0 | DC - Protocol SAD illegal or non-existent memory for outbound snoop |
| 14 | RW1CS | 0 | DB - Protocol Parity Error |
| 13 | RW1CS | 0 | DA - Protocol Queue/Table Overflow or Underflow |
| 12 | RW1CS | 0 | D9 - Protocol Layer Received Viral from Intel® QPI |
| 11 | RW1CS | 0 | D8 - Protocol Layer Received Illegal packet field or Incorrect Target NodeID |
| 10 | RW1CS | 0 | D7 - Protocol Layer Received Unexpected Response/Completion |
| 9 | RW1CS | 0 | D6 - Protocol Layer Received Failed Response |
| 8 | RW1CS | 0 | D5 - Protocol Layer Detected Time-Out in ORB |
| 7-5 | RV | 0 | Reserved |
| 4 | RW1CS | 0 | C3 - CSR access crossing 32-bit boundary |
| 3 | RW1CS | 0 | C2 - Write Cache Un-correctable ECC |
| 2 | RW1CS | 0 | C1 - Protocol Layer Received Poisoned Packet. |



| Register: QPIP[1:0]ERRST Device: 20 Function: 2 Offset: 2B0h, 230h | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 1 | RW1CS | 0 | B4 - Write Cache Correctable ECC |
| 0 | RW1CS | 0 | B3 - Intel® QPI CPEI Error Status |

21.8.1.11 QPIP[1:0]ERRCTL: Intel QuickPath Interconnect Protocol Error Control Register

This register enable the error status bit setting for an Intel QuickPath Interconnect detected error. Setting of the bit enables the setting of the corresponding error status bit in QPIPERRST register. If the bit is cleared, the corresponding error status will not be set.

| Register: QPIP[1:0]ERRCTL Device: 20 Function: 2 Offset: 2B4h, 234h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:20 | RV | 0 | <i>Reserved</i> |
| 19 | RWS | 0 | DH - Intel QPI Protocol Layer Detected unsupported/undefined packet Error Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 18 | RWS | 0 | DF - Illegal Inbound Request Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 17 | RWS | 0 | DE - Routing Table Invalid Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 16 | RV | 0 | <i>Reserved</i> |
| 15 | RWS | 0 | DC - Protocol SAD illegal or non-existent memory for outbound snoop Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 14 | RWS | 0 | DB - Protocol Parity Error Enable |
| 13 | RWS | 0 | DA - Protocol Queue/Table Overflow or Underflow Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 12 | RWS | 0 | D9 - Protocol Layer Received Viral from Intel QPI Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 11 | RWS | 0 | D8 - Protocol Layer Received Illegal packet field or Incorrect Target NodeID Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 10 | RWS | 0 | D7 - Protocol Layer Received Unexpected Response/Completion Enable 0 - Disable error status logging 1 - Enable Error status logging |



| Register: QPIP[1:0]ERRCTL Device: 20 Function: 2 Offset: 2B4h, 234h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 9 | RWS | 0 | D6 - Protocol Layer Received Failed Response Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 8 | RWS | 0 | D5 - Protocol Layer Detected Time-Out in ORB Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 7:5 | RV | 0 | Reserved |
| 4 | RWS | 0 | C3 - CSR access crossing 32-bit boundary Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 3 | RWS | 0 | C2 - Write Cache Un-correctable ECC Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 2 | RWS | 0 | C1 - Protocol Layer Received Poisoned Packet Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 1 | RWS | 0 | B4 - Write Cache Correctable ECC Enable 0 - Disable error status logging 1 - Enable Error status logging |
| 0 | RWS | 0 | B3 - Intel QPI CPEI Error Status Enable 0 - Disable error status logging 1 - Enable Error status logging |

21.8.1.12 Intel QuickPath Interconnect Protocol Error Log Register

This register logs the information associated with the reporting of Intel QuickPath Interconnect protocol layer errors. There are two sets of error log registers of identical format: FERR logs the first occurrence of an error, and NERR logs the next occurrence of the error. An individual error is reported with the corresponding severity in this register. Software can program the error severity to one of the three severities supported by IOH. This register is sticky and can only be reset by PWRGOOD. Clearing of the QPIP**ERRST is done by clearing the corresponding QPIPERRST bits.

21.8.1.13 QPIP[1:0]FFERRST: Intel QuickPath Interconnect Protocol Fatal FERR Status Register

| Register: QPIP[1:0]FFERRST Device: 20 Function: 2 Offset: 2B8h, 238h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:20 | RV | 0 | Reserved |
| 19:17 | ROS | 0 | Intel QPI Error Status Log2 The error status log indicates which error is causing the report of the first fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. |
| 16 | RV | 0 | Reserved |



| Register: QPIP[1:0]FFERRST Device: 20 Function: 2 Offset: 2B8h, 238h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:8 | ROS | 0 | Intel QPI Error Status Log1 The error status log indicates which error is causing the report of the first fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. |
| 7:5 | RV | 0 | Reserved |
| 4:0 | ROS | 0 | Intel QPI Error Status Log0 The error status log indicates which error is causing the report of the first fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. |

21.8.1.14 QPIP[1:0]FNERRST: Intel QuickPath Interconnect Protocol Fatal NERR Status Registers

| Register: QPIP[1:0]FNERRST Device: 20 Function: 2 Offset: 2BCh, 23Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:20 | RV | 0 | Reserved |
| 19:17 | ROS | 0 | Intel QPI Error Status Log2 The error status log indicates which error is causing the report of the next fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. Note: If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register. |
| 16 | RV | 0 | Reserved |
| 15:8 | ROS | 0 | Intel QPI Error Status Log1 The error status log indicates which error is causing the report of the next fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. Note: If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register. |
| 7:5 | RV | 0 | Reserved |
| 4:0 | ROS | 0 | Intel QPI Error Status Log0 The error status log indicates which error is causing the report of the next fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. Note: If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register. |



21.8.1.15 QPIP[1:0]FFERRHD: Intel QuickPath Interconnect Protocol Fatal FERR Header Log Register

| Register: QPIP[1:0]FFERRHD Device: 20 Function: 2 Offset: 2C0h, 240h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 127:0 | ROS | 0 | Intel QPI Error Header log Header log stores the header information of the associated with the first fatal error . The header stores the Intel QuickPath Interconnect packet fields of the erroneous Intel QuickPath Interconnect cycle. Refer to Intel QuickPath Interconnect specification chapter 4 for the header format of each type of Intel QuickPath Interconnect cycle. |

21.8.1.16 QPIP[1:0]NFERRST: Intel QuickPath Interconnect Protocol Non-Fatal FERR Status

| Register: QPIP[1:0]NFERRST Device: 20 Function: 2 Offset: 2D0h, 250h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:20 | RV | 0 | Reserved |
| 19:17 | ROS | 0 | Intel QPI Error Status Log2 The error status log indicates which error is causing the report of the first non-fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. |
| 16 | RV | 0 | Reserved |
| 15:8 | ROS | 0 | Intel QPI Error Status Log1 The error status log indicates which error is causing the report of the first non-fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. |
| 7:5 | RV | 0 | Reserved |
| 4:0 | ROS | 0 | Intel QPI Error Status Log0 The error status log indicates which error is causing the report of the first non-fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. |



21.8.1.17 QPIP[1:0]NNERRST: Intel QuickPath Interconnect Protocol Non-Fatal NERR Status

| Register: QPIP[1:0]NNERRST Device: 20 Function: 2 Offset: 2D4h, 254h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:20 | RV | 0 | Reserved |
| 19:17 | ROS | 0 | Intel QPI Error Status Log2 The error status log indicates which error is causing the report of the next non-fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. <i>Note:</i> If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register. |
| 16 | RV | 0 | Reserved |
| 15:8 | ROS | 0 | Intel QPI Error Status Log1 The error status log indicates which error is causing the report of the next non-fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. <i>Note:</i> If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register. |
| 7:5 | RV | 0 | Reserved |
| 4:0 | ROS | 0 | Intel QPI Error Status Log0 The error status log indicates which error is causing the report of the next non-fatal error event . The encoding indicates the corresponding bit position of the error in the Intel QuickPath Interconnect protocol error status register. <i>Note:</i> If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register. |

21.8.1.18 QPIP[1:0]NFERRHD: Intel QuickPath Interconnect Protocol Non-Fatal FERR Header Log Register

| Register: QPIP[1:0]NFERRHD Device: 20 Function: 2 Offset: 2D8h, 258h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 127:0 | ROS | 0 | Intel QPI Error Header log Header log stores the header information of the associated with the first non-fatal error . The header stores the Intel QuickPath Interconnect packet fields of the erroneous Intel QuickPath Interconnect cycle. Refer to Intel QuickPath Interconnect specification chapter 4 for the header format of each type of Intel QuickPath Interconnect cycle. |



21.8.1.19 QPIP[1:0]ERRCNTSEL: Intel QuickPath Interconnect Protocol Error Counter Selection Register

| Register: QPIP[1:0]ERRCNTSEL Device: 20 Function: 2 Offset: 2E8h, 268h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:20 | RV | 0 | <i>Reserved</i> |
| 19:17 | RWS | 0 | QPIPERRCNTSEL19_17 See QPIPERRST for per bit description of each error 0 - Do not select this error type for error counting 1 - Select this error type for error counting |
| 16 | RV | 0 | <i>Reserved</i> |
| 15:8 | RWS | 0 | QPIPERRCNTSEL15_8 See QPIPERRST for per bit description of each error 0 - Do not select this error type for error counting 1 - Select this error type for error counting |
| 7:5 | RV | 0 | <i>Reserved</i> |
| 4:0 | RWS | 0 | QPIPERRCNTSEL4_0 See QPIPERRST for per bit description of each error 0 - Do not select this error type for error counting 1 - Select this error type for error counting |

21.8.1.20 QPIP[1:0]ERRCNT: Intel QuickPath Interconnect Protocol Error Counter Register

| Register: QPIP[1:0]ERRCNTSEL Device: 20 Function: 2 Offset: 2ECh, 26Ch | | | |
|---|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:8 | RV | 0 | <i>Reserved</i> |
| 7 | RW1CS | | ERROVF: Error Accumulator Overflow 0: No overflow occurred 1: Error overflow. The error count may not be valid. |
| 6:0 | RW1CS | 0 | ERRCNT: Error Accumulator This counter accumulates errors that occur when the associated error type is selected in the ERRCNTSEL register. This register is cleared by writing 7Fh. Maximum counter available is 127d (7Fh). |

21.8.2 IOHERRST: IOH Core Error Status Register

This register indicates the IOH internal core errors detected by the IOH error logic. An individual error status bit that is set indicates that a particular error occurred; software may clear an error status by writing a 1 to the respective bit. This register is sticky and can only be reset by PWRGOOD. Clearing of the IOH**ERRST is done by clearing the corresponding IOHERRST bits.



| Register: IOHERRST Device: 20 Function: 2 Offset: 300h | | | |
|---|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:7 | RV | 0 | Reserved |
| 6 | RW1CS | 0 | C6 - FIFO Overflow/Underflow error |
| 5 | RW1CS | 0 | C5 - Completor abort address error |
| 4 | RW1CS | 0 | C4 - Master abort address error |
| 3 | RV | 0 | Reserved |
| 2 | RV | 0 | Reserved |
| 1 | RV | 0 | Reserved |
| 0 | RV | 0 | Reserved |

21.8.2.1 IOHERRCTL: IOH Core Error Control Register

This register enables the error status bit setting for IOH internal core errors detected by the IOH. Setting of the bit enables the setting of the corresponding error status bit in IOHERRST register. If the bit is cleared, the corresponding error status will not be set.

This register is sticky and can only be reset by PWRGOOD.

| Register: IOHERRCTL Device: 20 Function: 2 Offset: 304h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:7 | RV | | Reserved |
| 6 | RWS | 0 | C6 - FIFO Overflow/Underflow error Enable |
| 5 | RWS | 0 | C5 - Completor abort address error Enable |
| 4 | RWS | 0 | C4 - Master abort address error Enable |
| 3:0 | RV | 0 | Reserved |

21.8.2.2 IOHFFERRST: IOH Core Fatal FERR Status Register

The error status log indicates which error is causing the report of **first fatal error** event.

| Register: IOHFFERRST Device: 20 Function: 2 Offset: 308h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:7 | RV | 0 | Reserved |
| 6:4 | ROS | 0 | IOH Core Error Status Log The error status log indicates which error is causing the report of the first error event. The encoding indicates the corresponding bit position of the error in the error status register. |
| 3:0 | RV | 0 | Reserved |



21.8.2.3 IOHFFERRHD: IOH Core Fatal FERR Header Register

| Register: IOHFFERRHD Device: 20 Function: 2 Offset: 30Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 127:0 | ROS | 0 | IOH Core Error Header log Header log stores the IOH data path header information of the associated IOH core error. The header indicates where the error is originating from and the address of the cycle. [127:90] Reserved [89] Error Type MA/CA [88:81] Message Code [7:0] [80:65] MSI Data [15:0] [64:58] Reserved [57:51] TType {Fmt [1:0], Type[4:0]} [50:0] Address [50:0] Note: For interrupts Address(50:0) will be logged as follows when Interrupt Remapping is enabled: Address(50:19) = DW Address = (Dest ID[29:0], Redirect Hint, Mode) Mode denotes 0 for physical and 1 for logical Address(18:0) - NA for interrupts and could be zeros or ones. The two upper bits of the Destination ID (Dest ID[31:30]) will not be logged when Interrupt Remapping is enabled. |

21.8.2.4 IOHFNERRST: IOH Core Fatal NERR Status Register

The error status log indicates which error is causing the report of the **next error event**.

Note: If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register.

| Register: IOHFNERRST Device: 20 Function: 2 Offset: 31Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:7 | RV | 0 | Reserved |
| 6:4 | ROS | 0 | IOH Core Error Status Log The error status log indicates which error is causing the report of the next error event. The encoding indicates the corresponding bit position of the error in the error status register. Note: If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register. |
| 3:0 | RV | 0 | Reserved |



21.8.2.5 IOHNFERRST: IOH Core Non-Fatal FERR Status Register

The error status log indicates which error is causing the report of the **first error event**.

| Register: IOHNFERRST Device: 20 Function: 2 Offset: 320h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:7 | RV | 0 | Reserved |
| 6:4 | ROS | 0 | IOH Core Error Status Log The error status log indicates which error is causing the report of the first error event. The encoding indicates the corresponding bit position of the error in the error status register. |
| 3:0 | RV | 0 | Reserved |

21.8.2.6 IOHNFERRHD[0:3]: Local Non-Fatal FERR Header Register

| Register: IOHNFERRHD Device: 20 Function: 2 Offset: 324h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 127:0 | ROS | 0 | IOH Core Error Header log Header log stores the IOH data path header information of the associated IOH core error. The header indicates where the error is originating from and the address of the cycle. [127:90] Reserved [89] Error Type MA/CA [88:81] Message Code [7:0] [80:65] MSI Data [15:0] [64:58] Reserved [57:51] TType {Fmt [1:0], Type[4:0]} [50:0] Address [50:0] <i>Note:</i> For interrupts Address(50:0) will be logged as follows when Interrupt Remapping is enabled: Address(50:19) = DW Address = (Dest ID[29:0], Redirect Hint, Mode) Mode denotes 0 for physical and 1 for logical Address(18:0) - NA for interrupts and could be zeros or ones. The two upper bits of the Destination ID (Dest ID[31:30]) will not be logged when Interrupt Remapping is enabled. |



21.8.2.7 IOHNNERRST: IOH Core Non-Fatal NERR Status Register

The error status log indicates which error is causing the report of the **next error event**.

Note: If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register.

| Register: IOHNNERRST Device: 20 Function: 2 Offset: 334h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:7 | RV | 0 | Reserved |
| 6:4 | ROS | 0 | IOH Core Error Status Log The error status log indicates which error is causing the report of the next error event. The encoding indicates the corresponding bit position of the error in the error status register. Note: If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register. |
| 3:0 | RV | 0 | Reserved |

21.8.2.8 IOHERRCNTSEL: IOH Error Counter Selection Register

| Register: IOHERRCNTSEL Device: 20 Function: 2 Offset: 33Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:7 | RV | 0 | Reserved |
| 6 | RW | 0 | C6 - FIFO Overflow/Underflow error Count Select 0 - Do not select this error type for error counting 1 - Select this error type for error counting |
| 5 | RW | 0 | C5 - Completer abort address error Count Select 0 - Do not select this error type for error counting 1 - Select this error type for error counting |
| 4 | RW | 0 | C4 - Master abort address error Count Select 0 - Do not select this error type for error counting 1 - Select this error type for error counting |
| 3:0 | RV | 0 | Reserved |

21.8.2.9 IOHERRCNT: IOH Core Error Counter Register

| Register: IOHERRCNT Device: 20 Function: 2 Offset: 340h | | | |
|--|------|---------|-------------|
| Bit | Attr | Default | Description |
| 31:8 | RV | | Reserved |



| Register: IOHERRCNT Device: 20 Function: 2 Offset: 340h | | | |
|--|-------|---------|---|
| Bit | Attr | Default | Description |
| 7 | RW1CS | | ERROVF: Error Accumulator Overflow 0: No overflow occurred 1: Error overflow. The error count may not be valid. |
| 6:0 | RW1CS | 0 | ERRCNT: Error Accumulator This counter accumulates errors that occur when the associated error type is selected in the ERRCNTSEL register. This register is cleared by writing 7Fh. Maximum counter available is 127d (7Fh). |

21.8.3 THRERRST: Thermal Error Status

This register indicates the thermal errors detected by the IOH error logic. An individual error status bit that is set indicates that a particular error occurred; software may clear an error status by writing a 1 to the respective bit. This register is sticky and can only be reset by PWRGOOD. Clearing of the THR**ERRST is done by clearing the corresponding THRERRST bits.

| Register: THRERRST Device: 20 Function: 2 Offset: 360h | | | |
|---|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3 | | | F3 - Throttling History This error is generated when the most recent throttle event occurs. |
| 2 | RW1CS | 0 | F2 - Catastrophic Thermal Event This error is generated when the temperature at the thermal sensor reaches the TSTHRCATA threshold. |
| 1 | RV | 0 | F1 - TSMAX Updated |
| 0 | RW1CS | 0 | F0 - Thermal Alert This error is generated when the temperature at the thermal sensor exceeds the TSTHRHI threshold. |



21.8.3.1 THRERRCTL: Thermal Error Control

This register controls the reporting of thermal errors detected by the IOH error logic. An individual error control bit that is set allows reporting of that particular error; software may set or clear the respective bit. This register is sticky and can only be reset by PWRGOOD.

| Register: THRERRCTL Device: 20 Function: 2 Offset: 364h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3 | RWS | 0 | F3 - Throttling History 0: Disable setting this status bit on this error 1: Enable setting this status bit on this error |
| 2 | RWS | 0 | F2 - Catastrophic Thermal Event 0: Disable setting this status bit on this error 1: Enable setting this status bit on this error |
| 1 | RWS | 0 | F1 - TSMAX Updated 0: Disable setting this status bit on this error 1: Enable setting this status bit on this error |
| 0 | RWS | 0 | F0 - Thermal Alert 0: Disable setting this status bit on this error 1: Enable setting this status bit on this error |

21.8.3.2 THRNERRST: Thermal Non-Fatal FERR Status

| Register: THRNERRST Device: 20 Function: 2 Offset: 370h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3 | ROS | 0 | Thermal Error Status Log 3 The error status log indicates which error is causing the report of the first error event . The encoding indicates the corresponding bit position of the error in the thermal error status register. |
| 2 | RV | 0 | Reserved |
| 1:0 | ROST | 0 | Thermal Error Status Log 0 and 1 The error status log indicates which error is causing the report of the first error event . The encoding indicates the corresponding bit position of the error in the thermal error status register. |



21.8.3.3 THRNERRST: Thermal Non-Fatal NERR Status

| Register: THRNERRST Device: 20 Function: 2 Offset: 374h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3 | ROS | 0 | Thermal Error Status Log 3 The error status log indicates which error is causing the report of the second error event . The encoding indicates the corresponding bit position of the error in the thermal error status register. <i>Note:</i> If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register. <i>Note:</i> Only second error gets logged into THRFNERRST (subsequent error does not get logged into THRFNERRST. |
| 2 | RV | 0 | Reserved |
| 1:0 | ROST | 0 | Thermal Error Status Log 0 and 1 The error status log indicates which error is causing the report of the second error event . The encoding indicates the corresponding bit position of the error in the thermal error status register. <i>Note:</i> If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register. <i>Note:</i> Only second error gets logged into THRFNERRST (subsequent error does not get logged into THRFNERRST. |

21.8.3.4 THRERRCNTSEL: Thermal Error Counter Selection

| Register: THRERRCNTSEL Device: 20 Function: 2 Offset: 378h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:3 | RV | | Reserved |
| 2 | RW | 0 | F2 - Catastrophic Thermal Event |
| 1 | RW | 0 | F1 - TSMAX Updated |
| 0 | RW | 0 | F0 - Thermal Alert |



21.8.3.5 THRERRCNT: Thermal Error Counter

| Register: THRERRCNT Device: 20 Function: 2 Offset: 37Ch | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:8 | RV | | Reserved |
| 7 | RW1CS | | ERROVF: Error Accumulator Overflow 0: No overflow occurred 1: Error overflow. The error count may not be valid. |
| 6:0 | RW1CS | 0 | ERRCNT: Error Accumulator This counter accumulates errors that occur when the associated error type is selected in the ERRCNTSEL register. This register is cleared by writing 7Fh. Maximum counter available is 127d (7Fh). |

21.8.4 MIERRST: Miscellaneous Error Status

This register indicates the miscellaneous errors detected by the IOH error logic. An individual error status bit that is set indicates that a particular error occurred; software may clear an error status by writing a 1 to the respective bit. This register is sticky and can only be reset by PWRGOOD. Clearing of the MI*ERRST is done by clearing the corresponding MIERRST bits. For details on usage of local error logging.

| Register: MIERRST Device: 20 Function: 2 Offset: 380h | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3 | RW1CS | 0 | 23 - Virtual Pin Port Error Status This bit indicates that VPP Interface has detected an error. This bit is N/A in 0E80, 0F80h registers |
| 2 | RW1CS | 0 | 22 - JTAG TAP Port Status This bit (set to 1) indicates that an error occurred on the JTAG TAP port. This bit is N/A in 0E80, 0F80h registers |
| 1 | RW1CS | 0 | 21 - SM Bus Port Error Status This bit indicates that SMBus Interface has detected an error. This bit is N/A in 0E80, 0F80h registers |
| 0 | RW1CS | 0 | 20 - IOH Configuration Register Parity Error Status This bit indicates that IOH configuration registers have detect a parity error on its critical configuration bits. This bit is N/A in 0E80, 0F80h registers |



21.8.4.1 MIERRCTL: Miscellaneous Error Control

This register controls the reporting of miscellaneous errors detected by the IOH error logic. Setting of the bit enables the setting of the corresponding error status bit in MIERRST register. If the bit is cleared, the corresponding error status will not be set. This register is sticky and can only be reset by PWRGOOD.

| Register: MIERRCTL Device: 20 Function: 2 Offset: 384h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3 | RWS | 0 | 23 - VPP Error Enable |
| 2 | RWS | 0 | 22 - Persistent JTAG Error Enable |
| 1 | RWS | 0 | 21 - Persistent SMBus Retry Status Enable |
| 0 | RWS | 0 | 20 - IOH Configuration Register Parity error Enable |

21.8.4.2 MIFFERRST: Miscellaneous Fatal FERR Status

The error status log indicates which error is causing the report of the **first error event**.

Note: If two non-fatal errors occur in the same cycle, both errors will be logged.

21.8.4.3 MIFFERRHD: Miscellaneous Fatal FERR Header

| Register: MIFFERRHD Device: 20 Function: 2 Offset: 38Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 127:0 | ROS | 0 | Miscellaneous Error Header log Header log stores the IOH data path header information of the associated IOH miscellaneous error. The header indicates where the error is originating from and the address of the cycle. [127:44] Reserved [43:0] Address [43:0] |

21.8.4.4 MIFNERRST: Miscellaneous Fatal NERR Status

The error status log indicates which error is causing the report of the **next error event**.

Note: If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register.

Note: Only second error gets logged into MIFNERRST (subsequent error does not get logged into MIFNERRST).

21.8.4.5 MINFERRST: Miscellaneous Non-Fatal FERR Status

The error status log indicates which error is causing the report of the **first error event**.



Note: If two non-fatal errors occur in the same cycle, both errors will be logged.

| Register: MINFERRST Device: 20 Function: 2 Offset: 3A0h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3 | ROS | 0 | 23 - Virtual Pin Port Error Status |
| 2 | ROS | 0 | 22 - JTAG TAP Port Status |
| 1 | ROS | 0 | 21 - SM Bus Port Error Status |
| 0 | ROS | 0 | 20 - IOH Configuration Register Parity Error Status |

21.8.4.6 MINFERRHD: Miscellaneous Local Non-Fatal FERR Header

| Register: MINFERRHD Device: 20 Function: 2 Offset: 3A4h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 127:0 | ROS | 0 | Miscellaneous Error Header log Header log stores the IOH data path header information of the associated IOH miscellaneous error. The header indicates where the error is originating from and the address of the cycle. [127:46] Reserved [43:0] Address [43:0] |

21.8.4.7 MINNERRST: Miscellaneous Non-Fatal NERR Status

The error status log indicates which error is causing the report of the **next error event**.

Note: If the same error occurs before the FERR status register bit is cleared, it is logged again in the NERR status register.

Note: Only second error gets logged into MIFNERRST (subsequent error does not get logged into MIFNERRST).

| Register: MINNERRST Device: 20 Function: 2 Offset: 3B4h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3 | ROS | 0 | 23 - Virtual Pin Port Error Status |
| 2 | ROS | 0 | 22 - JTAG TAP Port Status |
| 1 | ROS | 0 | 21 - SM Bus Port Error Status |
| 0 | ROS | 0 | 20 - IOH Configuration Register Parity Error Status |



21.8.4.8 MIERRCNTSEL: Miscellaneous Error Counter Selection

| Register: MIERRCNTSEL Device: 20 Function: 2 Offset: 3BCh | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3 | RW | 0 | 23 - VPP Error CNTSEL 0 - Do not select this error type for error counting 1 - Select this error type for error counting |
| 2 | RW | 0 | 22 - Persistent JTAG Error CNTSEL 0 - Do not select this error type for error counting 1 - Select this error type for error counting |
| 1 | RW | 0 | 21 - Persistent SMBus Retry Status CNTSEL 0 - Do not select this error type for error counting 1 - Select this error type for error counting |
| 0 | RW | 0 | 20 - IOH Configuration Register Parity Error Status This bit indicates that IOH configuration registers have detect a parity error on its critical configuration bits. This bit is N/A in 0E80, 0F80h registers |

21.8.4.9 MIERRCNT: Miscellaneous Error Counter

| Register: MIERRCNT Device: 20 Function: 2 Offset: 3C0h | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:8 | RV | 0 | Reserved |
| 7 | RW1CS | 0 | ERROVF: Error Accumulator Overflow 0: No overflow occurred 1: Error overflow. The error count may not be valid. |
| 6:0 | RW1CS | 0 | ERRCNT: Error Accumulator This counter accumulates errors that occur when the associated error type is selected in the ERRCNTSEL register. This register is cleared by writing 7Fh. Maximum counter available is 127d (7Fh). |



21.8.5 QPI[1:0]FERRFLIT0: Intel QuickPath Interconnect FERR FLIT log Register 0

See Section 21.8.1.1 to find out which errors caused the FLIT logging.

| Register: QPI[1:0]FERRFLIT0 Device: 20 Function: 2 Offset: 480h, 400h | | | |
|--|------|---------|-------------|
| Bit | Attr | Default | Description |
| 95:81 | RV | 0 | Reserved |
| 80 | ROS | 0 | ACTIVE_16B |
| 79:0 | ROS | 0 | FLIT |

21.8.5.1 QPI[1:0]FERRFLIT1: Intel QuickPath Interconnect FERR FLIT log Register 1

See Section 21.8.1.1 to find out which errors caused the FLIT logging.

| Register: QPI[1:0]FERRFLIT1 Device: 20 Function: 2 Offset: 48Ch, 40Ch | | | |
|--|------|---------|-------------|
| Bit | Attr | Default | Description |
| 95:81 | RV | 0 | Reserved |
| 80 | ROS | 0 | ACTIVE_16B |
| 79:0 | ROS | 0 | FLIT |

21.8.5.2 QPIP[1:0]FERRLFLIT0: Intel QuickPath Interconnect Protocol FERR Logical FLIT log Register 0

This register is used to log when Intel QPI Protocol Layer Detects unsupported/undefined packet errors.

| Register: QPIP[1:0]FERRLFLIT0 Device: 20 Function: 2 Offset: 4B0h, 430h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 95:72 | RV | 0 | Reserved |
| 71:0 | ROS | 0 | LFLIT: Intel QPI Logical FLIT Format is the format of Intel QuickPath Interconnect. |



21.8.5.3 QPIP[1:0]FERRLFLIT1: Intel QuickPath Interconnect Protocol FERR Logical FLIT log Register 1

This register is used to log when Intel QPI Protocol Layer Detects unsupported/undefined packet errors.

| Register: QPIP[1:0]FERRLFLIT1 Device: 20 Function: 2 Offset: 4BCh, 43Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 95:72 | RV | 0 | Reserved |
| 71:0 | ROS | 0 | LFLIT: Intel QPI Logical FLIT Format is the format of Intel QuickPath Interconnect. |

21.8.5.4 QPIP[1:0]FERRLFLIT2: Intel QuickPath Interconnect Protocol FERR Logical FLIT log Register 2

This register is used to log when Intel QuickPath Interconnect Protocol Layer Detects unsupported/undefined packet errors.

| Register: QPIP[1:0]FERRLFLIT2 Device: 20 Function: 2 Offset: 4C8h, 448h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 95:72 | RV | 0 | Reserved |
| 71:0 | ROS | 0 | LFLIT: Intel QPI Logical FLIT Format is the format of Intel QuickPath Interconnect. |



21.9 On-Die Throttling Register Map and Coarse-Grained Clock Gating

Table 21-18. Device 20, Function 3: On-Die Throttling and Coarse-Grained Clock Gating

| | | | | | | |
|--------|--------|-----|-----------|-----------|-------|-----|
| DID | VID | 00h | | 80h | | |
| PCISTS | PCICMD | 04h | | 84h | | |
| CCR | RID | 08h | | 88h | | |
| HDR | | 0Ch | | 8Ch | | |
| | | 10h | | 90h | | |
| | | 14h | | 94h | | |
| | | 18h | | 98h | | |
| | | 1Ch | | 9Ch | | |
| | | 20h | | A0h | | |
| | | 24h | | A4h | | |
| | | 28h | | A8h | | |
| SID | SVID | 2Ch | | ACh | | |
| | | 30h | | B0h | | |
| | | 34h | | B4h | | |
| | | 38h | | B8h | | |
| | | 3Ch | | BCh | | |
| | | 40h | | C0h | | |
| | | 44h | | C4h | | |
| | | 48h | | C8h | | |
| | | 4Ch | | CCh | | |
| | | 50h | | D0h | | |
| | | 54h | | D4h | | |
| | | 58h | | D8h | | |
| | | 5Ch | | DCh | | |
| | | 60h | TSTHRCATA | E0 | | |
| | | 64h | | E4h | | |
| | | 68h | | TCTRL | E8h | |
| | | 6Ch | TSTHRHI | TSTHRLO | ECh | |
| | | 70h | TSFSC | CTHINT | F0h | |
| | | 74h | CTCTRL | TSTHRRQPI | CTSTS | F4h |
| | | 78h | | TSTIMER | F8h | |
| | | 7Ch | TSTHRNOMC | | FCh | |



21.9.1 On-Die Throttling Registers

21.9.1.1 TSTHRCATA: On-Die Thermal Sensor Catastrophic Threshold Register

| Register: TSTHRCATA Device: 20 Function: 3 Offset: E2h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:9 | RV | 0h | Reserved |
| 8:0 | RWO | DCh | TSTHRCATALM: Thermal Sensor Threshold Catastrophic Limit The field is initialized by software to set the “catastrophic” threshold for the Thermal sensor logic. Resolution of this register is 0.5°C. Default value is 110°C. i.e DCh (220d). |

21.9.1.2 TSVAL: On-Die Thermal Sensor Output Value Register

| Register: TSVAL Device: 20 Function: 3 Offset: E4h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:9 | RV | 0h | Reserved |
| 8:0 | RO | DCh | TSOUTVAL: Thermal Sensor Output Value This field contains the temperature value from the thermal sensor. The output of the thermal sensor is latched in this register based on the settings of the TSCTRL register. This register has a resolution of 0.5°C. Hence, a register value in the range of 0 to 255 (decimal) refers to a temperature range of 0 to 127.5°C. |

21.9.1.3 TSCTRL: On-Die Thermal Sensor Control Register

| Register: TSCTRL Device: 20 Function: 3 Offset: E8h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15 | RW | 0h | Therm alert / trip logic selector. Refer to the Intel® 7500 Chipset BIOS specification for setting information. |
| 14 | RW | 0h | BGTRIPSELECT: BandGap Trip Select. This bit selects either the hot or catastrophic trip output of the Sensor. 0: Hot trip 1: Catastrophic trip |
| 13:10 | RV | 0h | Reserved |
| 9 | RW | 0h | SWTHROTTLE: Software Throttle. 0: Software throttling is disabled. TSDIS gates throttling. 1: Throttling is forced to all interfaces. THERMALERT_N is asserted. |
| 8 | RW | 0h | TSDIS: Thermal Sensor Throttling Disable. 0: The thermal sensor determines closed-loop thermal throttling events when SWTHROTTLE = 0. 1: Thermal sensor throttling is disabled. SWTHROTTLE controls throttling. |



| Register: TCTRL Device: 20 Function: 3 Offset: E8h | | | |
|---|-------|---------|---|
| Bit | Attr | Default | Description |
| 7 | RW1CS | 0 | STSEVHI : Status Event High 0: Thermal sensor event/interrupt is not generated by the sensor logic 1: Thermal sensor event/interrupts is set when the thermal sensor high threshold trip point is crossed. |
| 6 | RW1CS | 0 | STSEVLO : Status Event Low 0: Thermal sensor event/interrupt is not generated by the sensor logic. 1: Thermal sensor event/interrupts is set when the thermal sensor low threshold trip point is crossed. |
| 5:0 | RV | 0 | Reserved |

21.9.1.4 TSTHRLO: On-Die Thermal Sensor Low Threshold Register

| Register: TSTHRLO Device: 20 Function: 3 Offset: ECh | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:9 | RV | 0h | Reserved |
| 8:0 | RW | B4h | TSTHRLOLM : Thermal Sensor Threshold Low Limit The field is initialized by software to set the "low" threshold mark for the thermal sensor logic. (Tsr_lo). Resolution of this register is 0.5°C. Two's-complement binary, range of -128°C to 127.5°C Default value is 90°C. i.e B4h (180d) |

21.9.1.5 TSTHRHI: On-Die Thermal Sensor High Threshold Register

| Register: TSTHRHI Device: 20 Function: 3 Offset: EEh | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:9 | RV | 0h | Reserved |
| 8:0 | RW | 0C8h | TSTHRHILM : Thermal Sensor Threshold High Limit The field is initialized by software to set the "high" threshold for the Thermal sensor logic. (Tsr_hi) Resolution of this register is 0.5°C. Two's-complement binary, range of -128°C to 127.5°C Default value is 100°C. i.e C8h (200d) |



21.9.1.6 CTHINT: On-Die Throttling Hint Register

| Register: CTHINT Device: 20 Function: 3 Offset: F0h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 23:17 | RV | 0h | Reserved |
| 16 | RO | 0h | OVFLO: Hint Overflow 0: Eight or less THERMALERT windows have elapsed since the last time this register was read while CTCTRL.HINTEN was set. 1: More than eight THERMALERT windows have elapsed since the last time this register was read while CTCTRL.HINTEN was set, and THROTTLED hints were lost. |
| 15:8 | RO | 0h | VALID: Maximum Cooling List of valid throttle-histories. Each bit corresponds to its "THROTTLED" bit. Cleared on read (but delivered intact to reading agent). |
| 7:0 | RO | 0h | THROTTLED: Maximum Cooling History of on-die throttling during the last eight consecutive "THERMALERT" windows. If on-die throttling occurred during a THERMALERT window, then its throttle history bit is set |

21.9.1.7 TSFSC: On-Die Thermal Sensor Fan-Speed-Control Register

This register provides the ability to read a relative thermal sensor indication.

| Register: TSFSC Device: 20 Function: 3 Offset: F3h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RO | 0h | TSFSC: Thermal Sensor Fan Speed Control This field contains the difference between the die temperature and the maximum permissible die temperature. This register has a resolution of 0.5°C. Hence, a register value in the range of 0 to 127 (decimal) refers to a temperature difference of 0 to 63.5°C, and a value in the range of 128 to 255 (decimal) refers to a temperature difference of -64°C to 0.5°C. Positive values indicate that the die temperature is lower than the maximum permissible die temperature. A negative difference beyond -64°C "floors" at -64°C. A positive difference beyond 63.5°C "ceilings" at 63.5°C. |

21.9.1.8 CTSTS: On-Die Throttling Status Register

| Register: CTSTS Device: 20 Function: 3 Offset: F4h | | | |
|---|-------|---------|---|
| Bit | Attr | Default | Description |
| 7:2 | RV | 0h | Reserved |
| 1 | RW1CS | 0 | THRMALRT: On-Die Throttling Event |
| 0 | RW1CS | 0 | THRMTRIP: Catastrophic Thermal Event |



21.9.1.9 TSTHRRQPI: Intel QuickPath Interconnect Throttling Threshold Ratio Register

This register provides the ability to vary the amount/ratio of port throttling for the Intel QuickPath Interconnect.

| Register: TSTHRRQPI Device: 20 Function: 3 Offset: F5h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:3 | RV | 0h | Reserved |
| 2:0 | RW | 000 | TTQPIR: Intel QuickPath Interconnect Throttle Ratio This register sets the throttling ratio for the processor buses. This setting is an approximate percentage of peak theoretical request bandwidth for this interface. Value ThrottleLevel PeakBandwidth 000: 00.0% 100.0% Normal Unthrottled setting 001: 50.0% 50.0% 010: Reserved 011: Reserved 100: Reserved 101: Reserved 110: Reserved 111: Reserved |

21.9.1.10 CTCTRL: On-Die Throttling Control Register

| Register: CTCTRL Device: 20 Function: 3 Offset: F7h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:4 | RV | 0h | Reserved |
| 3 | RW | 0 | NOBMC: No BMC Mode '1: TSTHRHI and TSTHRLO drive THERMALERT#. TSTHRNOBMC drives throttling. '0: TSTHRHI and TSTHRLO drive THERMALERT#. TSTHRHI and TSTHRLO also drive throttling, TSTHRNOBMC is un-used. |
| 2 | RV | 0h | Reserved |
| 1 | RW | 0h | HINTEN: On-Die Throttle Hint Enable When this bit is set, on-die throttling hints are enabled. |
| 0 | RW | 0 | NOMAX: Initialize Throttling Mode. Set then clear this bit to initialize the throttling threshold. |



21.9.1.11 TSTIMER: On-Die Thermal Sensor Timer Control

| Register: TSTIMER Device: 20 Function: 3 Offset: F8h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:30 | RV | 0h | Reserved |
| 29:20 | RW | 7dh | FILTER: THERMALERT_N Filter Period Each increment represents one PRESCALER interval. The THERMALERT_N pin updates follow the period specified by this field. (default is 125 for 62.5 ms at 500 us prescaler). |
| 19:0 | RW | 30d40h | PRESCALER: Thermal Sensor Sample Period Each increment represents one core cycle. Thermal sensor updates follow the period specified by this field. (default is 200,000 for 500 us at 400 MHz core). |

21.9.1.12 TSTHRNOBMC: On-Die Thermal Sensor Throttling Threshold Register for NOBMC Mode

| Register: TSTHRNOBMC Device: 20 Function: 3 Offset: FEh | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:9 | RV | 0h | Reserved |
| 8:0 | RW | 0h | TSTHRNOBMCCLM: Thermal Sensor Threshold Throttling Limit The field is initialized by software to set the throttling threshold for the Thermal sensor logic when CTCTRL.NOBMC=1. Resolution of this register is 0.5°C. Two's-complement binary, range of -128°C to 127.5°C |

21.10 Intel QuickPath Interconnect Register Map

Registers assigned to the Link or Physical layers of Intel QuickPath Interconnect need an independent register set per Intel QuickPath Interconnect port. This requires that each register belonging to physical or link be duplicated for each port. QPI[0]RegName is assigned to Intel QuickPath Interconnect port 0 and QPI[1]RegName is assigned to Intel QuickPath Interconnect port 1.

All registers for the routing and protocol layers are defined as a single register, no duplication.

Many control registers that have restrictions on when the register can be modified. If there is a restriction it will be mentioned in the register description, and generally applies to the entire register. The two possibilities for restrictions are: at boot time only, or during quiescence. At boot time only refers to the time immediately following Reset deassertion before any non-configuration requests are flowing within the IOH. During quiescence is a state where only configuration accesses are flowing in the Intel QuickPath Interconnect network, this is generally used for hot-plug type operations.



21.11 Intel QuickPath Interconnect Link Layer Registers

The Link layer registers are defined per Intel QuickPath Interconnect port. There is a special attribute on some link layer registers to handle the Link layer specific reset. The Link layer only hard and soft reset. 'K' attribute indicates that the register is reset on a Link layer hard reset. 'KK' indicates that the register is reset on any Link layer reset (hard or soft).

Table 21-19. Intel QuickPath Interconnect Link Map Port 0 (Dev 16), Port 1 (Dev 17)

| | | | | |
|-----------------|---------------------|-----|----------|-----|
| DID | VID | 00h | | 80h |
| PCISTS | PCICMD | 04h | | 84h |
| CCR | | 08h | | 88h |
| | RID | 0Ch | | 8Ch |
| HDR | CLS | 10h | | 90h |
| | | 14h | | 94h |
| | | 18h | | 98h |
| | | 1Ch | | 9Ch |
| | | 20h | | A0h |
| | | 24h | | A4h |
| | | 28h | | A8h |
| SID | SVID | 2Ch | | ACh |
| | | 30h | | B0h |
| | CAPPTR ¹ | 34h | | B4h |
| | | 38h | | B8h |
| | INTP | 3Ch | | BCh |
| | INTL | 40h | QPILCP | C0h |
| | | 44h | QPILCL | C4h |
| | | 48h | QPILS | C8h |
| | | 4Ch | QPILP0 | CCh |
| | | 50h | QPILP1 | D0h |
| | | 54h | QPILP2 | D4h |
| | | 58h | QPILP3 | D8h |
| QPI[1:0]AGTIDEN | | 5Ch | QPILPOC0 | DCh |
| | | 60h | QPILPOC1 | E0h |
| | | 64h | QPILPOC2 | E4h |
| | | 68h | QPILPOC3 | E8h |
| | | 6Ch | | ECh |
| | | 70h | QPILTC | F0h |
| | | 74h | QPILTS | F4h |
| | | 78h | QPILCRDC | F8h |
| | | 7Ch | | FCh |

Notes:

1. CAPPTR points to the first capability block



21.11.1 Intel QuickPath Interconnect Link Layer Register Tables

21.11.1.1 QPI[1:0]AGTIDEN: Intel QuickPath Interconnect Agent ID Enable Register

| Register: QPI[1:0]AGTIDEN Device: 17, 16 Function: 0 Offset: 5Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:2 | RV | 0 | Reserved |
| 1 | RWS | 0 | TXAGNTIDEN: Enables transmitting Agent ID in the header 0: Normal operation. 1: Enable debug information to be inserted in the address field [42:40] |
| 0 | RWS | 1 | RXAGNTIDEN: Enables receiving Agent ID in the header When this bit is enabled the address bit field [42:40] will be ignored. 0: Normal operation. 1: Enable agent ID debug information to used in place of the address field [42:40]. When enabled the IOH does not decode this bit field as address information. |

21.11.1.2 QPI[1:0]LCP: Intel QuickPath Interconnect Link Capability

Register per Intel QuickPath Interconnect port.

| Register: QPI[1:0]LCP Device: 17, 16 Function: 0 Offset: C0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:30 | RV | 0 | Reserved |
| 29:28 | RO | 0 | VN1 credits per supported Data VC 00 - 0 Credits (unsupported) 01 - 1 credit 10 - 2 to 8 credits 11 - 9+ credits |
| 27:26 | RO | 2h | VN0 credits per supported Data VC 00 - 0 credits (unsupported) 01 - 1 credit 10 - 2 to 8 credits 11 - 9+ credits Max value for VN0 is reflected in this register. Actual value is set by a different register. |
| 25:24 | RO | 0 | VN1 credits per supported non-data VC 00 - 0 Credits (unsupported) 01 - 1 credit 10 - 2 to 8 credits 11 - 9+ credits |
| 23:22 | RO | 2h | VN0 credits per supported non-data VC 00 - 0 credits (unsupported) 01 - 1 credit 10 - 2 to 8 credits 11 - 9+ credits Max value for VN0 is reflected in this register. Actual value is set by a different register. |



| Register: QPI[1:0]LCP Device: 17, 16 Function: 0 Offset: C0h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 21:16 | RO | 16h | VNA Credits Counted by 8s, rounded up. Max value for VNA is reflected in this register. Actual value is set by a different register. |
| 15:12 | RV | 0 | Reserved |
| 11 | RO | 1 | CRC Mode supported 0 - 8b CRC 1 - 8b CRC & 16b Rolling CRC |
| 10 | RO | 0 | Scheduled data Interleave 0 - Not Support 1 - Support |
| 9:8 | RO | 0 | Flit Interleave 00 - Idle flit only (default) 01 - Command insert interleave in data stream 1x -Reserved |
| 7:0 | RO | 0 | Intel QPI Version number 0h - rev 1.0 !0h - reserved |

21.11.1.3 QPI[1:0]LCL: Intel QuickPath Interconnect Link Control

Register per Intel QuickPath Interconnect port. This register is used for control of Link layer.

| Register: QPI[1:0]LCL Device: 17, 16 Function: 0 Offset: C4h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:22 | RV | 0 | Reserved |
| 21 | RWD | 0 | L1 Master/Slave Controls link L1 mode of operation. 0 - Slave 1 - Master |
| 20 | RWD | 0 | L1 enable Bit is ANDed with the parameter exchanged value for L1 to determine if the link may enter L1. 0 - disable 1 - enable Note: This is NOT a bit that determines the capability of a device. |
| 19 | RO | 0 | L0p enable L0p not supported by IOH |
| 18 | RWD | 0 | L0s enable Bit is ANDed with the parameter exchanged value for L0s to determine if the link may enter L0s. 0 - disable 1 - enable Note: This is NOT a bit that determines the capability of a device. |



| Register: QPI[1:0]LCL Device: 17, 16 Function: 0 Offset: C4h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 17 | RWD | 0 | Link Layer Initialization stall at Ready_For_Normal: Note: this bit is set and cleared only by software (no hardware clearing is supported). 0 - disable 1 - enable, stall initialization till this bit is cleared. |
| 16 | RWD | 0 | Link Layer Initialization stall at Ready_For_Init: Note: this bit is set and cleared only by software (no hardware clearing is supported). 0 - disable 1 - enable, stall initialization till this bit is cleared. |
| 15:14 | RWD | 0 | CRC mode (on next initialization) 00 - 8b CRC 01 - 16b rolling CRC, only enabled if peer agent also supports in Parameter0 1X - Reserved |
| 13:12 | RV | 0 | Reserved |
| 11:10 | RWD | 0 | Advertised VNO credits per supported VC (on next initialization) 00 - Max 01 - 2 if <Max 10 - 1 if <Max 11 - 0 Disabled VNO (Can cause deadlock) |
| 9:8 | RWD | 0 | Advertised VNA credits (on next initialization) 00 - Max 01 - 64 if <Max 10 - 32 if <Max 11 - 0 Disable VNA |
| 7:6 | RWD | 0 | Link Layer Retry (LLR) Timeout value in terms of flits recieved 00 – 4095 flits 01 – 1023 flits 10 – 255 flits 11 – 63 flits |
| 5:4 | RWD | 0 | Consecutive LLRs to Link Reset 00 - 16 01 - 8 10 - 4 11 - 0, disable LLR (If CRC error then error condition immediately) |
| 3:2 | RWD | 0 | Consecutive Link Reset from LLR till error condition (only applies if LLR enabled) 00 - up to 2 01 - up to 1 10 - up to 0 11 - Reserved |
| 1 | RW | 0 | Link Hard Reset Re-initialize clearing the values in all link layer registers including Sticky. Write 1 to reset link - this is a destructive reset - when reset asserts, register clears to 0. |
| 0 | RW | 0 | Link Soft Reset Re-initialize clearing the values in all link layer registers except Sticky. Write 1 to reset link - this is a destructive reset - when reset asserts, register clears to 0. |



21.11.1.4 QPI[1:0]LS: Intel QuickPath Interconnect Link Status

Register per Intel QuickPath Interconnect port. This register for holding link status and peer agent info.

| Register: QPI[1:0]LS Device: 17, 16 Function: 0 Offset: C8h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31 | RV | 0 | Reserved |
| 30:28 | RO | 0h | Link Layer Retry Queue Allocation Flits allocated 000 - 0 to 7 001 - 8 to 15 010 - 16 to 31 011 - 32 to 63 100 - 64 to 95 101 - 96 to 127 110 - 128 to 191 111 - 192 to 255 |
| 27:24 | RO | 0h | Link Initialization status 0000 - Waiting for Physical Layer Ready 0001 - Internal Stall Link Initialization 0010 - Sending ReadyForInit 0011 - Parameter Exchange 0100 - Sending ReadyForNormalOperation 0101 - Reserved 0110 - Normal Operation 0111 - Link Level Retry 1000 - Link Error 1001 = Parameter Exchange Done 1010 = WaitForNormal 1011 = LocalLinkReset 11XX, 1001, 101X - Reserved |
| 23:22 | RO | 0 | Link initialization Failure Count - Saturates at 011 All Link Init state machine arcs going into RDY_FOR_INIT excluding the arcs from NOT_RDY_FOR_INIT and from NORMAL_OPERATION. 00 - 0 01 - 1 10 - 2-15 11 - >15 |
| 21 | RO | 0 | Last Link Level Retry NUM_PHY_REINIT- Saturates at 1 Number of Phy ReInits since last Link Init 0: 0 1: 1+ |
| 20-19 | RO | 0 | Last Link Level Retry Count - Saturates at 011 Number of Retries since last Link Init or Phy Reinit 000 - 0 001 - 1 010 - 2-15 011 - >15 |



| Register: QPI[1:0]LS Device: 17, 16 Function: 0 Offset: C8h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 18:16 | RO | 0h | VNA credits at receiver VNA available credits for remote device to use in transmission of packets to IOH. 000 - 0 credits 001 - 1-7 credits 010 - 8-11 credits 011 - 12-15 credits 100 - 16-31 credits 101 - 32-63 credits 110 - 64-127 credits 111 - 128+ credits |
| 15 | RO | 0h | VNO Snp Credits at receiver VNA available credits for remote device to use in transmission of packets to IOH. 0 - 0 Credits 1 - 1+ Credits |
| 14 | RO | 0h | VNO Hom Credits at receiver 0 - 0 Credits 1 - 1+ Credits |
| 13 | RO | 0h | VNO NDR Credits at receiver VNA available credits for remote device to use in transmission of packets to IOH. 0 - 0 Credits 1 - 1+ Credits |
| 12 | RO | 0h | VNO DRS Credits at receiver VNA available credits for remote device to use in transmission of packets to IOH. 0 - 0 Credits 1 - 1+Credits |
| 11 | RO | 0h | VNO NCS Credits at receiver VNA available credits for remote device to use in transmission of packets to IOH. 0 - 0 Credits 1 - 1+ Credits |
| 10 | RO | 0h | VNO N Credits at receiver VNA available credits for remote device to use in transmission of packets to IOH. 0 - 0 Credits 1 - 1+Credits |
| 9:8 | RV | 0h | <i>Reserved</i> |
| 7 | RV | 0 | <i>Reserved.</i> Context shown for other components that support VN1. VN1 Snp Credits at receiver 0: 0 Credits 1: >0 Credits |
| 6 | RV | 0 | <i>Reserved.</i> Context shown for other components that support VN1. VN1 Hom Credits at receiver 0: 0 Credits 1: >0 Credits |
| 5 | RV | 0 | <i>Reserved.</i> Context shown for other components that support VN1. VN1 NDR Credits at receiver 0: 0 Credits 1: >0 Credits |



| Register: QPI[1:0]LS Device: 17, 16 Function: 0 Offset: C8h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 4 | RV | 0 | <i>Reserved.</i> Context shown for other components that support VN1. VN1 DRS Credits at receiver 0: 0 Credits 1: >0 Credits |
| 3 | RV | 0 | <i>Reserved.</i> Context shown for other components that support VN1. VN1 NCS Credits at receiver 0: 0 Credits 1: >0 Credits |
| 2 | RV | 0 | <i>Reserved.</i> Context shown for other components that support VN1. VN1 NCB Credits at receiver 0: 0 Credits 1: >0 Credits |
| 1:0 | RV | 0 | <i>Reserved.</i> |

21.11.1.5 QPI[1:0]LP0: Intel QuickPath Interconnect Link Parameter0

Register per Intel QuickPath Interconnect port. Parameter is exchanged as part of link initialization.

| Register: QPI[1:0]LP0 Device: 17, 16 Function: 0 Offset: CCh | | | |
|---|------|---------|-----------------------------|
| Bit | Attr | Default | Description |
| 31:0 | RONN | 0 | Parameter 0 from peer agent |

21.11.1.6 QPI[1:0]LP1: Intel QuickPath Interconnect Link Parameter1

Register per Intel QuickPath Interconnect port. Parameter is exchanged as part of link initialization.

| Register: QPI[1:0]LP1 Device: 17, 16 Function: 0 Offset: D0h | | | |
|---|------|---------|-----------------------------|
| Bit | Attr | Default | Description |
| 31:0 | RONN | 0 | Parameter 1 from peer agent |



21.11.1.7 QPI [1:0]LP2: Intel QuickPath Interconnect Link Parameter2

Register per Intel QuickPath Interconnect port. Parameter is exchanged as part of link initialization.

| Register: QPI[1:0]LP2 Device: 17, 16 Function: 0 Offset: D4h | | | |
|---|------|---------|-----------------------------|
| Bit | Attr | Default | Description |
| 31:0 | RONN | 0 | Parameter 2 from peer agent |

21.11.1.8 QPI [1:0]LP3: Intel QuickPath Interconnect Link Parameter3

Register per Intel QuickPath Interconnect port. Parameter is exchanged as part of link initialization.

| Register: QPI[1:0]LP3 Device: 17, 16 Function: 0 Offset: D8h | | | |
|---|------|---------|-----------------------------|
| Bit | Attr | Default | Description |
| 31:0 | RONN | 0 | Parameter 3 from peer agent |

21.11.1.9 QPI [1:0]LPOC0: Intel QuickPath Interconnect Link POC0

Register per Intel QuickPath Interconnect port. POC that was recieved as part of link initialization.

| Register: QPI[1:0]LPOC0 Device: 17, 16 Function: 0 Offset: DCh | | | |
|---|------|---------|-----------------------|
| Bit | Attr | Default | Description |
| 31:0 | RONN | 0 | POC 0 from peer agent |

21.11.1.10 QPI [1:0]LPOC1: Intel QuickPath Interconnect Link POC1

Register per Intel QuickPath Interconnect port. POC that was recieved as part of link initialization.

| Register: QPI[1:0]LPOC1 Device: 17, 16 Function: 0 Offset: E0h | | | |
|---|------|---------|-----------------------|
| Bit | Attr | Default | Description |
| 31:0 | RONN | 0 | POC 1 from peer agent |

21.11.1.11 QPI [1:0]LPOC2: Intel QuickPath Interconnect Link POC2

Register per Intel QuickPath Interconnect port. POC that was recieved as part of link initialization.



| Register: QPI[1:0]LPOC2 Device: 17, 16 Function: 0 Offset: E4h | | | |
|---|------|---------|-----------------------|
| Bit | Attr | Default | Description |
| 31:0 | RONN | 0 | POC 2 from peer agent |

21.11.1.12 QPI[1:0]LPOC3: Intel QuickPath Interconnect Link POC3

Register per Intel QuickPath Interconnect port. POC that was recieved as part of link initialization

| Register: QPI[1:0]LPOC3 Device: 17, 16 Function: 0 Offset: E8h | | | |
|---|------|---------|-----------------------|
| Bit | Attr | Default | Description |
| 31:0 | RONN | 0 | POC 3 from peer agent |

21.11.1.13 QPI[1:0]LCL_LATE: Intel QuickPath Interconnect Link Control Late Action

This register is a mirrored copy of the “QPI[1:0]LCL: Intel QuickPath Interconnect Link Control” that have the ‘D’ attribute. The value is captured at Link Layer initialization. These are the late action values that are currently active in the Link Layer.

| Register: QPI[1:0]LCL_LATE Device: 17, 16 Function: 0 Offset: F0h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:22 | RV | 0 | Reserved |
| 21 | RO | 0 | L1 Master/Slave This is a RO copy of the same bits in the “QPI[1:0]LCL: Intel QuickPath Interconnect Link Control” register. See its definition for details. |
| 20 | RO | 0 | L1 enable This is a RO copy of the same bits in the “QPI[1:0]LCL: Intel QuickPath Interconnect Link Control” register. See its definition for details. |
| 19 | RO | 0 | L0p enable This is a RO copy of the same bits in the “QPI[1:0]LCL: Intel QuickPath Interconnect Link Control” register. See its definition for details. Note: L0p is not supported. |
| 18 | RO | 0 | L0s enable This is a RO copy of the same bits in the “QPI[1:0]LCL: Intel QuickPath Interconnect Link Control” register. See its definition for details. |
| 17:16 | RV | 0 | Reserved |
| 15:14 | RO | 0 | CRC mode This is a RO copy of the same bits in the “QPI[1:0]LCL: Intel QuickPath Interconnect Link Control” register. See its definition for details. |
| 13:12 | RV | 0 | Reserved |
| 11:10 | RO | 0 | Advertised VNO credits per supported VC: (on next initialization) This is a RO copy of the same bits in the “QPI[1:0]LCL: Intel QuickPath Interconnect Link Control” register. See its definition for details. |



| Register: QPI[1:0]LCL_LATE Device: 17, 16 Function: 0 Offset: F0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 9:8 | RO | 0 | Advertised VNA credits: (on next initialization) This is a RO copy of the same bits in the "QPI[1:0]LCL: Intel QuickPath Interconnect Link Control" register. See its definition for details. |
| 7:6 | RO | 0 | Link Layer Retry (LLR): Timeout value in terms of flits received This is a RO copy of the same bits in the "QPI[1:0]LCL: Intel QuickPath Interconnect Link Control" register. See its definition for details. |
| 5:4 | RO | 0 | MAX_NUM_RETRY This is a RO copy of the same bits in the "QPI[1:0]LCL: Intel QuickPath Interconnect Link Control" register. See its definition for details. |
| 3:2 | RO | 0 | MAX_NUM_PHY_REINIT This is a RO copy of the same bits in the "QPI[1:0]LCL: Intel QuickPath Interconnect Link Control" register. See its definition for details. |
| 1:0 | RV | 0 | Reserved0 |

21.11.1.14 QPI[1:0]LCRDC: Intel QuickPath Interconnect Link Credit Control

Registers controls what credits are defined for each message class on VNO and VNA. These credits are made visible on the Intel QuickPath Interconnect during the initialize phase of in the link layer. The values programmed here must exist within the size limits defined. Incorrect programming can result in overflow of the receive queue. When returning credits on the Intel QuickPath Interconnect this register is used in conjunction with the Intel QuickPath Interconnect standard register "QPI[1:0]LCL: Intel QuickPath Interconnect Link Control" to determine how many credits are returned. In other words, the values specified in QPI[1:0]LCRDC act as the "Max" in the field descriptions for QPILCL[11:10] and QPILCL[9:8].

This value is captured and used by the Link Layer when exiting the parameter exchange. This state is referred to as "Begin Normal Operation".

| Register: QPI[1:0]LCRDC Device: 17, 16 Function: 0 Offset: F8h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31 | RV | 0 | Reserved |
| 30:28 | RWDS | 1h | VNO - Hom credits Allowed values: 0-7 credits |
| 27 | RV | 0 | Reserved |
| 26:24 | RWDS | 1h | VNO - NCB credits Allowed values: 0-7 credits |
| 23 | RV | 0 | Reserved |
| 22:20 | RWDS | 1h | VNO - NCS credits Allowed values: 0-7 credits |
| 19 | RV | 0 | Reserved |



| Register: QPI[1:0]LCRDC Device: 17, 16 Function: 0 Offset: F8h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 18:16 | RWDS | 1h | VNO - NDR credits Allowed values: 0-7 credits |
| 15 | RV | 0 | Reserved |
| 14:12 | RWDS | 1h | VNO - DRS credits Allowed values: 0-7 credits |
| 11 | RV | 0 | Reserved |
| 10:8 | RWDS | 1h | VNO - Snp credits Allowed values: 0-7 credits |
| 7 | RV | 0 | Reserved |
| 6:0 | RWDS | 64h | VNA credits BIOS must set this to 64h for standard header operation. 0 - 127 credits |

21.11.1.15 QPI[1:0]LCRDC_LATE: Intel QuickPath Interconnect Link Credit Control Late Action

This is a RO copy of the “QPI[1:0]LCRDC_LATE: Intel QuickPath Interconnect Link Credit Control Late Action” register. It is needed to hold the currently active value which is loaded on Link Layer Initialization.

| Register: QPI[1:0]LCRDC_LATE Device: 17, 16 Function: 0 Offset: FCh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31 | RV | 0 | Reserved |
| 30:28 | RO | 1h | VNO Hom credits This is a RO copy of the same bits in the “QPI[1:0]LCRDC: Intel QuickPath Interconnect Link Credit Control” register. See its definition for details. |
| 27 | RV | 0 | Reserved |
| 26:24 | RO | 1h | VNO NCB credits This is a RO copy of the same bits in the “QPI[1:0]LCRDC: Intel QuickPath Interconnect Link Credit Control” register. See its definition for details. |
| 23 | RV | 0 | Reserved |
| 22:20 | RO | 1h | VNO NCS credits This is a RO copy of the same bits in the “QPI[1:0]LCRDC: Intel QuickPath Interconnect Link Credit Control” register. See its definition for details. |
| 19 | RV | 0 | Reserved |
| 18:16 | RO | 1h | VNO NDR credits This is a RO copy of the same bits in the “QPI[1:0]LCRDC: Intel QuickPath Interconnect Link Credit Control” register. See its definition for details. |
| 15 | RV | 0 | Reserved |



| Register: QPI[1:0]LCRDC_LATE Device: 17, 16 Function: 0 Offset: FCh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 14:12 | RO | 1h | VNO DRS credits This is a RO copy of the same bits in the "QPI[1:0]LCRDC: Intel QuickPath Interconnect Link Credit Control" register. See its definition for details. |
| 11 | RV | 0 | Reserved |
| 10:8 | RO | 1h | VNO Snp credits This is a RO copy of the same bits in the "QPI[1:0]LCRDC: Intel QuickPath Interconnect Link Credit Control" register. See its definition for details. |
| 7 | RV | 0 | Reserved |
| 6:0 | RO | 0 | VNA credits This is a RO copy of the same bits in the "QPI[1:0]LCRDC: Intel QuickPath Interconnect Link Credit Control" register. See its definition for details. |

21.11.2 Intel QuickPath Interconnect Routing and Protocol Layer Registers

All Routing layer registers are used to define the routing table functionality. The routing table is used to route packets going out to the Intel QuickPath Interconnect to the correct Intel QuickPath Interconnect port. This is done based on NodeID when the table is enabled. When not enabled completions are routed to the port which their request was received, IB requests will result in an routing layer error.

Table 21-20. CSR Intel QPI Routing Layer, Protocol (Dev 16, Function 1) (Sheet 1 of 2)

| | | | | | | | | | |
|-----------|--------|-----|--|-----|-------------|------|--|-----|------------|
| DID | VID | | | 00h | QPIPAPICSAD | | | | 80h |
| PCISTS | PCICMD | | | 04h | | | | | 84h |
| CCR | | RID | | 08h | | | | | 88h |
| HDR | | CLS | | 0Ch | QPIPDCASAD | | | | 8Ch |
| | | | | 10h | QPIPPVGASAD | | | | 90h |
| | | | | 14h | | | | | 94h |
| | | | | 18h | 98h | | | | |
| | | | | 1Ch | QPIPLIOSAD | | | | 9Ch |
| | | | | 20h | QPIPBUSSAD | | | | A0h |
| | | | | 24h | | | | | A4h |
| | | | | 28h | A8h | | | | |
| | | | | SID | SVID | | | 2Ch | QPIPSUBSAD |
| | | | | 30h | QPIOPORB | | | | B0h |
| | | | | 34h | QPI1PORB | | | | B4h |
| | | | | 38h | | | | | B8h |
| | | | | | INTP | INTL | | 3Ch | QPIPQC |
| QPIRTCTRL | | | | 40h | QPIPLKMC | | | | C0h |
| QPIRTBL | | | | 44h | QPIPNCB | | | | C4h |
| | | | | 48h | | | | | C8h |



Table 21-20. CSR Intel QPI Routing Layer, Protocol (Dev 16, Function 1) (Sheet 2 of 2)

| | | | | |
|-------------|-----|-----|-------------|-----|
| DID | VID | 00h | QPIPAPICSAD | 80h |
| QPIPCTRL | | 4Ch | QPIPLKMS | CCh |
| QPIPSTS | | 50h | QPIQBPCPU | D0h |
| QPIPSB | | 54h | | D4h |
| QPIPRTO | | 58h | QPIQBIOH | D8h |
| QPIPPWCTRL | | 5Ch | | DCh |
| QPIPINT | | 60h | | E0h |
| QPIPRWMAD | | 64h | | E4h |
| QPIPMADDATA | | 68h | | E8h |
| | | 6Ch | | ECh |
| | | 70h | | F0h |
| | | 74h | | F4h |
| | | 78h | | F8h |
| | | 7Ch | | FCh |

Notes:

1. CAPPTR points to the first capability block

21.11.2.1 QPIRTCTRL: Intel QuickPath Interconnect Routing Table Control

This register is the control for the routing table.

| Register: QPIRTCTRL Device: 16 Function: 1 Offset: 40h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:1 | RV | 0 | Reserved |
| 0 | RWLB | 0 | Inbound Routing Method 0 - Route upstream completions to the same port that received the request. New upstream requests are not routed, snoop responses are dropped and a routing error is logged. 1 - Enable Routing Table. Notes: No routing error is logged when QPIRTCTRL[0] is set to 0. Notes: The Inbound Routing method is to be programmed to a '1' in all systems after setting up RT and SAD. Until such time, no coherent traffic is expected in the system |

21.11.2.2 QPIRTBL: Intel QuickPath Interconnect Routing Table

This table is used for fixed routing of Intel QuickPath Interconnect packets.



| Register: QPIRTBL Device: 16 Function: 1 Offset: 44h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 63:0 | RWLB | 0 | Bit per NodeID from 0-31, each bit defines which port that NodeID should target. NodeID mapping: 0 - NodeID 0 ... 30 - NodeID 30 31 - NodeID 31 bit encoding: 0 - Port 0 1 - Port 1 |

21.11.2.3 QPIPCTRL: Intel QuickPath Interconnect Protocol Control

Register can only be modified under system quiescence.

Note: In order for the QPIPCTRL.[44] to work for protecting remote peer to peer accesses to the BAR regions, two additional registers need to be programmed. These are the QPIPQBCPU and QPIPQBIOH. These registers should be programmed to reflect all the node IDs of the CPUs and IOHs in the system in order that the logic correctly distinguish a CPU access from a remote peer to peer access.



| Register: QPIPCTRL Device: 16 Function: 1 Offset: 4Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 63:52 | RV | 0 | Reserved |
| 51:48 | RWL | 0011 | Configuration Retry Timeout: Applies only to PCI Express/ESI ports. Controls how long a configuration request is reissued (when enabled via the root control register) whenever a CRS response is received. Reissue applies to all configuration requests when CRS software visibility is disabled (via the root control register) and to all configuration requests except on configuration reads to Vendor/Device ID field at offset 0x0, when CRS software visibility is enabled. The timer that is controlled by this field starts when a configuration request is issued the very first time on PCI Express. When this timer expires and following that if either a CRS response is received for the configuration request or a completion time-out occurs on the configuration request, the request is aborted (that is, not reissued anymore) and a UR (/equivalent) response is returned. Note that a configuration request is not immediately aborted when this timer expires. Aborting a configuration request only happens when either a completion timeout condition is reached or when a CRS response is received with the retry timeout expired. 0000: 1.28-1.92 ms 0001: 20-30 ms 0010: 120-180 ms 0011: 5-7.5 s 0100: 80-120 s 0101: 160-240 s 0110: 320-480 s 0111: 640-960 s 1000: 1280-1720 s 1001: 5000-7500 s 1010-1111: Reserved |
| 47 | RV | 0 | Reserved |
| 46 | RWL | 0 | Invalid DNID check enable Enables DNID check |
| 45 | RWL | 0 | Write cache flush Flushes the write cache |
| 44 | RWLB | 0 | Enable Peer-to-Peer Protection P2P memory requests have protection requirement that require checking. This mode should only be cleared in platforms that have disabled P2P memory requests. 0 - Disable P2P Protection 1 - Enable P2P Protection Note: In order for this bit to work for protecting remote peer to peer accesses to the BAR regions, two additional registers need to be programmed. These are the QPIPQBCPU and QPIPQBIOH. These registers should be programmed to reflect all the node IDs of the CPUs and IOHs in the system to distinguish a CPU access from a remote peer to peer access. |
| 43:40 | RWL | 0 | Write Cache Isoc Reservation Entries reserved for High Priority (VCp) Isoc traffic. 0 - 7 are legal values. |
| 39:36 | RV | 0 | Reserved |



| Register: QPIPCTRL Device: 16 Function: 1 Offset: 4Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 35 | RW | 1 | Enable p2p Failed Response When this bit is set IOH sends failed response whenever switch aborts a P2P transaction or a Dual IOH master receives failed response. When this bit is clear, IOH sends all 1 response with successful completion when switch aborts transaction. This applies during viral as well. 0 - Disabled |
| 34 | RWL | 0 | No Forwarding When set to 1, disables PQI-to-QPI forwarding sourced by this Intel QuickPath Interconnect. |
| 33 | RW | 0 | Enable Normal Mode Failed Response When this bit is set IOH sends failed response whenever switch aborts a non P2P transaction. When this bit is clear, IOH sends all 1 response with successful completion when switch aborts transaction. This applies during viral as well. 0 - Disabled 1 - Enabled |
| 32 | RW | 0 | Enable Failed Response in Viral When this bit is set IOH sends failed response when the switch aborts a transaction in viral mode. When this bit is not set, IOH sends all 1 response with successful completion. 0 - Disabled 1 - Enabled |
| 31:30 | RWL | 0 | VC1 Priority Setting only applies when "NodeID DP profile decode" mode is enabled. When Isoc is enabled this value should be expected to be set as Critical. 00 - Standard 01 - Reserved 10 - High 11 - Critical |
| 29:28 | RWL | 0 | VCp Priority Setting only applies when "NodeID DP profile decode" mode is enabled. When Isoc is enabled this value should be expected to be set as High. 00 - Standard 01 - Reserved 10 - High 11 - Critical |
| 27:26 | RWL | 0 | VC0 Priority Setting only applies when "NodeID DP profile decode" mode is enabled. Setting VC0 to critical or high may cause deadlocks 00 - Low 01 - Medium 10 - Reserved 11 - Reserved |
| 25 | RWL | 0 | Inbound Reading Snooping Enabling this mode causes IOH to send SnpInvltoE on Inbound Reads. This is used to force peer caching agent to update the home agent if the snoop hits in their cache rather than allow a cache-to-cache transfer directly to the IOH. This mode is required for Dual IOH proxy systems. 0 - Standard Snoop 1 - Force SnpInvltoE |
| 24:22 | RWL | 0 | Default SAD NodeID: [2:0] Used to specify the default home/target NodeID when the SAD is disabled. Only used in UP profile systems, so only 3-bits of NodeID is needed. |



| Register: QPIPCTRL Device: 16 Function: 1 Offset: 4Ch | | | |
|--|------|---------------------------|--|
| Bit | Attr | Default | Description |
| 21:20 | RWL | 0 | NodeID[5:4] This is the NodeID that is assigned to the IOH. It should only be changed via SMBus prior to QPI initialization. |
| 19 | RWL | Strap: QPI/ NodeID3 | NodeID[3] This is the NodeID that is assigned to the IOH. It should only be changed via SMBus prior to Intel QuickPath Interconnect initialization. |
| 18 | RWL | Strap: QPI/ NodeID2 | NodeID[2] This is the NodeID that is assigned to the IOH. It should only be changed via SMBus prior to Intel QuickPath Interconnect initialization. |
| 17:16 | RWL | 0 | NodeID[1:0] This is the NodeID that is assigned to the IOH. It should only be changed via SMBus prior to Intel QuickPath Interconnect initialization. |
| 15 | RV | 0 | Reserved |
| 14:13 | RWL | 0 | Snooping Mode Defines how snooping is done from the IOH. 00 - Broadcast Snoops based on Participant List. Home Node Broadcast will use this mode with the Participant List empty, and the Home agent will take care of snooping. 01 - Broadcast Snoop based on Participant List, but remove the Home NodeID from the list based on match of NodeID[5:0] 10 - Broadcast Snoop based on Participant List, but remove the Home NodeID from the list based on match of NodeID[5:2,0]. Used for a processor that can send snoop from home agent to any caching agent on that socket. 11 - Router Broadcast. Send Snoop to Home NodeID only. |
| 12 | RWL | 0 | Enable Poison by default During testing only, disables poison bit from being sent on Intel QuickPath Interconnect. Any uncorrectable data error will be treated in the same way as a header error. 0 - Enabled 1 - Disabled |
| 11:10 | RWL | 00 | Abort Time-out Mode Control how AbortTO is sent on Intel QuickPath Interconnect. AbortTO response will be sent for outbound CfgRd/Wr when it is pending within the IOH longer then the threshold value. This threshold will deviate by up to +100% of the value specified. 00 - Disable AbortTO 01 - 2 ¹¹ core clocks (5 us @ 400 MHz) 10 - 2 ¹⁷ core clocks (327 us @ 400 MHz) 11 - 2 ²⁴ core clocks (41 ms @ 400 MHz) |
| 9 | RWL | 0 | Disable Viral Disables viral bit from being sent on Intel® QPI or detected from Intel® QPI. 0 - Enabled 1 - Disabled |
| 8 | RWL | 0 | Disable Data forwarding before Completion Default behavior is to forward data immediately to PCI Express when the data phase is received on Intel QuickPath Interconnect. When this bit is set the data will not be forwarded until both data and completion phases have been received. 0 - Enable 1 - Disable |



| Register: QPIPCTRL Device: 16 Function: 1 Offset: 4Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7 | RWL | 0 | Address Mask: [45:41] - DP Profile decode Causes Address[45:41] to be decoded as defined in Intel QuickPath Interconnect DP Profile. This causes those bits to never be sent on Intel QuickPath Interconnect, and to be considered reserved when received on Intel QuickPath Interconnect. 0 - Disable 1 - Enable |
| 6 | RWL | 0 | NodeID DP Profile decode Causes NodeID to be decoded as defined in Intel QuickPath Interconnect DP Profile. This will also insure that PE[1:0] and PH[1:0] (that replace the NodeID bits from SMP and EMP profiles) are always cleared for sending and receiving. 0 - Disable 1 - Enable |
| 5 | RWL | 0 | Extended header Extends addressing beyond 46 bits to 51 bits (only allowed in EMP profile). IOH only supports extending of NodeID to 6-bits with extended headers. When set all headers will use extended format. 0 - Disable 1 - Enable Lock bit is connected to fuse-DISMPPRO |
| 4 | RV | 0 | Reserved |
| 3 | RWL | 0 | Disable write combining Causes all writes to send a EWB request as soon as M-state is acquired. See Section 4.8 for details. 0 - Enable Write Combining 1 - Disable Write Combining |
| 2 | RWL | 0 | RdCur/RdCode mode On Inbound Coherent Reads selection of RdCur or RdCode is done based on this configuration bit. 0 - RdCur 1 - RdCode |
| 1 | RWL | 0 | Inbound Coherent Write mode On Inbound Coherent Writes the request and snoops issued for the RFO phase is selected by this mode. In the "Standard" flow InvItoE/SnpInvItoE is issued. In the "Invalidating Write" flow InvWbMtoI/SnpInvWbMtoI is issued. See Section 4.5.6 for details. 0 - Standard flow 1 - Invalidating Write flow |
| 0 | RWL | 1 | SAD mode Determines how NodeID is decoded. 0 - Single Target specified by Default NodeID. SAD still used to decode memory holes. 1 - Multiple Targets decoded by the SAD |



21.11.2.4 QPIPSTS: Intel QuickPath Interconnect Protocol Status

Note: This register gives status for DRS TX and NDR TX explicitly while gross status for NCB/NCS/SNP TX and HOM TX can be inferred from Bit 2 “ORB Not Empty”. If ORB is empty then the IOH does not have pending NCS/NCB/SNP in TX.

| Register: QPIPSTS Device: 16 Function: 1 Offset: 54h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:5 | RV | 0 | Reserved |
| 4 | RO | 0 | TX DRS Queue NOT empty This indicates that the Protocol TX DRS Queue is not empty. 1 - Pending DRS packets to be transmitted 0 - No DRS packets pending |
| 3 | RO | 0 | TX NDR Queue NOT empty This indicates that the Protocol TX NDR Queue is not empty. 1 - Pending NDR packets to be transmitted 0 - No NDR packets pending |
| 2 | RO | 0 | ORB non-lock_arb Not empty This indicates that there are no pending requests in the ORB with the exception of StopReq*/StartReq* messages from the lock arbiter. 1 - Pending ORB requests 0 - ORB Empty (except StopReq*/StartReq*) |
| 1 | RO | 0 | ORB Empty This indicates that there are no pending requests in the ORB. 0 - Pending ORB requests 1 - ORB Empty |
| 0 | RO | 0 | Write Cache Empty This bit indicates that no E,M state lines exist within the write cache. 0 - Write Cache has current E or M state lines. 1 - Write Cache is empty of E or M state lines. |

21.11.2.5 QPIPSB: Intel QuickPath Interconnect Protocol Snoop Broadcast

Used in Broadcast of snoops for coherent traffic to main memory.

Register can only be modified under system quiescence.

| Register: QPIPSB Device: 16 Function: 1 Offset: 58h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 63:0 | RW | 0 | Snoop vector Each set bit in the vector corresponds to a NodeID. bit - NodeID[5:0] 0 - 000000 1 - 000001 2 - 000010 ... 7- 000111 ... 63 - 111111 |



21.11.2.6 QPIRTO: Intel QuickPath Interconnect Protocol Request Time-Out

The register defines the Intel QPI protocol layer timeout values for each timeout level. The values are proportional to the QPI operational frequency. Register can only be modified under system quiescence.

| Register: QPIRTO Device: 16 Function: 1 Offset: 60h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:24 | RV | 0 | Reserved |
| 23:20 | RW | 0 | Time-out class 5 Same definition as "Time-out class 1" |
| 19:16 | RW | 0 | Time-out class 4 Same definition as "Time-out class 1" |
| 15:12 | RW | 0 | Time-out class 3 Same definition as "Time-out class 1" |
| 11:8 | RV | 0 | <i>Reserved</i> , IOH doesn't support sending requests in Time-out class 2 |
| 7:4 | RW | 0 | Time-out class 1 Controls the timeout value for the request designated to this level. The mode here specifies the timeout counter rate. The actual timeout value will be between 3x and 4x the rate. Mode => Timeout counter rate 0x0 => Timeout disable 0x1 => 2^8 0x2 => 2^{10} 0x3 => 2^{12} 0x4 => 2^{14} 0x5 => 2^{16} 0x6 => 2^{18} 0x7 => 2^{20} 0x8 => 2^{22} 0x9 => 2^{24} 0xA => 2^{26} 0xB => 2^{28} 0xC => 2^{30} 0xD => 2^{32} 0xE => 2^{34} 0xF => 2^{36} Note: The 2^x mathematical term is used below defined as "2 to the power of x". |
| 3:0 | RV | 0 | <i>Reserved</i> |



21.11.2.7 QPIPPOWCTRL: Intel QuickPath Interconnect Protocol Power Control

Register is used to control the PMReq response type. IOH will give only a static response to all PMReq message that can be modified with this register's settings.

Register can only be modified under system quiescence.

| Register: QPIPPOWCTRL Device: 16 Function: 1 Offset: 64h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:20 | RV | 0 | Reserved |
| 19:16 | RW | 0 | State Type Sets the State type in the PMReq response 0000: C 0001: P 0010: S 0011: T others: Reserved |
| 15:0 | RW | 0040h | State Level Sets the State_Level[15:0] in the CmpD response to a PMReq message. The value is priority encoded (similar to one-hot). Default is a state_level of 7. It is required that at least one bit be set in this field. bit: state level 0: 1 1: 2 . 15: 16 |

21.11.2.8 QPIQIPINT: Intel QuickPath Interconnect Protocol Interleave Mask

Controls the system interleave determination used by the source address decoder for memory. This is a system wide parameter for interleave of DRAM. It is used to select from the target list, but exactly how it is used depends on the interleave mode of the SAD entry. Its primary usage model is to interleave between two DRAM home agents within an socket in a MP processor. The function that is expected to be used in the MP processors is parity of PA[19,13,10,6].

| Register: QPIQIPINT Device: 16 Function: 1 Offset: 68h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:16 | RV | 0 | Reserved |
| 15:0 | RW | 0 | System Interleave Bit Mask for PA[21:6] Any bit that is enabled will be included in the even parity calculation on an address being processed by the SAD. This output of this parity calculation may be used in the selection of the Target NodeID. |



21.11.2.9 QIPMADCTRL: Intel QuickPath Interconnect Protocol Memory Address Decoder Control

Controls reads and writes to the Memory Address Decoder. Given the nature of this register software must ensure that only a single producer is modifying this register.

| Register: QIPMADCTRL Device: 16 Function: 1 Offset: 6Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3:0 | RW | 0 | MAD offset Selects the Memory Address Decoder to access on a read or write to "QIPMADDATA: Intel QuickPath Interconnect Protocol Memory Address Decode Data" Valid offsets are 0-15 |

21.11.2.10 QIPMADDATA: Intel QuickPath Interconnect Protocol Memory Address Decode Data

Defines Source Address Decode for memory space. There are 16 decoder entries exist but which one is being accessed depends on the setting in "QIPMADCTRL: Intel QuickPath Interconnect Protocol Memory Address Decoder Control". Both reads and write to this register use the offset defined in that register. This means that software must ensure that only a single producer can be modifying these registers.

| Register: QIPMADDATA Device: 16 Function: 1 Offset: 70h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 127:121 | RV | 0 | Reserved |
| 120:96 | RW | 0 | Limit Address[50:26] Address is 64 MB aligned. |
| 95:89 | RV | 0 | Reserved |
| 88:64 | RW | 0 | Base Address[50:26] Address is 64 MB aligned. |
| 63:16 | RWL | 0 | Target List Bits [xx:nn] - Target NodeID[5:0] Bits [63:58] - Target NodeID7 Bits [57:52] - Target NodeID6 Bits [51:46] - Target NodeID5 Bits [45:40] - Target NodeID4 Bits [39:34] - Target NodeID3 Bits [33:28] - Target NodeID2 Bits [27:22] - Target NodeID1 Bits [21:16] - Target NodeID0 |
| 15:4 | RV | 0 | Reserved |



| Register: QPIPMADDATA Device: 16 Function: 1 Offset: 70h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 3:1 | RWL | 0 | Interleave Select 0x0 - Addr[8:6] 0x1 - Addr[8:7], Sys_Interleave 0x2 - Addr[9:8], Sys_Interleave 0x3 - Addr[8:6] XOR Addr[18:16] 0x4 - (Addr[8:7] XOR Addr[18:17]), Sys_Interleave 0x5 - Addr[8:6]: Similar to 0x0 encoding but the nodeid[1] will be overwritten by the Sys_Interleave bit (as calculated per the CSR setting), i.e. final dnodeid = dnodeid[5:2], Sys_Interleave, dnodeid[0]. 0x6 - Addr[8:6] XOR Addr[18:16]: Similar to 0x3 encoding but the nodeid[1] will be overwritten by the Sys_Interleave bit (as calculated per the CSR setting), i.e. final dnodeid = dnodeid[5:2], Sys_Interleave, dnodeid[0] >0x6 - Reserved |
| 0 | RWL | 0 | Valid 0 - Not Valid 1 - Valid |

21.11.2.11 QPIPAPICSD: Intel QuickPath Interconnect Protocol APIC Source Address Decode

Defines SAD address decode function for inbound interrupts that are not broadcast.

| Register: QPIPSAPICD Device: 16 Function: 1 Offset: 80h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 95:81 | RV | 0 | Reserved |
| 80:72 | RWL | 0 | Physical Mode Local Cluster ID[8:0] Cluster ID is used in hierarchical systems to determine if interrupt go to the local cluster or to the remote. APIC ID bits used to determine cluster are the ones immediately above the interleave mode bits. See Physical Mode Interleave for which bits are matched. |
| 71:68 | RV | 0 | Reserved |
| 67:64 | RWL | 0 | Extended Logical Mode Local Cluster ID[3:0] Cluster ID is used in hierarchical systems to determine if interrupt go to the local cluster or to the remote. APIC ID bits used to determine cluster are the ones immediately above the interleave mode bits. See Extended Logical Mode Interleave for which bits are matched. |
| 63:16 | RWL | 0 | Target List Bits [xx:nn] - Target NodeID[5:0] Bits [63:58] - Target NodeID7 Bits [57:52] - Target NodeID6 Bits [51:46] - Target NodeID5 Bits [45:40] - Target NodeID4 Bits [39:34] - Target NodeID3 Bits [33:28] - Target NodeID2 Bits [27:22] - Target NodeID1 Bits [21:16] - Target NodeID0 |
| 15:14 | RV | 0 | Reserved |



| Register: QPIPSAPICD Device: 16 Function: 1 Offset: 80h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 13:8 | RWLB | 0 | Remote NodeID[5:0] Used to indicate the Node Controller in hierarchical systems. Works in conjunction with the Cluster ID fields. |
| 7 | RV | 0 | Reserved |
| 6:4 | RWLB | 0 | Physical Mode Interleave Mode - Interleave - Local/remote Cluster ID 0x0 - APIC ID[5:3] - APIC ID[14:6] 0x1 - APIC ID[7:5] - 0, APIC ID[15:8] 0x2 - APIC ID[8:6] - 00 & APIC ID[15:9] 0x3 - APIC ID[14:12] - APIC ID[7:0] & APIC ID[15] (Needed for Itanium® processor based platforms) >0x3 - Reserved |
| 3:1 | RWLB | 0 | Extended Logical Mode Interleave Mode - Interleave - Local/remote Cluster ID 0x0 - APIC ID[18:16] - APIC ID[22:19] 0x1 - APIC ID[19:17] - APIC ID[23:20] >0x1 - Reserved |
| 0 | RWLB | 0 | Valid 0 - Not Valid 1 - Valid |

21.11.2.12 QPIPDCASAD: Intel QuickPath Interconnect Protocol DCA Source Address Decode

Sets mode for NodeID generation for the DCA hint. The NodeID is generated based on the PCI Express tag, this register includes the modes for how this NodeID is generated.

| Register: QPIPDCASAD Device: 16 Function: 1 Offset: 8Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:8 | RV | 0 | Reserved |
| 7:6 | RW | 0 | Cache Target Translation Mode Cache target is a bit in the Intel QuickPath Interconnect PrefetchHint message that indicates which target cache in the CPU should get the DCA data. The tag used is from the PCI Express memory write. 00 - "00" 01 - '0' & Tag[0] 10 - Tag[1:0] 11 - Reserved |
| 5:4 | RW | 0 | IDBase[1:0] Base NodeID bits used in some translation modes |



| Register: QPIPDCASAD Device: 16 Function: 1 Offset: 8Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 3:1 | RW | 0 | NodeID Translation Mode The Target NodeID[2:0] for the PrefetchHint on Intel QuickPath Interconnect is generated from based on these modes. 000 - Tag[4:1] & IDBase[1:0] 001 - 0 & Tag[4:2] & IDBase[1:0] 010 - 0 & Tag[4:1] & IDBase[0] 011 - 0 & Tag[4:0] 100- 00 & Tag[4:1] 101 - 000 & Tag[4:2] >101 - Reserved |
| 0 | RWL | 0 | Enable DCA When disabled PrefetchHint will not be sent on Intel QuickPath Interconnect. The inbound write will just follow the standard flow. 0 - Disable 1 - Enable |

21.11.2.13 QPIPVGASAD: QPI Protocol VGA Source Address Decode

Fixed address range (0A0000h-0BFFFFh). Same NodeID used for Legacy I/O requests to the fixed address range (3B0h-3BBh, 3C0h-3DFh).

| Register: QPIPVGASAD Device: 16 Function: 1 Offset: 94h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:14 | RV | 0 | Reserved |
| 13:8 | RW | 0 | NodeID[5:0] NodeID of the VGA controller |
| 7:1 | RV | 0 | Reserved |
| 0 | RWL | 0 | Valid 0 - Not Valid 1 - Valid |



21.11.2.14 QPIPLIOSAD: Intel QuickPath Interconnect Protocol Legacy I/O Source Address Decode

Divided into equal 8 equal 4K or 8K chunks interleaved. This space may contain holes.

| Register: QPIPLIOSAD Device: 16 Function: 1 Offset: 9Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 63:16 | RW | 0 | Target List Bits [xx:nn] - Target NodeID[5:0] Bits [63:58] - Target NodeID7 Bits [57:52] - Target NodeID6 Bits [51:46] - Target NodeID5 Bits [45:40] - Target NodeID4 Bits [39:34] - Target NodeID3 Bits [33:28] - Target NodeID2 Bits [27:22] - Target NodeID1 Bits [21:16] - Target NodeID0 |
| 15:14 | RV | 0 | Reserved |
| 13:8 | RW | 0 | Remote NodeID[5:0] Used if Sub Address match is a miss. Should be used to indicate the NodeID of the Node Controller in a hierarchical system. |
| 7:3 | RV | 0 | Reserved |
| 2 | RW | 0 | Sub Address Bit 15 Sub-Address matched based on Sub-Address enable |
| 1 | RW | 0 | Sub Address Enable When enabled, bit 15 of the legacy I/O space is matched against "Sub Address Bit 15" in this register. If they match then Target list is indexed by Legacy I/O Addr[14:12] to determine NodeID. If no match then Remote NodeID is used. When disabled, Legacy I/O Addr[15:13] indexes the Target List to determine the target NodeID. 0 - disabled 1 - enabled |
| 0 | RWL | 0 | Valid 0 - Not Valid 1 - Valid Locked by lock1 bit. |



21.11.2.15 QPIPBUSSAD: Intel QPI Protocol Bus# Source Address Decode

Used for decode that is based on the PCI Express Bus Number. This space is interleaved across the possible 256 buses.

| Register: QPIPBUSSAD Device: 16 Function: 1 Offset: A4h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 63:16 | RW | 0 | Target List Bits [16:21] - Target NodeID0 Bits [22:27] - Target NodeID1 Bits [28:33] - Target NodeID2 Bits [34:39] - Target NodeID3 Bits [40:45] - Target NodeID4 Bits [46:51] - Target NodeID5 Bits [52:57] - Target NodeID6 Bits [58:63] - Target NodeID7 |
| 15:14 | RV | 0 | Reserved |
| 13:8 | RW | 0 | Remote NodeID[5:0] Used if Sub Bus# match is a miss. Should be used to indicate the NodeID of the Node Controller in a hierarchical system. |
| 7:3 | RV | 0 | Reserved |
| 2 | RW | 0 | Sub Bus# Bits 7 Sub-Address is matched based on Sub-Address Mask |
| 1 | RW | 0 | Sub Bus# enable When enabled, bit 7 of the Bus# is matched against "Sub Bus# Bit 7" in this register. If they match then Target list is indexed by Bus#[6:4] to determine NodeID (16 Bus# per TargetID). If no match then Remote NodeID is used. When disabled, Bus#[7:5] indexes the Target List to determine the target NodeID (32 Bus# per TargetID). 0 - disabled 1 - enabled |
| 0 | RWL | 0 | Valid 0 - Not Valid 1 - Valid |

21.11.2.16 QPIPSUBSAD: Intel QuickPath Interconnect Protocol Subtractive Source Address Decode

Subtractive Decode NodeID. If current IOH is Legacy this should not be used.

| Register: QPIPSUBSAD Device: 16 Function: 1 Offset: ACh | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:14 | RV | 0 | Reserved |
| 13:8 | RW | 0 | Legacy IOH NodeID[5:0] |
| 7:1 | RV | 0 | Reserved |
| 0 | RW | 0 | Valid 0 - Not Valid 1 - Valid |



21.11.2.17 QPI [1:0]PORB: QPI [1:0] Protocol Outgoing Request Buffer

The Request outstanding list has a number of configuration requirements for tag allocation. This register is separated for Port 0, 1. Each port can be programmed independently.

| Register: QPI [1:0]PORB Device: 16 Function: 1 Offset: B4h, B0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:16 | RV | 0 | <i>Reserved</i> |
| 15:12 | RW | 0 | Pool Index This value is set per port because indexing may need to be different on each port because of asymmetric configurations and NodeID assignment. The value controls which 2 NodeID bits are used to select the RTID allocation pools (4 pools per port). 0000 - Single Pool No indexing 0001 - 1,0 0010 - 2,1 0011 - 3,1 0100 - 4,1 0101 - 5,1 0110 - 3,2 0111 - 4,2 1000 - 5,2 1001 - 4,3 1010 - 5,3 1011 - 5,4 1100 - 2,0 1101 - 3,0 1110 - 4,0 1111 - 5,0 |
| 11:9 | RW | 0 | Max Requests Allocation Pool 3 The Max Request value applies per allocation pool. The pool is associated with a single Intel QuickPath Interconnect port. MaxRequest value may be modified by "Merge Pool" bits above. 000 - 16 TID 001 - 24 TID 010 - 32 TID 011 - <i>Reserved</i> 100 - <i>Reserved</i> 101 - <i>Reserved</i> 110 - <i>Reserved</i> 111 - <i>Reserved</i> |
| 8:6 | RW | 0 | Max Requests Allocation Pool 2 Bit definition is the same as Max Request Pool 3 |
| 5:3 | RW | 0 | Max Requests Allocation Pool 1 Bit definition is the same as Max Request Pool 3 |
| 2:0 | RW | 0 | Max Requests Allocation Pool 0 Bit definition is the same as Max Request Pool 3 |



21.11.2.18 QPIPQC: Intel QuickPath Interconnect Protocol Quiescence Control

Used for initiating Quiescence and De-Quiescence of the system. See [Section 4.7](#), “Lock Arbiter” for more information.

Note: The start of the quiesce operation is signaled by setting of bit 0 of this register, whether or not the stop request needs to be sent to the CPU.

| Register: QPIPQC Device: 16 Function: 1 Offset: BCh | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:9 | RV | 0 | Reserved |
| 8 | RWL | 0 | Block PHold When set the Lock Arbiter must block any PHold request received until this bit is cleared. The usage model will be to set this bit prior to quiesce. Then verify the Lock Arbiter is idle. Then follow on to the rest of the quiesce flow. These extra steps are only necessary when PHold is allowed in the platform. 0 - PHold proceeds normally 1 - PHold is blocked Locked by lock1 bit. |
| 7:6 | RV | 0 | Reserved |
| 5:3 | RWLB | 0 | De-Quiescence Software modifying these bits must clear them when corresponding phases are complete. xx1 - StartReq1 (IOH only) x1x - StartReq2 (IOH only) 1xx - StartReq2 (CPU only) |
| 2:0 | RWLB | 0 | Quiescence Software modifying these bits must clear them when corresponding phases are complete. xx1 - StopReq1 (CPU only) x1x - StopReq1 (IOH only) 1xx - Stop Req2 (IOH only) |



21.11.2.19 QPIPLKMC: Intel QuickPath Interconnect Protocol Lock Master Control

Control for Lock Master.

Register modified only under system quiescence.

| Register: QPIPLKMC Device: 16 Function: 1 Offset: C0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:4 | RV | 0 | Reserved |
| 3:1 | RW | 0 | Delay between SysLock In Core Clocks, which are assumed to be at 400 MHz. This may be used to prevent starvation on frequent Lock usage. 000 - 0x0 001 - 0x200 (1.2 us) 010 - 0x1000 (10 us) 011 - 0x2000 (20 us) 100 - 0x4000 (40 us) 101 - 0x8000 (80 us) 110 - 0x10000 (160 us) 111 - 0x20000 (320 us) |
| 0 | RWLB | 0 | Disable Lock Causes NcMsgS-[ProcLock, ProcSplitLock, Quiesce, Unlock] to return immediate Cmp without going through StartReq*/StopReq* sequence. |

21.11.2.20 QPIPNCB: Intel QuickPath Interconnect Protocol Non-coherent Broadcast

List should contain all valid CPU caching agents. This broadcast list is used for some interrupts, Inbound VLW, and Power Management broadcasts on Intel QuickPath Interconnect.

| Register: QPIPNCB Device: 16 Function: 1 Offset: C4h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 63:0 | RW | 0 | Participant List Each set bit in the vector corresponds to a NodeID. bit - NodeID[5:0] 0 - 000000 1 - 000001 2 - 000010 ... 7- 000111 ... 63 - 111111 |



21.11.2.21 QPIPLKMS: Intel QuickPath Interconnect Protocol Lock Master Status

This register contains status of the lock arbiter.

| Register: QPIPLKMS Device: 16 Function: 1 Offset: CCh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:5 | RV | 0 | Reserved |
| 4:0 | RO | 0 | LOCK_ARB_STATE[4:0]: Lock Arbiter Current State Read only value that shows the current state of the lock arbiter. IDLE 5'b00000 (Lock master, the legacy IOH, is idle and ready to launch a lock process) SEND_STOP_REQ1_CPU 5'b00001 (Lock master broadcasts StopReq1 message to all CPUs) SENT_STOP_REQ1_CPU 5'b00010 CMPL_STOP_REQ1_CPU 5'b00011 CMPL_STOP_REQ1_CPU1 5'b00100 (Received completions from all CPUs) SEND_IOH_STOP_REQ15 5'b00101 (Lock master broadcasts StopReq1 message to all IOHs) STOP_REQ1_IOH 5'b00110 (Received completions from all non-legacy IOHs) WAIT_MYIOH_STOP_REQ1 5'b00111 (MYIOH is the lock master, the legacy IOH) STOPPED_REQ1_IOH 5'b01000 (Received completions from all IOHs) SEND_STOP_REQ2 5'b01001 (Lock master broadcasts StopReq2 message to all IOHs and optionally to all CPUs) SEND_STOP_REQ2_2 5'b01010 MYIOH_STOP_REQ2 5'b01011 MYIOH_QUSC_P 5'b01100 MYIOH_QUSC_ALL 5'b01101 MYIOH_PRE_QUSC_QPI 5'b01110 MYIOH_QUSC_QPI 5'b01111 STOPREQ2_SENDCMP 5'b10000 (Received completions from all IOHs and all CPUs if sent) STOPPED_REQ2 5'b10001 (Lock has now been established) START_REQ1 5'b10010 (Lock master launches the unlock process once previous lock cycles are done) SEND_UNLOCK_PCIE 5'b10011 (Unlock the lock target at the endpoint) SEND_STARTREQ1_IOH 5'b10100 (Lock master broadcasts StartReq1 to all IOHs) SENT_STARTREQ1_IOH 5'b10101 (After receiving all IOH completions, go to the state immediately below) SEND_STARTREQ1_CPU 5'b10110 (Lock master broadcasts StartReq1 to all CPUs) SENT_STARTREQ1_CPU 5'b10111 (After receiving all CPU completions, go to the state immediately below) STARTREQ1_DONE 5'b11000 START_REQ2_SEND_IOH 5'b11001 (Lock master broadcasts StartReq2 to all IOHs) START_REQ2_SENT_IOH 5'b11010 START_REQ2_CMPL_IOH 5'b11011 (Received all IOH completions) START_REQ2_SEND_CPU 5'b11100 (Lock master broadcasts StartReq2 to all CPUs) START_REQ2_SENT_CPU 5'b11101 SEND_UNLOCK_CMP_DONE 5'b11110 (Received all CPU completions) COMPLETE 5'b11111. (Unlock has been done, going back to idle above) |



21.11.2.22 QPIPQBCPU: Intel QuickPath Interconnect Protocol Quiesce Broadcast CPU

Controls what processors receive StopReq*/StartReq* messages from the lock arbiter.

| Register: QPIPQBCPU Device: 16 Function: 1 Offset: D0h | | | |
|---|----------|---------|---|
| Bit | Attr | Default | Description |
| 63:0 | RWL B | 0 | Participant List Each set bit in the vector corresponds to a NodeID. bit - NodeID[5:0] 0 - 000000 1 - 000001 2 - 000010 ... 7- 000111 ... 63 - 111111 |

21.11.2.23 QPIPQBIOH: Intel QuickPath Interconnect Protocol Quiesce Broadcast CPU

Controls which IOH's receive StopReq*/StartReq* messages from the lock arbiter. Register can only be modified under system quiescence. In a multi-IOH configuration, the QPIPQBIOH register is also used to determine if an outbound access received by the Intel QPI protocol layer originated in a CPU or a remote IOH. This determination is to support dropping of remote-p2p accesses in the system if necessary. This register must be programmed to indicate node IDs of all IOHs populated in the system.

| Register: QPIPQBIOH Device: 16 Function: 1 Offset: D8h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 63:0 | RWLB | 0 | Snoop vector Each set bit in the vector corresponds to a NodeID. bit - NodeID[5:0] 0 - 000000 1 - 000001 2 - 000010 ... 7- 000111 ... 63 - 111111 Note: In a multi-IOH configuration, the QPIPQBIOH register is also used to determine whether a particular outbound access received by the Intel QuickPath Interconnect protocol layer originated in a CPU or a remote IOH. This register must therefore be programmed to indicate node IDs of all IOHs populated in the system |



21.11.2.24 QPIPSMIC: Intel® QuickPath Interconnect Protocol SMI Control

| Register: QPIPSMIC Device: 17 Function: 1 Offset: A0h | | | |
|--|------|-----------|--|
| Bit | Attr | Default | Description |
| 63:32 | RW | FFFFFFFFh | Destination ID This field specifies the 32-bit Interrupt address. Set to all Fs to indicate broadcast by default. |
| 31:24 | RV | 0 | <i>Reserved</i> |
| 23:16 | RW | 0 | Interrupt Data Vector Software programs this to indicate the system vector number that is assigned to this interrupt. |
| 15:8 | RV | 0 | <i>Reserved</i> |
| 7:5 | RO | 2h | Delivery Mode This field is fixed at 02h to indicate SMI |
| 4 | RO | 0 | Trigger Mode 0 - Edge Triggered only is supported by IOH. |
| 3 | RO | 0 | Address Redirection Hint 0: directed only supported |
| 2 | RO | 0 | Address Destination Mode 0: physical. Only physical mode supported. |
| 1:0 | RV | 0 | <i>Reserved</i> |

21.11.2.25 QPIPNUMIC: Intel® QuickPath Interconnect Protocol NMI Control

| Register: QPIPNUMIC Device: 17 Function: 1 Offset: B0h | | | |
|---|------|-----------|--|
| Bit | Attr | Default | Description |
| 63:32 | RW | FFFFFFFFh | Destination ID This field specifies the 32-bit Interrupt address i.e. physical APICID. Set to all Fs by default, to indicate broadcast. |
| 31:24 | RV | 0 | <i>Reserved</i> |
| 23:16 | RO | 0 | Interrupt Data Vector This field is irrelevant for NMI |
| 15:8 | RV | 0 | <i>Reserved</i> |
| 7:5 | RO | 4h | Delivery Mode This field is fixed at 04h to indicate NMI |
| 4 | RO | 0 | Trigger Mode 0 - Edge Triggered only is supported by IOH. |
| 3 | RO | 0 | Address Redirection Hint 0: directed only supported |
| 2 | RO | 0 | Address Destination Mode 0: physical. Only physical mode supported. |
| 1:0 | RV | 0 | <i>Reserved</i> |



21.11.2.26 QPIPMCAC: Intel QuickPath Interconnect Protocol MCA Control

| Register: QPIPMCAC Device: 17 Function: 1 Offset: C0h | | | |
|--|------|-----------|--|
| Bit | Attr | Default | Description |
| 63:32 | RW | FFFFFFFFh | Destination ID This field specifies the 32-bit Interrupt address. Defaults to broadcast ID. |
| 31:24 | RV | 0 | Reserved |
| 23:16 | RW | FFh | Interrupt Data Vector Software programs this to indicate the system vector number that is assigned to this interrupt. |
| 15:8 | RV | 0 | Reserved |
| 7:5 | RO | 2h | Delivery Mode This field is fixed at 02h to indicate SMI/PMI/MCA. |
| 4 | RO | 0 | Trigger Mode 0 - Edge Triggered only is supported by IOH. |
| 3 | RO | 0 | Address Redirection Hint 0: directed only supported |
| 2 | RO | 0 | Address Destination Mode 0: physical. Only physical mode supported. |
| 1:0 | RV | 0 | Reserved |

21.11.2.27 QPIPINITC: Intel® QuickPath Interconnect Protocol INIT Control

| Register: QPIPINITC Device: 17 Function: 1 Offset: D0h | | | |
|---|------|-----------|--|
| Bit | Attr | Default | Description |
| 63:32 | RW | FFFFFFFFh | Destination ID This field specifies the 32-bit Interrupt address. Defaults to broadcast ID. |
| 31:24 | RV | 0 | Reserved |
| 23:16 | RO | 0 | Interrupt Data Vector N/A for INIT |
| 15:8 | RV | 0 | Reserved |
| 7:5 | RO | 5h | Delivery Mode This field is fixed at 05h to indicate INIT. |
| 4 | RO | 0 | Trigger Mode 0 - Edge Triggered only is supported by IOH. |
| 3 | RO | 0 | Address Redirection Hint 0: directed only supported |
| 2 | RO | 0 | Address Destination Mode 0: physical. Only physical mode supported. |
| 1:0 | RV | 0 | Reserved |



21.11.2.28 QPIPINTRC: Intel QuickPath Interconnect Protocol Interrupt Control

| Register: QPIPINTRC Device: 17 Function: 1 Offset: E0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 63:48 | RV | 0 | Reserved |
| 47:40 | RW | 41h | <p>Legacy Signal Edge/Level: When set, the corresponding legacy wire from ICH is considered as a edge sensitive signal by IOH. When clear, the corresponding legacy wire from ICH is considered as a level sensitive signal by IOH.</p> <p>40: NMI 41: INIT 42: SMI 43: INTR 44: A20M 45-47: Reserved</p> |
| 39:32 | RW | 16h | <p>Legacy Signal Invert: When set, the corresponding legacy wire from ICH is inverted by IOH.</p> <p>32: NMI 33: INIT 34: SMI 35: INTR 36: A20M 37: FERR 38-39: Reserved</p> |
| 31:26 | RV | 0 | Reserved |
| 25 | RW | 0 | <p>Disable PCI INTx Routing to ICH: When this bit is set, <i>local</i> INTx messages received from the CB DMA/PCI Express ports of the IOH are not routed to legacy ICH - they are either converted into MSI via the integrated I/OxAPIC (if the I/OxAPIC mask bit is clear in the appropriate entries) or cause no further action (when mask bit is set). When this bit is clear, <i>local</i> INTx messages received from the CB DMA/PCI Express ports of the IOH are routed to legacy ICH, provided the corresponding mask bit in the IOAPIC is set.</p> |
| 24 | RW | 0 | <p>Route NMI input to MCA: When set, the NMI input into IOH will be routed to MCA message (IntPhysical(MCA)) on Intel QuickPath Interconnect instead of NMI message on Intel QuickPath Interconnect. When clear, the NMI input routes to NMI message on Intel QuickPath Interconnect, either via VLW or IntPhysical(NMI) message, as selected by bits 23:16 of this register.</p> |
| 23:16 | RW | 0 | <p>Intel QPI Message Select for NMI/SMI/INIT: When set, the corresponding pin input (provided the message is also unmasked) from ICH or the IOH internally generated message (on error conditions IOH detects or for other RAS events) will be routed to IntPhysical(*) message on Intel QuickPath Interconnect, otherwise a VLW message is used instead. When IntPhysical message is selected on Intel QuickPath Interconnect, then the address and data for the IntPhysical message is obtained via the registers. Note that if the NMI pin is routed to MCA, then bit 16 only applies to the internally generated NMI from IOH.</p> <p>16: NMI 17: INIT 18: SMI 19-23: Reserved</p> <p>Note that INTR/A20M pins are only routable to VLW message on Intel QuickPath Interconnect whenever they are unmasked with bits 15:8 below.</p> |



| Register: QPIPINTRC Device: 17 Function: 1 Offset: E0h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:8 | RW | 3Fh | <p>Legacy Signal Mask: When set, the corresponding legacy wire from ICH is ignored by IOH and for FERR output, IOH does not assert FERR signal to ICH when masked.</p> <p>8: NMI 9: INIT 10: SMI 11: INTR 12: A20M 13: FERR 15-14: Reserved</p> |
| 7 | RW | 0 | <p>IA-32 or Itanium Processor Family</p> <p>This bit indicates if IOH is in an Intel Xeon processor 7500 series based system or Intel Itanium processor 9300 series based system.</p> <p>This is needed by a) the IOH interrupt redirection logic to know how to interpret an interrupt with APICID set to 0xFF, that is, to broadcast that interrupt or direct to a single processor b) treat logical mode interrupts as illegal in the Itanium Processor Family (IPF) mode</p> <p>0=IA32 1=IPF</p> |
| 6 | RW | 0 | <p>Physical Mode Interrupt & Extended Logical Interrupt Route/Broadcast</p> <p>Should be set to Route mode when firmware is able to set up a map of APIC ID to NodeID in the Intel QuickPath Interconnect SAD (Source Address Decoder).</p> <p>0 - Route physical/extended-logical mode interrupts to a single target. If redirection is performed by IOH, the RH (Redirection Hint) bit will be cleared. If redirection is not performed by IOH, the RH bit is preserved.</p> <p>1 - Broadcast physical/extended-logical mode interrupts using interrupt broadcast list. In this setting, RH bit is cleared for all physical mode interrupts (legacy or extended). But the RH is still preserved (from the original interrupt or from the interrupt table) for extended logical mode interrupts.</p> <p>Note: This field is expected to be programmed to 0 (default) by BIOS for most platforms. For custom configurations where broadcast of physical and extended-logical-mode interrupts are required, the BIOS should also set bit [5] of this register to force round-robin redirection.</p> |
| 5 | RW | 0 | <p>Select Round robin redirection for logical mode</p> <p>When set, this bit selects a simple round-robin redirection for logical flat and non-broadcast cluster mode interrupts in IA32. When clear, vector number based redirection is selected. Note that cluster mode redirected broadcast interrupts are illegal.</p> |
| 4:3 | RW | 0 | <p>Vector based interrupt redirection control</p> <p>00 select bits 6:4/5:4 for vector cluster/flat algorithm 01 select bits 5:3/4:3 10 select bits 3:1/2:1 11 select bits 2:0/1:0</p> |
| 2 | RW | 0 | <p>Disable extended cluster mode interrupt redirection:</p> <p>1: IOH does not perform any redirection of extended cluster mode interrupts. These interrupts are simply forwarded (either routed or broadcast based on bit 6 in this register) as-is to the cpu for redirection in the uncore</p> <p>0: IOH performs redirection of extended cluster mode interrupts as explained in the interrupt chapter. These interrupts are then forwarded (either routed or broadcast based on bit 6 in this register) to the cpu with only the selected cpu thread indicated in the interrupt mask field of the interrupt packet on Intel QuickPath Interconnect.</p> |



| Register: QPIPINTRC Device: 17 Function: 1 Offset: E0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 1 | RW | 0 | IA-32 Logical Flat or Cluster Mode Set by bios to indicate if the OS is running logical flat or logical cluster mode. This bit can also be updated by IntPrioUpd messages. 0=flat, 1=cluster. |
| 0 | RW | 0 | Cluster Check Sampling Mode 0: Disable checking for Logical_APICID[31:0] being non-zero when sampling flat/cluster mode bit in the IntPrioUpd message as part of setting bit 1 in this register 1: Enable the above checking See Interrupt chapter for more details |

21.11.2.29 QPIPINTRS: Intel QuickPath Interconnect Protocol Interrupt Status

This register is to be polled by bios to determine if internal pending system interrupts are drained out of IOH. General usage model is for software to quiesce the source (for example, IOH global error logic) of a system event like SMI, then poll this register till this register indicates that the event is not pending inside IOH. One additional read is required from software, after the register first reads 0 for the associated event.

| Register: QPIPINTRS Device: 17 Function: 1 Offset: E8h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:8 | RV | 0 | Reserved |
| 7 | RO | 0 | MCA RAS event pending |
| 6 | RO | 0 | NMI RAS event pending |
| 5 | RO | 0 | SMI RAS event pending |
| 4 | RO | 0 | INTR# event (either VLW or IntPhysical) pending |
| 3 | RO | 0 | A20M# pin event pending |
| 2 | RO | 0 | INIT# pin event (either VLW or IntPhysical) pending |
| 1 | RO | 0 | NMI pin event (either VLW or IntPhysical) pending |
| 0 | RO | 0 | SMI# pin event (either VLW or IntPhysical) pending |



21.11.3 Intel QuickPath Interconnect Physical Layer Registers

The Physical layer has an internal reset, hard and soft, that result in special register requirements unique to the physical layer. There are two attributes, 'P' and 'PP', which indicate the register is affected by a physical layer reset. 'P' indicates the register is reset on a hard physical layer reset. 'PP' indicates the register is reset on any physical layer reset (hard or soft).

The following state encoding are used in Table 21-11 are used in a number of register encoding below.

Table 21-21. QPIPH-Intel QuickPath Interconnect Tracking State Table

| Bits | State Name |
|--------|--------------------------------------|
| 0 0000 | Reset.Soft & Reset.Default |
| 0 0001 | Reset.Calibrate |
| 0 0010 | Detect.ClkTerm |
| 0 0011 | Detect.FwdClk |
| 0 0100 | Detect.DCPattern |
| 0 0101 | Polling.BitLock |
| 0 0110 | Polling.LaneDeskew |
| 0 0111 | Polling.Param |
| 0 1000 | Config.LinkWidth |
| 0 1001 | Config.FlitLock |
| 0 1010 | Reserved |
| 0 1100 | Reserved |
| 0 1101 | Reserved |
| 0 1110 | LOR (Periodic Retraining in process) |
| 0 1111 | LO |
| 1 0010 | Loopback.Marker Master |
| 1 0011 | Loopback.Marker Slave |
| 1 0000 | Loopback.Pattern Master |
| 1 0001 | Loopback.Pattern Slave |
| 1 1111 | Compliance |
| Others | Reserved. |



21.11.3.1 QPI[1:0]PH_CPR: Intel QuickPath Interconnect Physical Layer Capability Register

| Register: QPI[1:0]PH_CPR Device: 13 Function: 1-0 Offset: 828h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:29 | RV | 0 | <i>Reserved</i> |
| 28:24 | RO | 10h | NumTxLanes: Number of Lanes Intel QuickPath Interconnect lanes supported in full width mode. 1XXXX: 20 lanes X1XXX: 19 lanes XX1XX: 18 lanes XXX1X: 17 lanes XXXX1: 16 lanes |
| 23 | RO | 1 | BitlockRetrainPatt: Bit-lock and Retrain with pattern If Set, the implementation supports using a specified patter in bit-lock/retraining. |
| 22 | RO | 0 | DatScrambleLFSR: Data Scramble with LFSR Intel QPI Link Behavior If set, implementation capable of data scrambling/descrambling with LFSR |
| 21:20 | RO | 0 | RASCap: RAS capability 00: Intel QuickPath Interconnect clock failover not supported |
| 19:18 | RV | 0 | <i>Reserved</i> |
| 17:16 | RO | 1 | DeterminismSupport: 1X: Master mode capability X1: Slave mode capability |
| 15 | RV | 0 | <i>Reserved</i> |
| 14:12 | RO | 1h | PwrMgmtCap: Power management Capability Note: Intel QPI Specific Field Bit 12: L0s entry capable. Bit 13: LWM capable. Bit 14: L1 entry capable. |
| 11 | RV | 0 | <i>Reserved</i> |
| 10:8 | RO | 7h | LinkWidthCap: Link Width Capability XX1: Full Width X1X: Half Width 1XX: Quarter Width |
| 7:5 | RO | 0 | DebugCap: Debug Capability 1XX: Not capable of extracting slave electrical parameter from TS.Loopback and apply during test X1X: Not capable of running in compliance slave mode as well as transitioning to Loopback.pattern from Compliance state XX1: Not capable of doing Loopback.Stall |
| 4 | RO | 0 | RetrainDurationGranulariity: Retraining duration granularity 0: No support for retraining on a 16UI granularity 1: Support for retraining on a 16UI granularity |
| 3:0 | RO | 0 | PhyVersion: Intel QPI physical layer version 0: Rev0 all other encoding are reserved. |



21.11.3.2 QPI [1:0]PH_CTR: Intel QuickPath Interconnect Physical Layer Control Register

| Register: QPI [1:0]PH_CTR Device: 13 Function: 1-0 Offset: 82Ch | | | |
|--|-------|------------------------|--|
| Bit | Attr | Default | Description |
| 31:24 | RV | 0 | Reserved |
| 23 | RWS | 0 | EnableBitlockRetrainwithPatt: Enable Bit-Lock and retraining with pattern 0: Use clock pattern for bitlock/retraining 1: Use pattern in bitlock/retraining |
| 22 | RWS | 0 | EnableScrambleLFSR: Enable Scrambling with LFSR Intel QPI Behavior 0: Data not scrambled/descrambled 1: Data scrambled/descrambled with LFSR |
| 21:14 | RV | 0 | Reserved |
| 13 | RWS | 0 | DisableAutocompliance: Disable Auto-Compliance 0: Path from Detect.Clkterm to compliance is allowed. 1: Path from Detect.Clkterm to compliance is disabled. |
| 12:8 | RV | 0 | Reserved |
| 7 | RWDPP | 0 | LinkSpeed: Full Speed Initialization 0: Slow speed initialization. 1: Force direct operational speed initialization. |
| 6 | RV | 0 | Reserved |
| 5 | RWS | Strap: NOT(BMCInit) | PhyInitBegin - Strap dependence: If BMCinit = 0 then default = 1 If BMCinit = 1 then default = 0 |
| 4 | RWDS | 0 | Single Step Mode - Intel QPI Link Behavior 0: Link behaves as defined by initialization mode. 1: Physical layer is in single step mode. |
| 3:2 | RV | 0 | Reserved |
| 1 | RWS | 0 | ResetModifier: Modifies the behavior of <i>Physical Layer Reset</i> . <i>Reset Modifier</i> = 0 - Upon setting <i>Physical Layer Reset</i> to 1 the state machine will transition to Reset.Soft. <i>Reset Modifier</i> = 1 - Upon setting <i>Physical Layer Reset</i> to 1 the state machine transition to Reset.Default. |
| 0 | RW1S | 0 | PhyLayerReset: Physical Layer Reset (re-initialization) Used to Reset the Physical Layer, and is <i>Link Type</i> dependent in it usage and definition. Setting <i>Physical Layer Reset</i> to 1 initiates an inband Reset by transitioning to either Reset.Soft or Reset.Default depending on the value of <i>Reset Modifier</i> . If Reset Modifier = 0 the setting <i>Physical Layer Rest</i> to 1 will cause a transition to Reset.Soft. If Reset Modifier = 1 the setting <i>Physical Layer Rest</i> to 1 will cause a transition to Reset.Default. <i>Physical Layer Reset</i> will be cleared to 0 in Reset.Calibrate state. |



21.11.3.3 QPI[1:0]PH_TDS: Intel QPI Tx Data Lane Ready Status Register

| Register: QPI[1:0]PH_TDS Device: 13 Function: 1-0 Offset: 834h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31 | ROPP | 1 | TxFwdClkDetectSta: Tx forwarded clock lane detection status Intel QPI Behavior A bit value of 0 indicates forwarded clock lane is dropped |
| 30:20 | RV | 0 | <i>Reserved</i> |
| 19:0 | ROPP | FFFFh | TxLaneDetectStat: Tx Lane Status Mask A bit value of 0 indicates the TX lane is dropped. Bit 0: Status of lane 0. Bit 1: Status of lane 1. .. and so on. 1:1 correspondence to data lanes on Intel QPI. 1 - Indicates corresponding Tx lane is successfully initialized 0 - Indicates corresponding Tx lane is not initialized. Dual Data & Clock lanes are disabled only in Data mode of operation. |

21.11.3.4 QPI[1:0]PH_RDS: Intel QPI Rx Data Lane Ready Status Register

| Register: QPI[1:0]PH_RDS Device: 13 Function: 1-0 Offset: 838h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:20 | RV | 0 | <i>Reserved</i> |
| 19:0 | ROPP | FFFFh | RxDataReadyStat: Rx Status Lane Mask QPI Behavior A bit value of 0 indicates Rx lane is dropped. Bit 0: Status of Lane 0 Bit 1: Status of Lane 1. .. and so on. |



21.11.3.5 QPI [1:0]PH_PIS: Intel QuickPath Interconnect Physical Layer Initialization Status

| Register: QPI [1:0]PH_PIS Device: 13 Function: 1-0 Offset: 840h | | | |
|--|--------|---------|---|
| Bit | Attr | Default | Description |
| 31:28 | RV | 0 | <i>Reserved</i> |
| 27 | RW1C P | 0 | StateMachineHold: State Machine Hold Intel QPI Link Behavior <i>State Machine Hold</i> is only used when <i>Single Step Mode</i> is set to 1. When <i>State Machine Hold</i> is set to 1 this indicates that Physical layer state machine is holding at the end of a particular initialization state (indicated by <i>RX State Tracker</i>). <i>Initialization Mode</i> is also important to poll when <i>State Machine Tracker</i> is set because if Initialization Mode may indicate and Initialization Failure has occurred. Clearing the <i>State Machine Hold</i> bit once it is set to 1 will cause state machine to advance to whatever next state would normally occur. |
| 26 | RO | 0 | InitializeSpeed: Init Speed Note: Intel QPI Specific Field 0: Slow Speed Initialization. 1: Operational Speed Initialization. |
| 25:21 | RV | 0 | <i>Reserved</i> |
| 20:16 | RO | 0 | RxStateTracker: Rx State Tracker Intel QuickPath Interconnect Behavior Indicates the current state of local Rx. State tracker encoding is given in Table 21-22 . |
| 15:13 | RV | 0 | <i>Reserved</i> |
| 12:8 | RO | 0 | TxStateTracker: Tx State Tracker Intel QuickPath Interconnect Behavior Indicates the current state of local Tx. State tracker encoding is given in Table 21-22 |
| 7:1 | RV | 0 | <i>Reserved</i> |
| 0 | RW1CP | 0 | LinkupIdentifier: Linkup Identifier Intel QuickPath Interconnect Behavior Set to 0 during Reset.Default Set to 1 when initialization completes and link enters L0. Note: The attribute is ROS when retraining is enabled. |

Table 21-22. QPIPH-Intel QuickPath Interconnect Tracking State Table (Sheet 1 of 2)

| Bits | State Name |
|--------|----------------------------|
| 0 0000 | Reset.Soft & Reset.Default |
| 0 0001 | Reset.Calibrate |
| 0 0010 | Detect.ClkTerm |
| 0 0011 | Detect.FwdClk |
| 0 0100 | Detect.DCPattern |
| 0 0101 | Polling.BitLock |
| 0 0110 | Polling.LaneDeskew |
| 0 0111 | Polling.Param |



Table 21-22. QPIPH-Intel QuickPath Interconnect Tracking State Table (Sheet 2 of 2)

| Bits | State Name |
|--------|--------------------------------------|
| 0 1000 | Config.LinkWidth |
| 0 1001 | Config.FlitterLock |
| 0 1010 | Reserved |
| 0 1100 | Reserved |
| 0 1101 | Reserved |
| 0 1110 | L0R (Periodic Retraining in process) |
| 0 1111 | L0 |
| 1 0010 | Loopback.Marker Master |
| 1 0011 | Loopback.Marker Slave |
| 1 0000 | Loopback.Pattern Master |
| 1 0001 | Loopback.Pattern Slave |
| 1 1111 | Compliance |
| Others | Reserved. |

21.11.3.6 QPI [1:0]PH_PTV: Intel QuickPath Interconnect Physical Primary Time-Out Value

| Register: QPI [1:0]PH_PTV Device: 13 Function: 1-0 Offset: 854h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:20 | RV | 0 | Reserved |
| 19:16 | RWDS | 1h | ETPollingBitlock: Exponential Time Polling Bit Lock Intel QPI Behavior Exponential count for $T_{POLLING.BitLock}$ Time-out value is $2^{(value)} * 128$ TSL (Training Sequence Length) |
| 15:12 | RV | 0 | Reserved |
| 11:8 | RWDS | 1h | ETInbandRstInit: Exponential Time Inband Reset until Initialization Time-out value is $2^{(value)} * 128$ TSL (Training Sequence Length) |
| 7:4 | RV | 0 | Reserved |
| 3:0 | RV | 2h | Reserved |



21.11.3.7 QPI [1:0]PH_PRT: Intel QuickPath Interconnect Physical Periodic Retraining

| Register: QPI[1:0]PH_PRT Device: 13 Function: 1-0 Offset: 864h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:23 | RV | 0 | Reserved |
| 22 | RWDP | 0 | DurationGranularity: Duration Granularity 0: Indicates agent is using 64 UI granularity 1: Indicates agent is using 16 UI granularity |
| 21:14 | RWDP | 0 | RetrainPacketCnt: Retraining Packet Count retraining packet count; used for retraining duration calculation |
| 13:10 | RWDP | 0 | ExpCntRetrainInterval: Exponential Retraining Interval Exponential count for Retraining Interval. Interval value is multiplied by 2^(count in this field). Although these values are specified in exponential form, counting still needs to be accurate to single UI. |
| 9:8 | RV | 0 | Reserved |
| 7:0 | RWDP | 0 | RetrainInterval: Periodic Retraining Interval Periodic Retraining Interval. A value of 0 indicates periodic retraining is disabled. Value to be programmed by firmware. Each count represents 1024 UI (16 TSL) |

21.11.3.8 QPI [1:0]EP_SR: Electrical Parameter Select Register

| Register: QPI[1:0]EP_SR Device: 13 Function: 1-0 Offset: 8A0h | | | |
|--|------|---------|---|
| Bits | Attr | Default | Description |
| 31:24 | RV | 0 | Reserved |
| 23:16 | RWS | 0 | EParamSel: Electrical parameter select [7:0] This selects the particular Electrical Parameter to be interfaced. 5h: TEQ (TX Equalization) |
| 15:0 | RV | 0 | Reserved |



21.11.3.9 QPI[1:0]MCTR: Electrical Parameter Miscellaneous Control Register

This register is defined for use by Electrical Parameter ID requiring additional control bits for operation.

| Register: QPI[1:0]EP_MCTR Device: 13 Function: 1-0 Offset: 8B4h | | | |
|--|------|---------|--|
| Bits | Attr | Default | Description |
| 31:21 | RV | 0 | MiscEPCtrl: Miscellaneous electrical parameter. The electrical parameter defined in EParamSel is written here |
| 20 | RWS | 0 | TAPDIS: No tap select 0: Tx EQ is enabled 1: TXEQ is disabled |
| 19:16 | RWS | 2h | EQCPRE1: Equalization pre-cursor 1 upper[3:0] |
| 15:13 | RV | 0 | Reserved |
| 12 | RWS | 0 | SGNPOST2: Sign bit for 2nd post upper |
| 11:8 | RWS | 0 | EQPOST2: Equalization Coefficient 2nd post cursor upper[3:0] |
| 7:5 | RV | 0 | Reserved |
| 4:0 | RWS | 10h | EQPOST1: Equalization Coefficient 1st post cursor upper[4:0] |

21.11.3.10 MAX_FCO:

| Register: MAX_FCO Device: 13 Function: 7 Offset: 3B8h | | | |
|--|------|---------|---|
| Bits | Attr | Default | Description |
| 31:28 | RV | 0 | Reserved |
| 27 | RWS | 0 | 1: IOH is lock/quiesce master 0: non-legacy IOH or IOH not lock/quiesce master |
| 26:0 | RV | 0 | Reserved |



21.12 PCI Express, ESI Configuration Space Registers

This section covers the configuration space registers for PCI Express and ESI. See [Section 21.10](#) for Intel QuickPath Interconnect configuration registers.

The next PCIe sections will cover register definitions for devices 0-10 and this description will be divided into three parts. One part that describes the standard PCI header space from 0x0 to 0x3F. The second part describes the device specific region from 0x40 to 0xFF. The third part describes the PCI Express enhanced configuration region.

Notes on register descriptions below:

- Note that in the following sections, PCI Express has been generically used to indicate either a standard PCI Express port or an ESI port and any exceptions to this are called out where applicable.
- When N/A is used in any of the "Device" number rows that indicates the register does not apply to the indicated devices and the register descriptor in the remainder of the table hence will not apply to those devices. There could be other registers defined at the same offset for these device numbers or the offset could be reserved.

21.12.1 Other Register Notes

Note that in general, all register bits in the standard PCI header space (offset 0-3F) or in any OS-visible capability registers, which control the address decode like MSE, IOSE, VGAEN or otherwise control transaction forwarding must be treated as dynamic bits in the sense that these register bits could be changed by the OS when there is traffic flowing through the IOH. Note that the address register themselves can be treated as static in the sense that they will not be changed without the decode control bits being clear. Registers outside of this standard space will be noted as dynamic when appropriate.



Figure 21-1. PCI Express Root Port (Devices 1-10), ESI Port (Device 0) Type1 Configuration Space

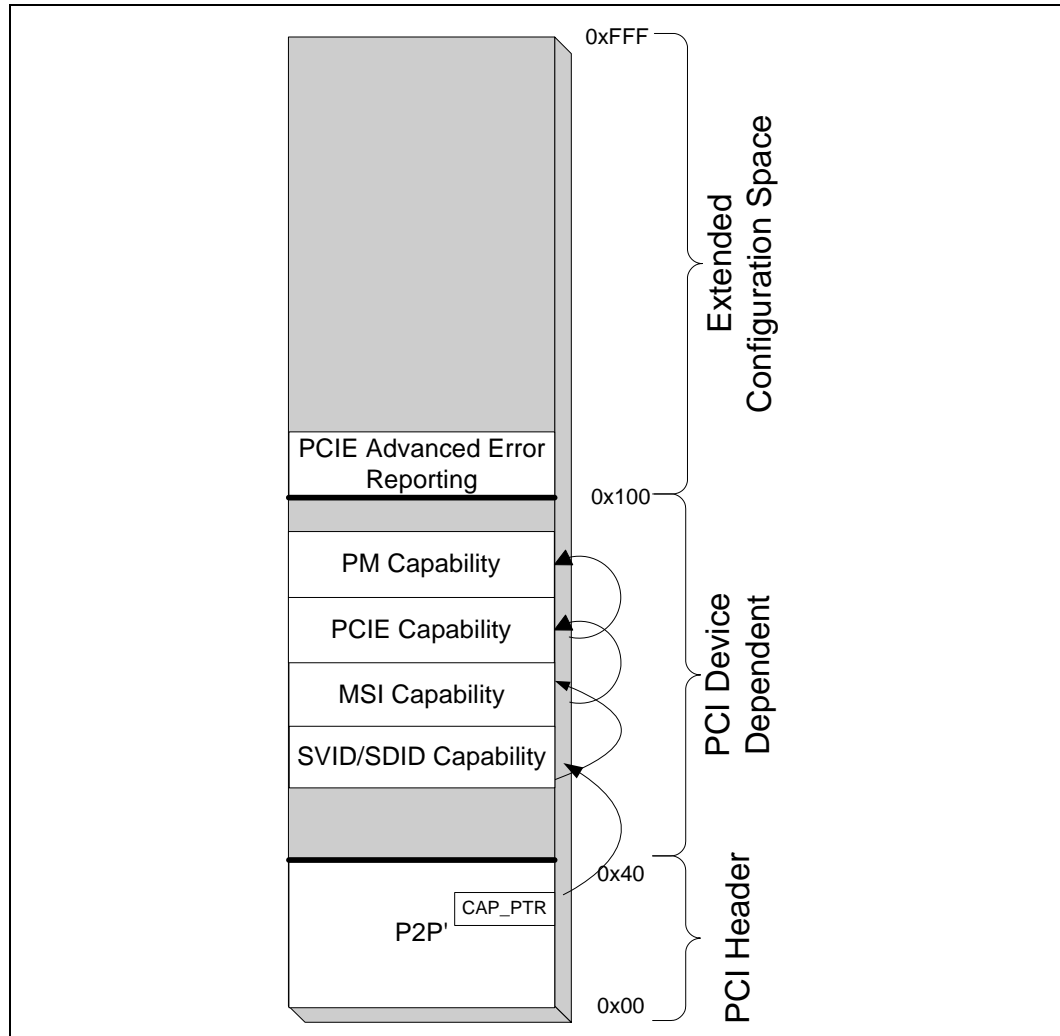


Figure 21-1 illustrates how each PCI Express port's configuration space appears to software. Each PCI Express configuration space has three regions:

- **Standard PCI Header** - This region is the standard PCI-to-PCI bridge header providing legacy OS compatibility and resource management.
- **PCI Device Dependent Region** - This region is also part of standard PCI configuration space and contains the PCI capability structures and other port specific registers. For the IOH, the supported capabilities are:
 - SVID/SDID Capability
 - Message Signalled Interrupts
 - Power Management
 - PCI Express Capability
- **PCI Express Extended Configuration Space** - This space is an enhancement beyond standard PCI and only accessible with PCI Express aware software. The IOH supports the Advanced Error Reporting Capability in this configuration space.



Not all the capabilities listed above for a PCI Express port are required for a ESI port. Through the rest of the chapter, as each register elaborated upon, it will be mentioned which registers are applicable to the PCI Express port and which are applicable to the ESI port.

Table 21-23. IOH Device 0 (ESI mode) Configuration Map

| | | | | | | | | | | | | | | | |
|------------|-----|--------|------|-----|---------|-----------|----------|----------|---------|----------|-----|-------|-----|-----|-----|
| DID | | VID | | 00h | | 80h | | | | | | | | | |
| PCISTS | | PCICMD | | 04h | | 84h | | | | | | | | | |
| CCR | | RID | | 08h | | 88h | | | | | | | | | |
| BIST | HDR | PLAT | CLSR | 0Ch | | 8Ch | | | | | | | | | |
| | | | | 10h | | PXPCAP | PXPNTPTR | PXPCAPID | 90h | | | | | | |
| | | | | 14h | | DEVCAP | | | 94h | | | | | | |
| | | | | 18h | | DEVSTS | | DEVCON | | 98h | | | | | |
| | | | | 1Ch | | LNKCAP | | | 9Ch | | | | | | |
| | | | | 20h | | LNKSTS | | LNKCON | | A0h | | | | | |
| | | | | 24h | | SLTCAP | | | A4h | | | | | | |
| | | | | 28h | SLTSTS | | SLTCON | | A8h | | | | | | |
| | | | | SID | | SVID | | 2Ch | ROOTCAP | ROOTCON | ACh | | | | |
| | | | | | | | | 30h | ROOTSTS | | | B0h | | | |
| | | | | | | | | 34h | DEVCAP2 | | | B4h | | | |
| | | | | 38h | DEVCON2 | | | B8h | | | | | | | |
| | | | | 3Ch | | | | BCh | | | | | | | |
| | | INTPIN | INTL | 40h | LNKSTS2 | | LNKCON2 | C0h | | | | | | | |
| | | | | | | 44h | | | | C4h | | | | | |
| 48h | C8h | | | | | | | | | | | | | | |
| 4Ch | CCh | | | | | | | | | | | | | | |
| 50h | D0h | | | | | | | | | | | | | | |
| 54h | D4h | | | | | | | | | | | | | | |
| 58h | D8h | | | | | | | | | | | | | | |
| 5Ch | DCh | | | | | | | | | | | | | | |
| MSICTL | | | | | | MSINXTPTR | | | | MSICAPID | 60h | PMCAP | | | E0h |
| MSGADR | | | | | | 64h | | | | PMCSR | | | E4h | | |
| | | | | | | MSGDAT | | | | 68h | | | | E8h | |
| | | MSIMSK | | 6Ch | ECh | | | | | | | | | | |
| MSIPENDING | | | | 70h | F0h | | | | | | | | | | |
| | | | | 74h | F4h | | | | | | | | | | |
| | | | | 78h | F8h | | | | | | | | | | |
| | | | | 7Ch | FCh | | | | | | | | | | |



Table 21-24. IOH Device 0 (ESI mode) Extended Configuration Map

| | | | |
|-----------|------|--------------------------------|------|
| ERRCAPHDR | 100h | PERFCTRLSTS | 180h |
| UNCERRSTS | 104h | | 184h |
| UNCERRMSK | 108h | MISCCTRLSTS | 188h |
| UNCERRSEV | 10Ch | | |
| COERRSTS | 110h | PCIE_IOU_BIF_CTRL ¹ | 190h |
| CORERRMSK | 114h | | |
| ERRCAP | 118h | | |
| HDRLOG | 11Ch | | |
| | 120h | | |
| | 124h | | |
| | 128h | | |
| RPERRCMD | 12Ch | | |
| RPERRSTS | 130h | | |
| ERRSID | 134h | | |
| CORSRCID | 138h | | |
| | 13Ch | | |
| APICLIMIT | 140h | | |
| APICBASE | 144h | | |
| | 148h | | |
| | 14Ch | | |
| ACSCAPHDR | 150h | | |
| ACCTRL | 154h | | |
| ACSCAP | 158h | | |
| | 15Ch | | 1DCh |
| | 160h | CTOCTRL | 1E0h |
| | 164h | PCI_LER_SS_CTRLSTS | 1E4h |
| | 168h | | 1E8h |
| | 16Ch | | |
| | 170h | | |
| | 174h | | |
| | 178h | | |
| | 17Ch | | 1FCh |

Note: 1: Applicable only to devices #1, 3, 7.



Table 21-25. IOH Devices 0(ESI Mode) Configuration Map

| | | | | |
|-------------|-----------------|------|--|--|
| XPCORERRSTS | | 200h | | |
| XPCORERRMSK | | 204h | | |
| XPUNCERRSTS | | 208h | | |
| XPUNCERRMSK | | 20Ch | | |
| XPUNCERRSEV | | 210h | | |
| | XPUNCERRP TR | 214h | | |
| | | 218h | | |
| | | 21Ch | | |
| | | 220h | | |
| | | 224h | | |
| | | 228h | | |
| XPGLBERRPTR | | 230h | | |
| XPGLBERRSTS | | 230h | | |
| | | 234h | | |
| | | 238h | | |
| | | 23Ch | | |
| | | 240h | | |
| | | 244h | | |
| | | 248h | | |
| | | 24Ch | | |
| | | 250h | | |
| | | 254h | | |
| | | 258h | | |
| | | 25Ch | | |
| | | 260h | | |
| | | 264h | | |
| 268h | | | | |
| 26Ch | | | | |
| 270h | | | | |
| 274h | | | | |
| 278h | | | | |
| 27Ch | | | | |



Table 21-26. IOH Devices 0(PCIe Mode)-10 Legacy Configuration Map (PCI Express Registers)

| | | | | | | | | | | | | | | | |
|--------|--------|---------|--------|-----------|---------|---------|----------|----------|-----|-------|-----|--|-----|-----|--|
| DID | | VID | | 00h | | 80h | | | | | | | | | |
| PCISTS | | PCICMD | | 04h | | 84h | | | | | | | | | |
| CCR | | | RID | 08h | | 88h | | | | | | | | | |
| BIST | HDR | PLAT | CLSR | 0Ch | | 8Ch | | | | | | | | | |
| | | | | 10h | | PXPCAP | PXPNTPTR | PXPCAPID | 90h | | | | | | |
| | | | | 14h | | DEVCAP | | | 94h | | | | | | |
| | | SUBBUS | SECBUS | PBUS | | 18h | DEVSTS | DEVCON | 98h | | | | | | |
| | | SSTS | IOLIM | IOBAS | | 1Ch | LNKCAP | | | 9Ch | | | | | |
| MLIM | | MBAS | | 20h | | LNKSTS | LNKCON | | A0h | | | | | | |
| PLIM | | PBAS | | 24h | | SLTCAP | | | A4h | | | | | | |
| PBASU | | | | 28h | | SLTSTS | SLTCON | | A8h | | | | | | |
| PLIMU | | | | 2Ch | | ROOTCAP | ROOTCON | | ACH | | | | | | |
| | | | | 30h | | ROOTSTS | | | B0h | | | | | | |
| | | | | 34h | | DEVCAP2 | | | B4h | | | | | | |
| | | | | 38h | | | | | B8h | | | | | | |
| | | | | 3Ch | | | | | BCh | | | | | | |
| BCR | INTPIN | INTL | | 40h | LNKSTS2 | LNKCON2 | | C0h | | | | | | | |
| | | SNXTPTR | SCAPID | 44h | | | | C4h | | | | | | | |
| | | SID | SVID | 48h | | | | C8h | | | | | | | |
| | | | | 4Ch | | | | | | | CCh | | | | |
| | | | | 50h | | | | | | | | | | D0h | |
| | | | | 54h | | | | | | | | | | D4h | |
| | | | | 58h | | | | | | | | | | | |
| 5Ch | | | | DCh | | | | | | | | | | | |
| MSICTL | | | | MSINXTPTR | | | | MSICAPID | 60h | PMCAP | | | E0h | | |
| MSGADR | | | | 64h | PMCSR | | | E4h | | | | | | | |
| | | | | MSGDAT | | 68h | | | | E8h | | | | | |
| | | 6Ch | | | | ECh | | | | | | | | | |
| | | 70h | | | | | | | | F0h | | | | | |
| | | 74h | | | | | | | | | | | F4h | | |
| | | 78h | | | | | | | | | | | | | |
| 7Ch | | | | | | | | | FCh | | | | | | |



**Table 21-27. IOH Devices 0(PCIe Mode)-10 Extended Configuration Map
(PCI Express Registers) Page#0**

| | | | |
|-----------|------|-------------|------|
| ERRCAPHDR | 100h | PERFCTRLSTS | 180h |
| UNCERRSTS | 104h | | 184h |
| UNCERRMSK | 108h | MISCCTRLSTS | 188h |
| UNCERRSEV | 10Ch | | 18Ch |
| CORERRSTS | 110h | | |
| CORERRMSK | 114h | | |
| ERRCAP | 118h | | |
| HDRLOG | 11Ch | | |
| | 120h | | |
| | 124h | | |
| | 128h | | |
| RPERRCMD | 12Ch | | |
| RPERRSTS | 130h | | |
| ERRSID | 134h | | |
| CORSRCID | 134h | | |
| SSMSK | 138h | | |
| | 13Ch | | |
| APICLIMIT | 140h | ADDPICCTRL | 1C0h |
| APICBASE | 140h | | |
| | 144h | | |
| | 148h | | |
| | 14Ch | | |
| ACSCAPHDR | 150h | | |
| ACCTRL | 154h | | |
| ACSCAP | 154h | | |
| | 158h | | |
| | 15Ch | | |
| | 160h | CTOCTRL | 1E0h |
| | 164h | PCIELERCTRL | 1E4h |
| | 168h | | |
| | 16Ch | | |
| | 170h | | |
| | 174h | | |
| | 178h | | |
| | 17Ch | | 1FCh |



Table 21-28. IOH Devices 0-10 Extended Configuration Map (PCI Express Registers)
Page#1

| | | | |
|-------------|-----------------|------|--|
| XPCORERRSTS | | 200h | |
| XPCORERRMSK | | 204h | |
| XPUNCERRSTS | | 208h | |
| XPUNCERRMSK | | 20Ch | |
| XPUNCERRSEV | | 210h | |
| | XPUNCERRP TR | 214h | |
| UNCEMASK | | 218h | |
| COREDMSK | | 21Ch | |
| PREDMSK | | 220h | |
| XPUNCEDMASK | | 224h | |
| XPCOREDMSK | | 228h | |
| | | 22Ch | |
| XPGLBERRPTR | XPGLBERRSTS | 230h | |
| | | 234h | |
| | | 238h | |
| | | 23Ch | |
| | | 240h | |
| | | 244h | |
| | | 248h | |
| | | 24Ch | |
| | | 250h | |
| | | 254h | |
| | | 258h | |
| | | 25Ch | |
| | | 260h | |
| | | 264h | |
| | | 268h | |
| | | 26Ch | |
| | | 270h | |
| | | 274h | |
| | | 278h | |
| | | 27Ch | |



21.12.2 Standard PCI Configuration Space (0x0 to 0x3F) - Type 0/1 Common Configuration Space

This section covers registers in the 0x0 to 0x3F region that are common to all the devices 0 through 11. Comments at the top of the table indicate what devices/functions the description applies to. Exceptions that apply to specific functions are noted in the individual bit descriptions.

21.12.2.1 VID: Vendor Identification Register

| Register: VID Device: 0-10 Function: 0 Offset: 00h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:0 | RO | 8086h | Vendor Identification Number The value is assigned by PCI-SIG to Intel. |

21.12.2.2 DID: Device Identification Register

| Register: DID Device: 0 Function: 0 Offset: 02h | | | |
|--|------------------------------------|-------------|---|
| Bit | Attr | Default | Description |
| 15:0 | RO (Dev#10) , RO (Others) | 0x3400-3407 | Device Identification Number The value is assigned by Intel to each product. IOH will have a unique device id for each device. |

21.12.2.3 DID: Device Identification Register

| Register: DID Device: 1-10 Function: 0 Offset: 02h | | | |
|---|------------------------------------|---|---|
| Bit | Attr | Default | Description |
| 15:0 | RO (Dev#10) , RO (Others) | Dev: Def 1: 3408h 2: 3409h 3: 340Ah 4: 340Bh 5: 340Ch 6: 340Dh 7: 340Eh 8: 340Fh 9: 3410h 10: 3411h | Device Identification Number The value is assigned by Intel to each product. IOH will have a unique device ID for each device. |



21.12.2.4 PCICMD: PCI Command Register (Dev #0 ESI mode)

This register defines the PCI 3.0 compatible command register values applicable to PCI Express space.

| Register: PCICMD Device: 0 (ESI only) Function: 0 Offset: 04h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:11 | RV | 00h | Reserved. (by PCI SIG) |
| 10 | RO | 0 | Not applicable to PCI Express |
| 9 | RO | 0 | Fast Back-to-Back Enable Not applicable to PCI Express and is hardwired to 0 |
| 8 | RW | 0 | SERR Enable For PCI Express/ESI ports, this field enables notifying the internal core error logic of occurrence of an uncorrectable error (fatal or non-fatal) at the port. The internal core error logic of IOH then decides if/how to escalate the error further (pins/message, and so on). This bit also controls the propagation of PCI Express ERR_FATAL and ERR_NONFATAL messages received from the port to the internal IOH core error logic. 1: Fatal and Non-fatal error generation and Fatal and Non-fatal error message forwarding is enabled 0: Fatal and Non-fatal error generation and Fatal and Non-fatal error message forwarding is disabled <i>Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for details of how this bit is used in conjunction with other control bits in the Root Control register for forwarding errors detected on the PCI Express interface to the system core error logic.</i> This bit has no impact on error reporting from the other device - I/OxAPIC. |
| 7 | RO | 0 | IDSEL Stepping/Wait Cycle Control Not applicable to internal IOH devices. Hardwired to 0. |
| 6 | RW | 0 | Parity Error Response For PCI Express/ESI ports, IOH ignores this bit and always does ECC/parity checking and signaling for data/address of transactions both to and from IOH. This bit though affects the setting of bit 8 in the PCISTS (see bit 8 in Section 21.12.2.6) register. |
| 5 | RO | 0 | VGA palette snoop Enable Not applicable to internal IOH devices. Hardwired to 0. |
| 4 | RO | 0 | Memory Write and Invalidate Enable Not applicable to internal IOH devices. Hardwired to 0. |
| 3 | RO | 0 | Special Cycle Enable Not applicable to PCI Express. Hardwired to 0. |
| 2 | RW | 0 | Bus Master Enable Controls the ability of the PCI Express/ESI port in generating/forwarding memory (including MSI writes) or I/O transactions (and not messages) or configuration transactions from the secondary side to the primary side. For I/OxAPIC, this bit enables them to generate memory write/MSI. 1: Enables the PCI Express/ESI port or I/OxAPIC to generate/forward memory, config or I/O read/write requests. 0: The Bus Master is disabled. When this bit is 0, IOH root ports will treat upstream PCI Express memory writes/reads, IO writes/reads, and configuration reads and writes as unsupported requests (and follow the rules for handling unsupported requests). This behavior is also true towards transactions that are already pending in the IOH root port's internal queues when the BME bit is turned off. I/OxAPIC cannot generate any memory transactions when this bit is 0. |



| Register: PCICMD Device: 0 (ESI only) Function: 0 Offset: 04h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 1 | RW | 0 | Memory Space Enable 1: Enables a PCI Express/ESI port's memory range registers and internal I/OxAPIC's MBAR register (ABAR range decode is not enabled by this bit) to be decoded as valid target addresses for transactions from primary side. 0: Disables a PCI Express/ESI port's memory range registers (including the CSR range registers) to be decoded as valid target addresses for transactions from primary side Note that if a PCI Express/ESI port's MSE bit is clear, that port can still be target of any memory transaction if subtractive decoding is enabled on that port. |
| 0 | RW | 0 | IO Space Enable Applies only to PCI Express/ESI ports 1: Enables the I/O address range, defined in the IOBASE and IOLIM registers of the PCI-to-PCI bridge header, for target decode from primary side 0: Disables the I/O address range, defined in the IOBASE and IOLIM registers of the PCI-to-PCI bridge header, for target decode from primary side. Note that if a PCI Express/ESI port's IOSE bit is clear, that port can still be target of an I/O transaction if subtractive decoding is enabled on that port. |

21.12.2.5 PCICMD: PCI Command Register (Dev #0 PCIe mode and Dev #1-10)

This register defines the PCI 3.0 compatible command register values applicable to PCI Express space.

| Register: PCICMD Device: 0 (PCIe only) Device: 1-10 Function: 0 Offset: 04h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:11 | RV | 00h | <i>Reserved.</i> (by PCI SIG) |
| 10 | RO | 0 | Not applicable to PCI Express |
| 9 | RO | 0 | Fast Back-to-Back Enable Not applicable to PCI Express and is hardwired to 0 |
| 8 | RW | 0 | SERR Enable For PCI Express/ESI ports, this field enables notifying the internal core error logic of occurrence of an uncorrectable error (fatal or non-fatal) at the port. The internal core error logic of IOH then decides if/how to escalate the error further (pins/message, and so on). This bit also controls the propagation of PCI Express ERR_FATAL and ERR_NONFATAL messages received from the port to the internal IOH core error logic. 1: Fatal and Non-fatal error generation and Fatal and Non-fatal error message forwarding is enabled 0: Fatal and Non-fatal error generation and Fatal and Non-fatal error message forwarding is disabled Refer to <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s</i> for details of how this bit is used in conjunction with other control bits in the Root Control register for forwarding errors detected on the PCI Express interface to the system core error logic. This bit has no impact on error reporting from the other device - I/OxAPIC. |
| 7 | RO | 0 | IDSEL Stepping/Wait Cycle Control Not applicable to internal IOH devices. Hardwired to 0. |



| Register: PCICMD Device: 0 (PCIe only) Device: 1-10 Function: 0 Offset: 04h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 6 | RW | 0 | Parity Error Response For PCI Express/ESI ports, IOH ignores this bit and always does ECC/parity checking and signaling for data/address of transactions both to and from IOH. This bit though affects the setting of bit 8 in the PCISTS (see bit 8 in Section 21.12.2.6) register. |
| 5 | RO | 0 | VGA palette snoop Enable Not applicable to internal IOH devices. Hardwired to 0. |
| 4 | RO | 0 | Memory Write and Invalidate Enable Not applicable to internal IOH devices. Hardwired to 0. |
| 3 | RO | 0 | Special Cycle Enable Not applicable to PCI Express. Hardwired to 0. |
| 2 | RW | 0 | Bus Master Enable Controls the ability of the PCI Express/ESI port in generating/forwarding memory (including MSI writes) or I/O transactions (and not messages) or configuration transactions from the secondary side to the primary side. For I/OxAPIC, this bit enables them to generate memory write/MSI. 1: Enables the PCI Express/ESI port or I/OxAPIC to generate/forward memory, config or I/O read/write requests. 0: The Bus Master is disabled. When this bit is 0, IOH root ports will treat upstream PCI Express memory writes/reads, IO writes/reads, and configuration reads and writes as unsupported requests (and follow the rules for handling unsupported requests). This behavior is also true towards transactions that are already pending in the IOH root port's internal queues when the BME bit is turned off. I/OxAPIC cannot generate any memory transactions when this bit is 0. |
| 1 | RW | 0 | Memory Space Enable 1: Enables a PCI Express/ESI port's memory range registers and internal I/OxAPIC's MBAR register (ABAR range decode is not enabled by this bit) to be decoded as valid target addresses for transactions from primary side. 0: Disables a PCI Express/ESI port's memory range registers (including the CSR range registers) to be decoded as valid target addresses for transactions from primary side Note that if a PCI Express/ESI port's MSE bit is clear, that port can still be target of any memory transaction if subtractive decoding is enabled on that port. |
| 0 | RW | 0 | IO Space Enable Applies only to PCI Express/ESI ports 1: Enables the I/O address range, defined in the IOBASE and IOLIM registers of the PCI-to-PCI bridge header, for target decode from primary side 0: Disables the I/O address range, defined in the IOBASE and IOLIM registers of the PCI-to-PCI bridge header, for target decode from primary side. Note that if a PCI Express/ESI port's IOSE bit is clear, that port can still be target of an I/O transaction if subtractive decoding is enabled on that port. |

21.12.2.6 PCISTS: PCI Status Register

The PCI Status register is a 16-bit status register which reports the occurrence of various events associated with the primary side of the “virtual” PCI-PCI bridge embedded in PCI Express ports and also primary side of the other devices on the internal IOH bus.



| Register: PCISTS Device: 0-10 Function: 0 Offset: 06h | | | |
|--|--------------------|---------|--|
| Bit | Attr | Default | Description |
| 15 | RW1C | 0 | Detected Parity Error This bit is set by a device when it receives a packet on the primary side with an uncorrectable data error (that is, a packet with poison bit set or an uncorrectable data ECC error was detected at the XP-DP interface when ECC checking is done) or an uncorrectable address/control parity error. The setting of this bit is regardless of the Parity Error Response bit (PERRE) in the PCICMD register. |
| 14 | RW1C (Dev#0-10) | 0 | Signaled System Error 1: The device reported fatal/non-fatal (and not correctable) errors it detected on its PCI Express interface. Software clears this bit by writing a '1' to it. For Express ports this bit is also set (when SERR enable bit is set) when a FATAL/NON-FATAL message is forwarded from the Express link to the ERR[2:0] pins or to ICH via a message. Note that IOH internal 'core' errors (like parity error in the internal queues) are not reported via this bit. 0: The device did not report a fatal/non-fatal error |
| 13 | RW1C | 0 | Received Master Abort This bit is set when a device experiences a master abort condition on a transaction it mastered on the primary interface (IOH internal bus). Note that certain errors might be detected right at the PCI Express interface and those transactions might not 'propagate' to the primary interface before the error is detected (for example, accesses to memory above TOCM in cases where the PCIe interface logic itself might have visibility into TOCM). Such errors do not cause this bit to be set, and are reported via the PCI Express interface error bits (secondary status register). Conditions that cause bit 13 to be set, include: <ul style="list-style-type: none"> • Device receives a completion on the primary interface (internal bus of IOH) with Unsupported Request or master abort completion Status. This includes UR status received on the primary side of a PCI Express port on peer-to-peer completions also. • Device accesses to holes in the main memory address region that are detected by the Intel QuickPath Interconnect source address decoder. • Other master abort conditions detected on the IOH internal bus |
| 12 | RW1C | 0 | Received Target Abort This bit is set when a device experiences a completer abort condition on a transaction it mastered on the primary interface (IOH internal bus). Note that certain errors might be detected right at the PCI Express interface and those transactions might not 'propagate' to the primary interface before the error is detected (for example, accesses to memory above VTCSRBASE). Such errors do not cause this bit to be set, and are reported via the PCI Express interface error bits (secondary status register). Conditions that cause bit 12 to be set, include: <ul style="list-style-type: none"> • Device receives a completion on the primary interface (internal bus of IOH) with completer abort completion Status. This includes CA status received on the primary side of a PCI Express port on peer-to-peer completions also. • Accesses to Intel QuickPath Interconnect that return a failed completion status • Other completer abort conditions detected on the IOH internal bus |
| 11 | RW1C | 0 | Signaled Target Abort This bit is set when a device signals a completer abort completion status on the primary side (internal bus of IOH). This condition includes a PCI Express port forwarding a completer abort status received on a completion from the secondary side and passed to the primary side on a peer-to-peer completion. I/OxAPIC sets this bit when it receives config/memory transactions larger than a DWORD or cross a DWORD boundary. |
| 10:9 | RO | 0h | DEVSEL# Timing Not applicable to PCI Express. Hardwired to 0. |



| Register: PCISTS Device: 0-10 Function: 0 Offset: 06h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 8 | RW1C | 0 | Master Data Parity Error This bit is set by a device if the Parity Error Response bit in the PCI Command register is set and it receives a completion with poisoned data from the primary side or if it forwards a packet with data (including MSI writes) to the primary side with poison. |
| 7 | RO | 0 | Fast Back-to-Back Not applicable to PCI Express. Hardwired to 0. |
| 6 | RV | 0 | <i>Reserved</i> |
| 5 | RO | 0 | 66MHz capable Not applicable to PCI Express. Hardwired to 0. |
| 4 | RO | 1h | Capabilities List This bit indicates the presence of a capabilities list structure |
| 3:0 | RV | 0h | <i>Reserved</i> |

21.12.2.7 RID: Revision Identification Register

This register contains the revision number of the IOH. The revision number steps the same across all devices and functions, that is, individual devices do not step their RID independently. Note that the revision id for the JTAG IDCODE register also steps with this register.

The IOH supports the CRID feature where in this register's value can be changed by BIOS.

| Register: RID Device: 0-10 Function: 0 Offset: 08h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:4 | RO | 0 | Major Revision Steppings which require all masks to be regenerated. 0: A stepping 1: B stepping |
| 3:0 | RO | 0 | Minor Revision Incremented for each stepping which does not modify all masks. Reset for each major revision. 0: x0 stepping 1: x1 stepping 2: x2 stepping |

21.12.2.8 CCR: Class Code Register

This register contains the Class Code for the device.



| Register: CCR Device: 0-10 Function: 0 Offset: 09h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 23:16 | RO | 06h | Base Class For PCI Express/ESI ports, this field is hardwired to 06h, indicating it is a "Bridge Device". For I/OxAPIC devices, this field defaults to 08h, indicating it is a "Generic System Peripherals". |
| 15:8 | RO | 04h | Sub-Class For PCI Express/ESI ports, this field defaults to 04h indicating "PCI-PCI bridge". This register changes to the sub class of 00h to indicate "Host Bridge", when bit 0 in Section 21.6.3, "IOHMISCCTRL: IOH MISC Control Register" on page 336 is set. |
| 7:0 | RO | 00h | Register-Level Programming Interface This field is hardwired to 00h for PCI Express/ESI ports. |

21.12.2.9 CLSR: Cache Line Size Register

| Register: CLSR Device: 0-10 Function: 0 Offset: 0Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RW | 0 | Cache line Size This register is set as RW for compatibility reasons only. Cache line size for IOH is always 64B. IOH hardware ignore this setting. |

21.12.2.10 PLAT: Primary Latency Timer

| Register: PLAT Device: 0-10 Function: 0 Offset: 0Dh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RO | 0h | Prim_Lat_timer: Primary Latency Timer Not applicable to PCI Express. Hardwired to 00h. |

This register denotes the maximum timeslice for a burst transaction in legacy PCI 2.3 on the primary interface. It does not affect/influence PCI Express functionality.

21.12.2.11 HDR: Header Type Register (Dev#0, ESI Mode)

This register identifies the header layout of the configuration space.



| Register: HDR Device: 0 Function: 0 Offset: 0Eh ESI mode only | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7 | RO | 0 | Multi-function Device This bit defaults to 0 for PCI Express/ESI ports. |
| 6:0 | RO | 00h | Configuration Layout This field identifies the format of the configuration header layout. For Device 0 in ESI mode, default is 00h indicating a conventional type 00h PCI header ESI. |

21.12.2.12 HDR: Header Type Register (Dev#0, PCIe Mode and Dev#1-10)

This register identifies the header layout of the configuration space.

| Register: HDR Device: 1-10 Function: 0 Offset: 0Eh PCIe mode only | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7 | RO | 0 | Multi-function Device This bit defaults to 0 for PCI Express/ESI ports. |
| 6:0 | RO | 01h | Configuration Layout This field identifies the format of the configuration header layout. Its Type1 for all PCI Express/ESI ports The default is 01h, indicating a "PCI to PCI Bridge." |

21.12.2.13 BIST: Built-In Self Test

| Register: BIST Device: 0-10 Function: 0 Offset: 0Fh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RO | 0h | BIST_TST: BIST Tests Not supported. Hardwired to 00h |

This register is used for reporting control and status information of BIST checks within a PCI Express port.

21.12.2.14 SVID: Subsystem Vendor ID (Dev#0, ESI Mode)

This register identifies the manufacturer of the system. This 16-bit register combined with the Device Identification Register uniquely identifies any PCI device.



| Register: SVID Device: 0 Function: 0 Offset: 2Ch ESI mode only | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:0 | RWO | 8086h | Vendor Identification Number. The default value specifies Intel. |

21.12.2.15 SID: Subsystem Identity (Dev#0, ESI Mode)

This register identifies the system.

| Register: SID Device: 0 Function: 0 Offset: 2Eh ESI mode only | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:0 | RWO | 00h | Subsystem Identification Number: Assigned by the subsystem vendor to uniquely identify the subsystem. |

21.12.2.16 CAP: Capability Pointer

The CAPPTR is used to point to a linked list of additional capabilities implemented by the device. It provides the offset to the first set of capabilities registers located in the PCI compatible space from 40h.

| Register: CAP Device: 0-10 Function: 0 Offset: 34h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RWO | 40h | Capability Pointer Points to the first capability structure for the device. |

21.12.2.17 INTL: Interrupt Line Register

The Interrupt Line register is used to communicate interrupt line routing information between initialization code and the device driver. This register is not used in newer OSes and is just kept as is.

| Register: INTL Device: 0-10 Function: 0 Offset: 3Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RWO | 00h | Interrupt Line This bit is RO for PCI Express/ESI ports |



21.12.2.18 INTPIN: Interrupt Pin Register

The INTPIN register identifies legacy interrupts for INTA, INTB, INTC and INTD as determined by BIOS/firmware. These are emulated over the ESI port using the appropriate Assert_Intx commands.

| Register: INTPIN Device: 0-10 Function: 0 Offset: 3Dh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RWO | 01h | Interrupt Pin This field defines the type of interrupt to generate for the PCI Express port. 001: Generate INTA 010: Generate INTB 011: Generate INTC 100: Generate INTD Others: Reserved BIOS/configuration Software has the ability to program this register once during boot to set up the correct interrupt for the port. Note: IOH always generates INTA irrespective of INTPIN register configuration. |

21.12.2.19 TSWCTLO : General Control Register

This is a data manager control register.

| Register: TSWCTLO Device: 13 Function: 7 Offset: 30Ch | | | |
|--|------|---------|------------------------|
| Bit | Attr | Default | Description |
| 31:11 | RV | rsvd | Reserved |
| 10 | RW | 0 | global_nosnoop_disable |
| 9:0 | RV | rsvd | Reserved |

21.12.3 Standard PCI Configuration Space (0x0 to 0x3F) - Type 1 - Only Common Configuration Space

This section covers registers that are applicable only to PCI express/ESI ports.

21.12.3.1 PBUS: Primary Bus Number Register

This register identifies the bus number on the on the primary side of the PCI Express port.



| Register: PBUS Device: 0-10 Function: 0 Offset: 18h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RW | 00h | Primary Bus Number Configuration software programs this field with the number of the bus on the primary side of the bridge. This register has to be kept consistent with the IOHBUSNO0/1 register (see Section 21.6.5.12) in the CSRCFG space. BIOS (and OS if internal bus number gets moved) must program this register to the correct value since IOH hardware would depend on this register for inbound decode purposes. |

21.12.3.2 SECBUS: Secondary Bus Number

This register identifies the bus number assigned to the secondary side (PCI Express) of the “virtual” PCI-PCI bridge. This number is programmed by the PCI configuration software to allow mapping of configuration cycles to devices connected to PCI Express.

| Register: SECBUS Device: 0-10 Function: 0 Offset: 19h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RW | 00h | Secondary Bus Number This field is programmed by configuration software to assign a bus number to the secondary bus of the virtual P2P bridge. IOH uses this register to forward a configuration transaction as either a Type 1 or Type 0 to PCI Express. |

21.12.3.3 SUBBUS: Subordinate Bus Number Register

This register identifies the subordinate bus (if any) that resides at the level below the secondary bus of the PCI Express interface. This number is programmed by the PCI configuration software to allow mapping of configuration cycles to devices subordinate to the secondary PCI Express port.

| Register: SUBBUS Device: 0-10 Function: 0 Offset: 1Ah | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RW | 00h | Subordinate Bus Number This register is programmed by configuration software with the number of the highest subordinate bus that is behind the PCI Express port. Any transaction that falls between the secondary and subordinate bus number (both inclusive) of an Express port is forwarded to the express port. |

21.12.3.4 IOBAS: I/O Base Register

The I/O Base and I/O Limit registers define an address range that is used by the PCI Express port to determine when to forward I/O transactions from one interface to the other using the following formula:

$$IO_BASE \leq A[15:12] \leq IO_LIMIT$$



The bottom of the defined I/O address range will be aligned to a 4KB (1KB if EN1K bit is set) boundary while the top of the region specified by IO_LIMIT will be one less than a 4 KB (1KB if EN1K bit is set) multiple. Setting the I/O limit less than I/O base disables the I/O range altogether.

| Register: IOBAS Device: 0-10 Function: 0 Offset: 1Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:4 | RW | 0h | I/O Base Address Corresponds to A[15:12] of the I/O addresses at the PCI Express port. |
| 3:2 | RWL | 0h | When EN1K is set (Refer to Section 21.6.5.6 for definition of EN1K bit), these bits become RW and allow for 1K granularity of I/O addressing, otherwise these are RO. |
| 1:0 | RO | 0h | I/O Address capability IOH supports only 16 bit addressing |

Note that in general the I/O base and limit registers won't be programmed by software without clearing the IOSE bit first.

21.12.3.5 IOLIM: I/O Limit Register

| Register: IOLIM Device: 0-10 Function: 0 Offset: 1Dh | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:4 | RW | 0h | I/O Address Limit Corresponds to A[15:12] of the I/O addresses at the PCI Express port. |
| 3:2 | RWL | 0h | When EN1K is set, these bits become RW and allow for 1K granularity of I/O addressing, otherwise these bits are RO. |
| 1:0 | RO | 0h | I/O Address Limit Capability IOH only supports 16 bit addressing |

21.12.3.6 SECSTS: Secondary Status Register

Secondary Status register is a 16-bit status register that reports the occurrence of various events associated with the secondary side (that is, PCI Express/ESI side) of the “virtual” PCI-PCI bridge.

| Register: SECSTS Device: 0-10 Function: 0 Offset: 1Eh | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15 | RW1C | 0 | Detected Parity Error This bit is set by the IOH whenever it receives a poisoned TLP in the PCI Express port. This bit is set regardless of the state the Parity Error Response Enable bit in the Bridge Control register. |
| 14 | RW1C | 0 | Received System Error This bit is set by the IOH when it receives a ERR_FATAL or ERR_NONFATAL message. |



| Register: SECSTS Device: 0-10 Function: 0 Offset: 1Eh | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 13 | RW1C | 0 | Received Master Abort Status This bit is set when the PCI Express port receives a Completion with "Unsupported Request Completion" Status or when IOH master aborts a Type0 configuration packet that has a non-zero device number. |
| 12 | RW1C | 0 | Received Target Abort Status This bit is set when the PCI Express port receives a Completion with "Completer Abort" Status. |
| 11 | RW1C | 0 | Signaled Target Abort Status This bit is set when the PCI Express port sends a completion packet with a "Completer Abort" Status (including peer-to-peer completions that are forwarded from one port to another) |
| 10:9 | RO | 00 | DEVSEL# Timing Not applicable to PCI Express. Hardwired to 0 |
| 8 | RW1C | 0 | Master Data Parity Error Status This bit is set by the PCI Express port on the secondary side (PCI Express link) if the Parity Error Response Enable bit (PERRE) is set in Bridge Control register and either of the following two conditions occurs: <ul style="list-style-type: none"> The PCI Express port receives a Completion from PCI Express marked poisoned The PCI Express port poisons a packet with data If the Parity Error Response Enable bit in Bridge Control Register is cleared, this bit is never set. |
| 7 | RO | 0 | Fast Back-to-Back Transactions Capable Status Not applicable to PCI Express. Hardwired to 0. |
| 6 | RV | 0 | Reserved |
| 5 | RO | 0 | 66 MHz capability Status Not applicable to PCI Express. Hardwired to 0. |
| 4:0 | RV | 0h | Reserved |

21.12.3.7 MBAS: Memory Base

The Memory Base and Memory Limit registers define a memory mapped I/O non-prefetchable address range (32-bit addresses) and the IOH directs accesses in this range to the PCI Express port based on the following formula:

$$\text{MEMORY_BASE} \leq \text{A}[31:20] \leq \text{MEMORY_LIMIT}$$

The upper 12 bits of both the Memory Base and Memory Limit registers are read/write and corresponds to the upper 12 address bits, A[31:20] of 32-bit addresses. Thus, the bottom of the defined memory address range will be aligned to a 1 MB boundary and the top of the defined memory address range will be one less than a 1 MB boundary. Refer to [Chapter 7](#) for further details on decoding.

| Register: MBAS Device: 0-10 Function: 0 Offset: 20h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:4 | RW | 0h | Memory Base Address Corresponds to A[31:20] of the memory address on the PCI Express port. |
| 3:0 | RV | 0h | Reserved |



Setting the memory limit less than memory base disables the 32-bit memory range altogether.

Note: In general the memory base and limit registers won't be programmed by software without clearing the MSE bit first.

21.12.3.8 MLIM: Memory Limit

| Register: MLIM Device: 0-10 Function: 0 Offset: 22h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:4 | RW | 0h | Memory Limit Address Corresponds to A[31:20] of the memory address that corresponds to the upper limit of the range of memory accesses that will be passed by the PCI Express bridge |
| 3:0 | RV | 0h | <i>Reserved.</i> (by PCI SIG) |

21.12.3.9 PBAS: Prefetchable Memory Base Register

The Prefetchable Memory Base and Memory Limit registers define a memory mapped I/O prefetchable address range (64-bit addresses) which is used by the PCI Express bridge to determine when to forward memory transactions based on the following

Formula:

$$\text{PREFETCH_MEMORY_BASE_UPPER}::\text{PREFETCH_MEMORY_BASE} \leq A[63:20] \leq \text{PREFETCH_MEMORY_LIMIT_UPPER}::\text{PREFETCH_MEMORY_LIMIT}$$

The upper 12 bits of both the Prefetchable Memory Base and Memory Limit registers are read/write and corresponds to the upper 12 address bits, A[31:20] of 32-bit addresses. The bottom of the defined memory address range will be aligned to a 1 MB boundary and the top of the defined memory address range will be one less than a 1 MB boundary.

| Register: PBAS Device: 0-10 Function: 0 Offset: 24h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:4 | RW | 0h | Prefetchable Memory Base Address Corresponds to A[31:20] of the prefetchable memory address on the PCI Express port. |
| 3:0 | RO | 1h | Prefetchable Memory Base Address Capability IOH sets this bit to 01h to indicate 64bit capability. |

The bottom 4 bits of both the Prefetchable Memory Base and Prefetchable Memory Limit registers are read-only, contain the same value, and encode whether or not the bridge supports 64-bit addresses. If these four bits have the value 0h, then the bridge supports only 32 bit addresses. If these four bits have the value 01h, then the bridge supports 64-bit addresses and the Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers hold the rest of the 64-bit prefetchable base and limit addresses respectively.



Setting the prefetchable memory limit less than prefetchable memory base disables the 64-bit prefetchable memory range altogether.

Note that in general the memory base and limit registers won't be programmed by software without clearing the MSE bit first.

21.12.3.10 PLIM: Prefetchable Memory Limit

| Register: PLIM Device: 0-10 Function: 0 Offset: 26h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:4 | RW | 0h | Prefetchable Memory Limit Address Corresponds to A[31:20] of the memory address on the PCI Express bridge |
| 3:0 | RO | 1h | Prefetchable Memory Limit Address Capability IOH sets this field to 01h to indicate 64bit capability. |

21.12.3.11 PBASU: Prefetchable Memory Base (Upper 32 Bits)

The Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers are extensions to the Prefetchable Memory Base and Prefetchable Memory Limit registers to support a 64-bit prefetchable memory address range.

| Register: PBASU Device: 0-10 Function: 0 Offset: 28h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:0 | RW | 0h | Prefetchable Upper 32-bit Memory Base Address Corresponds to A[63:32] of the memory address that maps to the upper base of the prefetchable range of memory accesses that will be passed by the PCI Express bridge. OS should program these bits based on the available physical limits of the system. |

21.12.3.12 PLIMU: Prefetchable Memory Limit (Upper 32 Bits)

| Register: PLIMU Device: 0-10 Function: 0 Offset: 2Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:0 | RW | 0h | Prefetchable Upper 32-bit Memory Limit Address Corresponds to A[63:32] of the memory address that maps to the upper limit of the prefetchable range of memory accesses that will be passed by the PCI Express bridge. OS should program these bits based on the available physical limits of the system. |



21.12.3.13 BCR: Bridge Control Register

The Bridge Control register provides additional control for the secondary interface (that is, PCI Express) as well as some bits that affect the overall behavior of the “virtual” PCI-PCI bridge embedded within the IOH, for example, VGA compatible address range mapping.

| Register: BCR Device: 0-10 Function: 0 Offset: 3Eh | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:12 | RV | 0h | <i>Reserved</i> |
| 11 | RO | 0 | Discard Timer SERR Status Not applicable to PCI Express. This bit is hardwired to 0. |
| 10 | RO | 0 | Discard Timer Status Not applicable to PCI Express. This bit is hardwired to 0. |
| 9 | RO | 0 | Secondary Discard Timer Status Not applicable to PCI Express. This bit is hardwired to 0. |
| 8 | RO | 0 | Primary Discard Timer Not applicable to PCI Express. This bit is hardwired to 0. |
| 7 | RO | 0 | Fast Back-to-Back Enable Not applicable to PCI Express. This bit is hardwired to 0. |
| 6 | RW | 0 | Secondary Bus Reset 1: Setting this bit triggers a hot reset on the link for the corresponding PCI Express port and the PCI Express hierarchy domain subordinate to the port. This sends the LTSSM into the Training (or Link) Control Reset state, which necessarily implies a reset to the downstream device and all subordinate devices. The transaction layer corresponding to port will be emptied by IOH when this bit is set. This means that in the outbound direction, all posted transactions are dropped and non-posted transactions are sent a UR response. In the inbound direction, completions for inbound NP requests are dropped when they arrive. Inbound posted writes are required to be flushed as well either by dropping the packets or by retiring them normally. Note also that a secondary bus reset will not reset the virtual PCI-to-PCI bridge configuration registers of the targeted PCI Express port. 0: No reset happens on the PCI Express port |
| 5 | RO | 0 | Master Abort Mode Not applicable to PCI Express. This bit is hardwired to 0. |
| 4 | RW | 0 | VGA 16-bit decode This bit enables the virtual PCI-to-PCI bridge to provide 16-bit decoding of VGA I/O address precluding the decoding of alias addresses every 1 KB. 0: execute 10-bit address decodes on VGA I/O accesses. 1: execute 16-bit address decodes on VGA I/O accesses. This bit only has meaning if bit 3 of this register is also set to 1, enabling VGA I/O decoding and forwarding by the bridge. Refer to <i>PCI-PCI Bridge Specification Revision 1.2</i> for further details of this bit behavior. |
| 3 | RW | 0 | VGA Enable Controls the routing of CPU initiated transactions targeting VGA compatible I/O and memory address ranges. This bit must only be set for one PCI Express port. |



| Register: BCR Device: 0-10 Function: 0 Offset: 3Eh | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 2 | RW | 0 | ISA Enable Modifies the response by the IOH to an I/O access issued by the CPU that target ISA I/O addresses. This applies only to I/O addresses that are enabled by the IOBASE and IOLIM registers. 1: The IOH will <i>not</i> forward to PCI Express any I/O transactions addressing the last 768 bytes in each 1KB block even if the addresses are within the range defined by the IOBASE and IOLIM registers. 0: All addresses defined by the IOBASE and IOLIM for CPU I/O transactions will be mapped to PCI Express. |
| 1 | RW | 0 | SERR Enable This bit controls forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL messages from the PCI Express port to the primary side. 1: Enables forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL messages. 0: Disables forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL |
| 0 | RW | 0 | Parity Error Response Enable IOH ignores this bit. This bit though affects the setting of bit 8 in the SECSTS register. |

21.12.4 Device-Specific PCI Configuration Space - 0x40 to 0xFF

21.12.4.1 SCAPID: Subsystem Capability ID

| Register: SCAPID Device: 0-10 Function: 0 Offset: 40h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RO | 0Dh | Capability ID Assigned by PCI-SIG for subsystem capability ID |

21.12.4.2 SNXTPTR: Subsystem ID Next Pointer

| Register: SNXTPTR Device: 0-10 Function: 0 Offset: 41h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RO | 60 | Next Ptr This field is set to 60h for the next capability list (MSI capability structure) in the chain. |



21.12.4.3 SVID: Subsystem Vendor ID

| Register: SVID Device: 0-10 Function: 0 Offset: 44h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RWO | 8086h | Assigned by PCI-SIG for the subsystem vendor |

21.12.4.4 SID: Subsystem Identity (Dev#0, PCIe mode and Dev#1-10)

| Register: SID Device: 0-10 Function: 0 Offset: 46h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RWO | 00h | Assigned by the subsystem vendor to uniquely identify the subsystem |

21.12.4.5 MSICAPID: MSI Capability ID

| Register: MSICAPID Device: 1-10; N/A for 0 Function: 0 Offset: 60h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RO | 05h | Capability ID Assigned by PCI-SIG for MSI (root ports). |

21.12.4.6 MSINXTPTR: MSI Next Pointer

| Register: MSINXTPTR Device: 1-10; N/A for 0 Function: 0 Offset: 61h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RO | 90h | Next Ptr This field is set to 90h for the next capability list (PCI Express capability structure) in the chain. |



21.12.4.7 MSICTL: MSI Control Register

| Register: MSICTL Device: 1-10; N/A for 0 Function: 0 Offset: 62h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:9 | RV | 00h | <i>Reserved.</i> |
| 8 | RO | 1 | Per-vector masking capable This bit indicates that PCI Express ports support MSI per-vector masking |
| 7 | RO | 0 | 64-bit Address Capable This field is hardwired to 0h since the message addresses are only 32-bit addresses (for example, FEEx_xxxxh). |
| 6:4 | RW | 000 | Multiple Message Enable Applicable only to PCI Express ports. Software writes to this field to indicate the number of allocated messages which is aligned to a power of two. When MSI is enabled, the software will allocate at least one message to the device. A value of 000 indicates 1 message. Any value greater than or equal to 001 indicates a message of 2. |
| 3:1 | RO | 001 | Multiple Message Capable IOH's Express ports support two messages for all their internal events. |
| 0 | RW | 0 | MSI Enable The software sets this bit to select platform-specific interrupts or transmit MSI messages. 0: Disables MSI from being generated. 1: Enables the Express port to use MSI messages for RAS, provided bit 4 in Section 21.6.3, "IOHMISCCTRL: IOH MISC Control Register" on page 336 is clear and also enables the Express port to use MSI messages for PM and HP events at the root port provided these individual events are not enabled for ACPI handling (see Section 21.6.3, "IOHMISCCTRL: IOH MISC Control Register" on page 336) for details. |

21.12.4.8 MSIAR: MSI Address Register

The MSI Address Register (MSIAR) contains the system specific address information to route MSI interrupts from the root ports and is breaks into their constituent fields where interrupts are located.

| Register: MSIAR Device: 1-10 Function: 0 Offset: 64h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:20 | RW | 0000h | Address MSB This field specifies the 12 most significant bits of the 32-bit MSI address. This field is R/W for compatibility reasons only. |
| 19:12 | RW | 00h | Address Destination ID This field is initialized by software for routing the interrupts to the appropriate destination. |
| 11:4 | RW | 00h | Address Extended Destination ID This field is not used by IA-32 processor. |
| 3 | RW | 0h | Address Redirection Hint 0: Directed 1: Redirectable |



| Register: MSIAR Device: 1-10 Function: 0 Offset: 64h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 2 | RW | 0h | Address Destination Mode 0: Physical 1: Logical |
| 1:0 | RV | 0h | <i>Reserved.</i> |

21.12.4.9 MSIDR: MSI Data Register

The MSI Data Register contains all the data (interrupt vector) related to MSI interrupts from the root ports.

| Register: MSIDR Device: 1-10 Function: 0 Offset: 68h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:16 | RV | 0h | <i>Reserved.</i> |
| 15 | RW | 0h | Trigger Mode 0 - Edge Triggered 1 - Level Triggered IOH does nothing with this bit other than passing it along to Intel® QPI |
| 14 | RW | 0h | Level 0 - Deassert 1: Assert IOH does nothing with this bit other than passing it along to Intel® QPI |
| 13:12 | RV | 0h | <i>Reserved</i> |
| 11:8 | RW | 0h | Delivery Mode 0000 – Fixed: Trigger Mode can be edge or level. 0001 – Lowest Priority: Trigger Mode can be edge or level. 0010 – SMI/PMI/MCA - Not supported via MSI of root port 0011 – Reserved - Not supported via MSI of root port 0100 – NMI - Not supported via MSI of root port 0101 – INIT - Not supported via MSI of root port 0110 – <i>Reserved</i> 0111 – ExtINT - Not supported via MSI of root port 1000-1111 - <i>Reserved</i> |
| 7:0 | RW | 0h | Interrupt Vector The interrupt vector (LSB) will be modified by the IOH to provide context sensitive interrupt information for different events that require attention from the processor, for example, Hot plug, Power Management and RAS error events. Depending on the number of Messages enabled by the processor, Table 21-2 illustrates how IOH distributes these vectors |



Table 21-29. MSI Vector Handling and Processing by IOH

| Number of Messages enabled by Software | Events | IV[7:0] |
|--|--------|----------------------|
| 1 | All | xxxxxxx ¹ |
| 2 | HP, PM | xxxxxx0 |
| | AER | xxxxxx1 |

Notes:

1. The term “xxxxxx” in the Interrupt vector denotes that software initializes them and IOH will not modify any of the “x” bits except the LSB as indicated in the table as a function of MMEN.

21.12.4.10 MSIMSK: MSI Mask Bit Register

The Mask Bit register enables software to disable message sending on a per-vector basis.

| Register: MSIMSK Device: 0-10 Function: 0 Offset: 6Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:2 | RV | 0h | <i>Reserved</i> |
| 1:0 | RW | 0h | Mask Bits : For each mask bit that is set, the PCI Express port is prohibited from sending the associated message. |

21.12.4.11 MSIPENDING: MSI Pending Bit Register

The Mask Pending register enables software to defer message sending on a per-vector basis.

| Register: MSIPENDING Device: 0-10 Function: 0 Offset: 70h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:2 | RV | 0h | <i>Reserved</i> |
| 1:0 | RO | 0h | Pending Bits : For each pending bit that is set, the PCI Express port has a pending associated message. |

21.12.4.12 PXPCAPID: PCI Express Capability List Register

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 3.0 configuration space.



| Register: PXPCAPID Device: 0-10 Function: 0 Offset: 90h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:0 | RO | 10h | Capability ID Provides the PCI Express capability ID assigned by PCI-SIG. |

21.12.4.13 PXPNTPTR: PCI Express Next Capability List Register

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 3.0 configuration space.

| Register: PXPNTPTR Device: 0-10 Function: 0 Offset: 91h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 7:0 | RO | E0h | Next Ptr This field is set to the PCI PM capability. |

21.12.4.14 PXPCAP: PCI Express Capabilities Register

The PCI Express Capabilities register identifies the PCI Express device type and associated capabilities.

| Register: PXPCAP Device: 0-10 Function: 0 Offset: 92h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:14 | RV | 0h | <i>Reserved</i> |
| 13:9 | RO | 00h | Interrupt Message Number This field indicates the interrupt message number that is generated for PM/HP events. When there are more than one MSI interrupt Number, this register field is required to contain the offset between the base Message Data and the MSI Message that is generated when the status bits in the slot status register or root port status registers are set. IOH assigns the first vector for PM/HP events and so this field is set to 0. |
| 8 | RWO | 0 | Slot Implemented Applies only to the root ports: 1: indicates that the PCI Express link associated with the port is connected to a slot. 0: indicates no slot is connected to this port. This register bit is of type "write once" and is controlled by BIOS/special initialization firmware. |
| 7:4 | RO | 0100 | Device/Port Type This field identifies the type of device. It is set to 0100 for all the PCI Express ports. |
| 3:0 | RO | 2h | Capability Version This field identifies the version of the PCI Express capability structure. Set to 2h for PCI Express for compliance with the extended base registers. |



21.12.4.15 DEVCAP: PCI Express Device Capabilities Register

The PCI Express Device Capabilities register identifies device specific information for the device.

| Register: DEVCAP Device: 0-10 Function: 0 Offset: 94h | | | |
|--|------|------------------------------|--|
| Bit | Attr | Default | Description |
| 31:28 | RV | 0h | <i>Reserved</i> |
| 27:26 | RO | 0h | Captured Slot Power Limit Scale Does not apply to root ports or integrated devices |
| 25:18 | RO | 00h | Captured Slot Power Limit Value Does not apply to root ports or integrated devices |
| 17:16 | RV | 0h | <i>Reserved</i> |
| 15 | RO | 1 | Role Based Error Reporting: IOH is 1.1 compliant and so supports this feature |
| 14 | RO | 0 | Power Indicator Present on Device Does not apply to root ports or integrated devices |
| 13 | RO | 0 | Attention Indicator Present Does not apply to root ports or integrated devices |
| 12 | RO | 0 | Attention Button Present Does not apply to root ports or integrated devices |
| 11:9 | RO | 000 | Endpoint L1 Acceptable Latency Does not apply to IOH |
| 8:6 | RO | 000 | Endpoint L0s Acceptable Latency Does not apply to IOH |
| 5 | RO | 1 | Extended Tag Field Supported IOH devices support 8-bit tag |
| 4:3 | RO | 0h | Phantom Functions Supported IOH does not support phantom functions. |
| 2:0 | RO | 001 (Dev# 1-10), 000 (Dev#0) | Max Payload Size Supported IOH supports 256B payloads on Express port and 128B on the remainder of the devices. |

21.12.4.16 DEVCTRL: PCI Express Device Control Register

The PCI Express Device Control register controls PCI Express specific capabilities parameters associated with the device.

| Register: DEVCTRL Device: 0-10 Function: 0 Offset: 98h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15 | RV | 0h | <i>Reserved</i> |
| 14:12 | RO | 000 | Max_Read_Request_Size Express/ESI ports in IOH do not generate requests greater than 128B and this field is ignored. |



| Register: DEVCTRL Device: 0-10 Function: 0 Offset: 98h | | | |
|---|-----------------------------------|---------|---|
| Bit | Attr | Default | Description |
| 11 | RO | 0 | Enable No Snoop Not applicable to root ports since they never set the 'No Snoop' bit for transactions they originate (not forwarded from peer) to PCI Express. This bit has no impact on forwarding of NoSnoop attribute on peer requests. |
| 10 | RO | 0 | Auxiliary Power Management Enable Not applicable to IOH |
| 9 | RO | 0 | Phantom Functions Enable Not applicable to IOH since it never uses phantom functions as a requester. |
| 8 | RW | 0h | Extended Tag Field Enable This bit enables the PCI Express port/ESI to use an 8-bit Tag field as a requester. |
| 7:5 | RW (Dev#1-10) RO (Dev#0) | 000 | Max Payload Size This field is set by configuration software for the maximum TLP payload size for the PCI Express port. As a receiver, the IOH must handle TLPs as large as the set value. As a requester (that is, for requests where IOH's own RequesterID is used), it must not generate TLPs exceeding the set value. Permissible values that can be programmed are indicated by the Max_Payload_Size_Supported in the Device Capabilities register: 000: 128B max payload size 001: 256B max payload size (applies only to standard PCI Express ports and other devices alias to 128B) others: alias to 128B |
| 4 | RO | 0 | Enable Relaxed Ordering Not applicable to root ports since they never set relaxed ordering bit as a requester (this does not include tx forwarded from peer devices). This bit has no impact on forwarding of relaxed ordering attribute on peer requests. |
| 3 | RW | 0 | Unsupported Request Reporting Enable Applies only to the PCI Express/ESI ports. This bit controls the reporting of unsupported requests that IOH itself detects on requests its receives from a PCI Express/ESI port. 0: Reporting of unsupported requests is disabled 1: Reporting of unsupported requests is enabled. Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for complete details of how this bit is used in conjunction with other bits to UR errors. |
| 2 | RW | 0 | Fatal Error Reporting Enable Applies only to the PCI Express/ESI ports. Controls the reporting of fatal errors that IOH detects on the PCI Express/ESI interface. 0: Reporting of Fatal error detected by device is disabled 1: Reporting of Fatal error detected by device is enabled Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for complete details of how this bit is used in conjunction with other bits to report errors. For the PCI Express/ESI ports, this bit is not used to control the reporting of other internal component uncorrectable fatal errors (at the port unit) in any way. |



| Register: DEVCTRL Device: 0-10 Function: 0 Offset: 98h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 1 | RW | 0 | Non Fatal Error Reporting Enable Applies only to the PCI Express/ESI ports. Controls the reporting of non-fatal errors that IOH detects on the PCI Express/ESI interface. 0: Reporting of Non Fatal error detected by device is disabled 1: Reporting of Non Fatal error detected by device is enabled Refer to <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s</i> for complete details of how this bit is used in conjunction with other bits to report errors. For the PCI Express/ESI ports, this bit is not used to control the reporting of other internal component uncorrectable non-fatal errors (at the port unit) in any way. |
| 0 | RW | 0 | Correctable Error Reporting Enable Applies only to the PCI Express/ESI ports. Controls the reporting of correctable errors that IOH detects on the PCI Express/ESI interface 0: Reporting of link Correctable error detected by the port is disabled 1: Reporting of link Correctable error detected by port is enabled Refer to <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s</i> for complete details of how this bit is used in conjunction with other bits to report errors. For the PCI Express/ESI ports, this bit is not used to control the reporting of other internal component correctable errors (at the port unit) in any way. |

21.12.4.17 DEVSTS: PCI Express Device Status Register

The PCI Express Device Status register provides information about PCI Express device specific parameters associated with the device.

| Register: DEVSTS Device: 0-10 Function: 0 Offset: 9Ah | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:6 | RV | 0h | <i>Reserved.</i> |
| 5 | RO | 0h | Transactions Pending: Does not apply to root/ESI ports or I/OxAPIC devices, that is, bit hardwired to 0 for these devices. |
| 4 | RO | 0 | AUX Power Detected Does not apply to IOH |
| 3 | RW1C | 0 | Unsupported Request Detected This bit applies only to the root/ESI ports and does not apply to I/OxAPIC device. This bit indicates that the root port detected an Unsupported Request. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control Register. 1: Unsupported Request detected at the device/port. These unsupported requests are NP requests inbound that the root port received and it detected them as unsupported requests (for example, address decoding failures that the root port detected on a packet, receiving inbound lock reads, BME bit is clear, and so on). Note that this bit is not set on peer-to-peer completions with UR status that are forwarded by the root port to the PCIe link. 0: No unsupported request detected by the root port |
| 2 | RW1C | 0 | Fatal Error Detected This bit indicates that a fatal (uncorrectable) error is detected by the device. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. 1: Fatal errors detected 0: No Fatal errors detected |



| Register: DEVSTS Device: 0-10 Function: 0 Offset: 9Ah | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 1 | RW1C | 0 | Non Fatal Error Detected This bit gets set if a non-fatal uncorrectable error is detected by the device. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. 1: Non Fatal errors detected 0: No non-Fatal Errors detected |
| 0 | RW1C | 0 | Correctable Error Detected This bit gets set if a correctable error is detected by the device. Errors are logged in this register regardless of whether error reporting is enabled or not in the PCI Express Device Control register. 1: correctable errors detected 0: No correctable errors detected |

21.12.4.18 LNKCAP: PCI Express Link Capabilities Register

The Link Capabilities register identifies the PCI Express specific link capabilities.

| Register: LNKCAP Device: 0-10 Function: 0 Offset: 9Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:24 | RWO | 0 | Port Number This field indicates the PCI Express port number for the link and is initialized by software/BIOS. |
| 23:22 | RV | 0h | <i>Reserved.</i> |
| 21 | RO | 1 | Link Bandwidth Notification Capability - A value of 1b indicates support for the Link Bandwidth Notification status and interrupt mechanisms. |
| 20 | RO | 1 | Data Link Layer Link Active Reporting Capable: IOH supports reporting status of the data link layer so software knows when it can enumerate a device on the link or otherwise know the status of the link. |
| 19 | RO | 1 | Surprise Down Error Reporting Capable: IOH supports reporting a surprise down error condition |
| 18 | RO | 0 | Clock Power Management: Does not apply to IOH. |
| 17:15 | RO | 010h | L1 Exit Latency This field indicates the L1 exit latency for the given PCI Express port. It indicates the length of time this port requires to complete transition from L1 to L0. 000: Less than 1 001: 1 is to less than 2 010: 2 is to less than 4 011: 4 is to less than 8 100: 8 is to less than 16 101: 16 is to less than 32 110: 32 is to 64 111: More than 64us |



| Register: LNKCAP Device: 0-10 Function: 0 Offset: 9Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 14:12 | RO | 7h | L0s Exit Latency |
| 11:10 | RO | 1 | Active State Link PM Support L0s and L1 is supported |
| 9:4 | RO | 0 | Maximum Link Width This field indicates the maximum width of the given PCI Express Link attached to the port. 000001: x1 000010: x2 ¹ 000100: x4 001000: x8 010000: x16 Others - <i>Reserved</i> This is left as a RWO register for bios to update based on the platform usage of the links. |
| 3:0 | RO | 0 | Link Speeds Supported IOH supports both 2.5 Gbps and 5 Gbps speeds if Gen2_OFF is OFF else it supports only Gen1 This register is RWO when Gen2_OFF is OFF, so that BIOS can change the supported speeds field to be 0001b (Gen1 only) if the board routing is not capable of Gen2 (even though IOH silicon itself is capable of Gen2) |

Notes:

1. There are restrictions with routing x2 lanes from IOH to a slot. See [Chapter 5](#) for details.

21.12.4.19 LNKCON: PCI Express Link Control Register

The PCI Express Link Control register controls the PCI Express Link specific parameters.

| Register: LNKCON Device: 0-10 Function: 0 Offset: A0h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:12 | RV | 0 | Reserved |
| 11 | RW | 0 | Link Autonomous Bandwidth Interrupt Enable - When set to 1b, this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been set. |
| 10 | RW | 0 | Link Bandwidth Management Interrupt Enable - When set to 1b, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been set. |
| 9 | RW | 0 | Hardware Autonomous Width Disable - IOH never changes a configured link width for reasons other than reliability. |
| 8 | RO | 0 | Enable Clock Power Management N/A to IOH |
| 7 | RW | 0 | Extended Synch When this bit set forces the transmission of additional ordered sets when exiting L0s and when in recovery. See PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for details. |



| Register: LNKCON Device: 0-10 Function: 0 Offset: A0h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 6 | RW | 0 | Common Clock Configuration When set, this bit indicates that this component and the component at the opposite side of the Link are operating with a distributed common reference clock. A value of 0b indicates this component and the component at the opposite end of this Link are operating with asynchronous reference clock. Components utilize this common clock configuration information to report the correct L0s and L1 Exit Latencies. After changing the value in this bit in both components on a Link, software must trigger the Link to retrain by writing a 1b to the Retrain Link bit. |
| 5 | WO | 0 | Retrain Link A write of 1 to this bit initiates link retraining in the given PCI Express port by directing the LTSSM to the recovery state if the current state is [L0, L0s or L1]. If the current state is anything other than L0, L0s, L1 then a write to this bit does nothing. This bit always returns 0 when read. If the Target Link Speed field has been set to a non-zero value different than the current operating speed, then the LTSSM will attempt to negotiate to the target link speed. It is permitted to write 1b to this bit while simultaneously writing modified values to other fields in this register. When this is done, all modified values that affect link retraining must be applied in the subsequent retraining. |
| 4 | RW | 0 | Link Disable This field controls whether the link associated with the PCI Express port is enabled or disabled. When this bit is a 1, a previously configured link (a link that has gone past the polling state) would return to the "disabled" state as defined in the <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s</i> . When this bit is clear, an LTSSM in the "disabled" state goes back to the detect state. 0: Enables the link associated with the PCI Express port 1: Disables the link associated with the PCI Express port |
| 3 | RO | 0 | Read Completion Boundary Set to zero to indicate IOH could return read completions at 64B boundaries. |
| 2 | RV | 0 | <i>Reserved.</i> |
| 1:0 | RW | 00 | Active State Link PM Control: Writing to these bits has no effect in ESI mode. Device 1-10: When 00, Disabled Device 1-10: When 01, L0s on transmitter is enabled Device 1-10: When 10, L1 on transmitter is enabled Device 1-10: When 11, L0s and L1 on transmitter is enabled |

21.12.4.20 LNKSTS: PCI Express Link Status Register

The PCI Express Link Status register provides information on the status of the PCI Express Link such as negotiated width, training, and so on.



| Register: LNKSTS Device: 0-10 Function: 0 Offset: A2h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15 | RO | 0 | <p>Link Autonomous Bandwidth Status - This bit is set to 1b by hardware to indicate that hardware has autonomously changed link speed or width, without the port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable link operation.</p> <p>IOH sets this bit when it receives eight consecutive TS1 or TS2 ordered sets with the Autonomous Change bit set.</p> <p>Note that if the status bit is set by hardware in the same clock software clears the status bit, the status bit should remain set and if MSI is enabled, the hardware should trigger a new MSI.</p> |
| 14 | RO | 0 | <p>Link Bandwidth Management Status - This bit is set to 1b by hardware to indicate that either of the following has occurred without the port transitioning through DL_Down status:</p> <p>a) A link retraining initiated by a write of 1b to the Retrain Link bit has completed</p> <p>b) Hardware has autonomously changed link speed or width to attempt to correct unreliable link operation</p> <p>Note that if the status bit is set by hardware in the same clock software clears the status bit, the status bit should remain set and if MSI is enabled, the hardware should trigger a new MSI.</p> |
| 13 | RO | 0 | <p>Data Link Layer Link Active</p> <p>Set to 1b when the Data Link Control and Management State Machine is in the DL_Active state, 0b otherwise.</p> <p>On a downstream port or upstream port, when this bit is 0b, the transaction layer associated with the link will abort all transactions that would otherwise be routed to that link.</p> |
| 12 | RO | 1 | <p>Slot Clock Configuration</p> <p>This bit indicates whether IOH receives clock from the same xtal that also provides clock to the device on the other end of the link.</p> <p>1: indicates that same xtal provides clocks to devices on both ends of the link</p> <p>0: indicates that different xtals provide clocks to devices on both ends of the link</p> |
| 11 | RO | 0 | <p>Link Training</p> <p>This field indicates the status of an ongoing link training session in the PCI Express port</p> <p>0: LTSSM has exited the recovery/configuration state</p> <p>1: LTSSM is in recovery/configuration state or the Retrain Link was set but training has not yet begun.</p> <p>The IOH hardware clears this bit once LTSSM has exited the recovery/configuration state. Refer to <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s</i> for details of which states within the LTSSM would set this bit and which states would clear this bit.</p> |
| 10 | RV | 0 | Reserved |
| 9:4 | RO | 0x0 | <p>Negotiated Link Width</p> <p>This field indicates the negotiated width of the given PCI Express link after training is completed. Only x1, x2, x4, x8 and x16 link width negotiations are possible in IOH. A value of 0x01 in this field corresponds to a link width of x1, 0x02 indicates a link width of x2 and so on, with a value of 0x8 for a link width of x8.</p> <p>The value in this field is reserved and could show any value when the link is not up. Software determines if the link is up or not by reading bit 13 of this register.</p> |



| Register: LNKSTS Device: 0-10 Function: 0 Offset: A2h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 3:0 | RO | 0x0 | Current Link Speed This field indicates the negotiated Link speed of the given PCI Express Link. 0001- 2.5 Gbps 0010 - 5 Gbps Others - <i>Reserved</i> The value in this field is not defined and could show any value, when the link is not up. Software determines if the link is up or not by reading bit 13 of this register. |

21.12.4.21 SLTCAP: PCI Express Slot Capabilities Register

The Slot Capabilities register identifies the PCI Express specific slot capabilities. These registers must be ignored by software on the ESI links.

| Register: SLTCAP Device: 0-10 Function: 0 Offset: A4h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:19 | RO | 0h | Physical Slot Number This field indicates the physical slot number of the slot connected to the PCI Express port and is initialized by bios. |
| 18 | RO | 0h | Command Complete Not Capable: IOH is capable of command complete interrupt. |
| 17 | RO | 0h | Electromechanical Interlock Present This bit when set indicates that an Electromechanical Interlock is implemented on the chassis for this slot and that lock is controlled by bit 11 in Slot Control register. Bios note: this capability is not set if the Electromechanical Interlock control is connected to main slot power control. |
| 16:15 | RO | 0h | Slot Power Limit Scale This field specifies the scale used for the Slot Power Limit Value and is initialized by bios. IOH uses this field when it sends a Set_Slot_Power_Limit message on PCI Express. Range of Values: 00: 1.0x 01: 0.1x 10: 0.01x 11: 0.001x |
| 14:7 | RO | 00h | Slot Power Limit Value This field specifies the upper limit on power supplied by slot in conjunction with the Slot Power Limit Scale value defined previously Power limit (in Watts) = SPLS x SPLV. This field is initialized by bios. IOH uses this field when it sends a Set_Slot_Power_Limit message on PCI Express. Design note: IOH can chose to send the Set_Slot_Power_Limit message on the link at first link up condition without regards to whether this register and the Slot Power Limit Scale register are programmed yet by bios. IOH must then be designed to discard a received Set_Slot_Power_Limit message without an error. |
| 6 | RO | 0h | Hot-plug Capable This field defines hot-plug support capabilities for the PCI Express port. 0: indicates that this slot is not capable of supporting Hot-plug operations. 1: indicates that this slot is capable of supporting Hot-plug operations This bit is programed by BIOS based on the system design. This bit must be programmed by bios to be consistent with the VPP enable bit for the port. |



| Register: SLTCAP Device: 0-10 Function: 0 Offset: A4h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 5 | RO | 0h | <p>Hot-plug Surprise</p> <p>This field indicates that a device in this slot may be removed from the system without prior notification (like for instance a PCI Express cable).</p> <p>0: indicates that hot-plug surprise is not supported 1: indicates that hot-plug surprise is supported</p> <p>Note that if platform implemented cable solution (either direct or via a SIOM with repeater), on a port, then this could be set. BIOS programs this field with a 0 for CEM/SIOM FFs.</p> <p>This bit is used by IOH hardware to determine if a transition from DL_active to DL_Inactive is to be treated as a surprise down error or not. If a port is associated with a hotpluggable slot and the hotplug surprise bit is set, then any transition to DL_Inactive is not considered an error. Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*'s for further details.</p> |
| 4 | RO | 0h | <p>Power Indicator Present</p> <p>This bit indicates that a Power Indicator is implemented for this slot and is electrically controlled by the chassis.</p> <p>0: indicates that a Power Indicator, which is electrically controlled by the chassis is not present 1: indicates that Power Indicator, which is electrically controlled by the chassis is present</p> <p>BIOS programs this field with a 1 for CEM/SIOM FFs and a 0 for Express cable.</p> |
| 3 | RO | 0h | <p>Attention Indicator Present</p> <p>This bit indicates that an Attention Indicator is implemented for this slot and is electrically controlled by the chassis</p> <p>0: indicates that an Attention Indicator, which is electrically controlled by the chassis is not present 1: indicates that an Attention Indicator, which is electrically controlled by the chassis is present</p> <p>BIOS programs this field with a 1 for CEM/SIOM FFs.</p> |
| 2 | RO | 0h | <p>MRL Sensor Present</p> <p>This bit indicates that an MRL Sensor is implemented on the chassis for this slot.</p> <p>0: indicates that an MRL Sensor is not present 1: indicates that an MRL Sensor is present</p> <p>BIOS programs this field with a 0 for SIOM/Express cable and with either 0 or 1 for CEM depending on system design.</p> |
| 1 | RO | 0h | <p>Power Controller Present</p> <p>This bit indicates that a software controllable power controller is implemented on the chassis for this slot.</p> <p>0: indicates that a software controllable power controller is not present 1: indicates that a software controllable power controller is present</p> <p>BIOS programs this field with a 1 for CEM/SIOM FFs and a 0 for Express cable.</p> |
| 0 | RO | 0h | <p>Attention Button Present</p> <p>This bit indicates that the Attention Button event signal is routed (from slot or on-board in the chassis) to the IOH's hotplug controller.</p> <p>0: indicates that an Attention Button signal is routed to IOH 1: indicates that an Attention Button is not routed to IOH</p> |



21.12.4.22 SLTCON: PCI Express Slot Control Register

The Slot Control register identifies the PCI Express specific slot control parameters for operations such as Hot-plug and Power Management.

| Register: SLTCON Device: 1-10 Function: 0 Offset: A8h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:13 | RV | 0h | <i>Reserved.</i> |
| 12 | RO | 0 | Data Link Layer State Changed Enable – When set to 1, this field enables software notification when Data Link Layer Link Active field is changed. Note: This bit should be RO for ESI. However this bit is implemented as a RW bit and hence SW should not write to this bit. |
| 11 | RO | 0 | Electromechanical Interlock Control When software writes either a 1 to this bit, IOH pulses the EMIL pin per <i>PCI Express Server/Workstation Module Electromechanical Spec Rev 0.5a</i> . Write of 0 has no effect. This bit always returns a 0 when read. If electromechanical lock is not implemented, then either a write of 1 or 0 to this register has no effect. Note: This bit should be RO for ESI. However this bit is implemented as a RW bit and hence SW should not write to this bit. |
| 10 | RO | 1 | Power Controller Control If a power controller is implemented, when written sets the power state of the slot per the defined encodings. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not executed yet at the VPP, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. 1: Power Off 0: Power On Note: This bit should be RO for ESI. However this bit is implemented as a RW bit and hence SW should not write to this bit. |
| 9:8 | RO | 3h | Power Indicator Control If a Power Indicator is implemented, writes to this register set the Power Indicator to the written state. Reads of this field must reflect the value from the latest write, even if the corresponding hot-plug command is not executed yet at the VPP, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. 00: <i>Reserved.</i> 01: On 10: Blink (IOH drives 1.5 Hz square wave for Chassis mounted LEDs) 11: Off When this register is written, the event is signaled via the virtual pins ¹ of the IOH over a dedicated SMBus port. IOH does not generate the Power_Indicator_On/Off/Blink messages on PCI Express when this field is written to by software. Note: This bit should be RO for ESI. However this bit is implemented as a RW bit and hence SW should not write to this bit. |



| Register: SLTCON Device: 1-10 Function: 0 Offset: A8h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:6 | RO | 3h | Attention Indicator Control If an Attention Indicator is implemented, writes to this register set the Attention Indicator to the written state. Reads of this field reflect the value from the latest write, even if the corresponding hot-plug command is not executed yet at the VPP, unless software issues a write without waiting for the previous command to complete in which case the read value is undefined. 00: <i>Reserved.</i> 01: On 10: Blink (The IOH drives 1.5 Hz square wave) 11: Off When this register is written, the event is signaled via the virtual pins ¹ of the IOH over a dedicated SMBus port. IOH does not generated the Attention_Indicator_On/Off/Blink messages on PCI Express when this field is written to by software. Note: This bit should be RO for ESI. However this bit is implemented as a RW bit and hence SW should not write to this bit. |
| 5 | RO | 0h | Hot-plug Interrupt Enable When set to 1b, this bit enables generation of Hot-Plug MSI interrupt (and not wake event) on enabled Hot-Plug events, provided ACPI mode for hotplug is disabled. 0: disables interrupt generation on Hot-plug events 1: enables interrupt generation on Hot-plug events |
| 4 | RO | 0h | Command Completed Interrupt Enable This field enables the generation of Hot-plug interrupts (and not wake event) when a command is completed by the Hot-plug controller connected to the PCI Express port 0: disables hot-plug interrupts on a command completion by a hot-plug Controller 1: Enables hot-plug interrupts on a command completion by a hot-plug Controller Note: This bit should be RO for ESI. However this bit is implemented as a RW bit and hence SW should not write to this bit. |
| 3 | RO | 0h | Presence Detect Changed Enable This bit enables the generation of hot-plug interrupts or wake messages via a presence detect changed event. 0: disables generation of hot-plug interrupts or wake messages when a presence detect changed event happens. 1- Enables generation of hot-plug interrupts or wake messages when a presence detect changed event happens. |
| 2 | RO | 0h | MRL Sensor Changed Enable This bit enables the generation of hot-plug interrupts or wake messages via a MRL Sensor changed event. 0: disables generation of hot-plug interrupts or wake messages when an MRL Sensor changed event happens. 1: Enables generation of hot-plug interrupts or wake messages when an MRL Sensor changed event happens. Note: This bit should be RO for ESI. However this bit is implemented as a RW bit and hence SW should not write to this bit. |
| 1 | RO | 0h | Power Fault Detected Enable This bit enables the generation of hot-plug interrupts or wake messages via a power fault event. 0: disables generation of hot-plug interrupts or wake messages when a power fault event happens. 1: Enables generation of hot-plug interrupts or wake messages when a power fault event happens. Note: This bit should be RO for ESI. However this bit is implemented as a RW bit and hence SW should not write to this bit. |



| Register: SLTCON Device: 1-10 Function: 0 Offset: A8h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 0 | RO | 0h | Attention Button Pressed Enable This bit enables the generation of hot-plug interrupts or wake messages via an attention button pressed event. 0: disables generation of hot-plug interrupts or wake messages when the attention button is pressed. 1: Enables generation of hot-plug interrupts or wake messages when the attention button is pressed. Note: This bit should be RO for ESI. However this bit is implemented as a RW bit and hence SW should not write to this bit. |

Notes:

1. More information on Virtual pins can be found in [Section 16.8.2](#).

21.12.4.23 SLTSTS: PCI Express Slot Status Register (Dev #0 ESI mode)

The PCI Express Slot Status register defines important status information for operations such as Hot-plug and Power Management.

Note: Hot Plug is not supported on the ESI port so all bits are read only and default to zero.

| Register: SLTSTS (ESI Only) Device: 0 Function: 0 Offset: AAh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:9 | RV | 0h | <i>Reserved.</i> |
| 8 | RO | 0h | Data Link Layer State Changed This bit is set (if it is not already set) when the state of the Data Link Layer Link Active bit in the Link Status register changes. Software must read Data Link Layer Active field to determine the link state before initiating configuration cycles to the hot plugged device. |
| 7 | RO | 0h | Electromechanical Latch Status When read this register returns the current state of the Electromechanical Interlock (the EMILS pin) which has the defined encodings as: 0b Electromechanical Interlock Disengaged 1b Electromechanical Interlock Engaged |
| 6 | RO | 0h | Presence Detect State For ports with slots (where the Slot Implemented bit of the PCI Express Capabilities Registers is 1b), this field is the logical OR of the Presence Detect status determined via an in-band mechanism and sideband Present Detect pins. Refer to how PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*'s for how the inband presence detect mechanism works (certain states in the LTSSM constitute "card present" and others don't). 0: Card/Module/Cable slot empty or Cable Slot occupied but not powered 1: Card/module Present in slot (powered or unpowered) or cable present and powered on other end For ports with no slots, IOH hardwires this bit to 1b. Note: OS could get confused when it sees an empty PCI Express root port that is, "no slots + no presence", since this is now disallowed in the spec. So bios must hide all unused root ports devices in IOH config space, via the DEVHIDE register in Intel® QPI CSR space. |
| 5 | RO | 0h | MRL Sensor State This bit reports the status of an MRL sensor if it is implemented. 0: MRL Closed 1: MRL Open |



| Register: SLTSTS (ESI Only) Device: 0 Function: 0 Offset: AAh | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 4 | RO | 0h | Command Completed This bit is set by the IOH when the hot-plug command has completed and the hot-plug controller is ready to accept a subsequent command. It is subsequently cleared by software after the field has been read and processed. This bit provides no guarantee that the action corresponding to the command is complete. |
| 3 | RO | 0h | Presence Detect Changed This bit is set by the IOH when a Presence Detect Changed event is detected. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support out-of-band presence detect. |
| 2 | RO | 0h | MRL Sensor Changed This bit is set by the IOH when an MRL Sensor Changed event is detected. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support MRL. |
| 1 | RO | 0h | Power Fault Detected This bit is set by the IOH when a power fault event is detected by the power controller. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support power fault detection. |
| 0 | RO | 0h | Attention Button Pressed This bit is set by the IOH when the attention button is pressed. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support attention button. IOH silently discards the Attention_Button_Pressed message if received from PCI Express link without updating this bit. |

21.12.4.24 SLTSTS: PCI Express Slot Status Register (Dev #0 PCIe mode and Dev #1-10)

The PCI Express Slot Status register defines important status information for operations such as Hot-plug and Power Management.

| Register: SLTSTS Device: 0 (PCIe only) Device: 1-10 Function: 0 Offset: AAh | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:9 | RV | 0h | <i>Reserved.</i> |
| 8 | RW1C | 0h | Data Link Layer State Changed This bit is set (if it is not already set) when the state of the Data Link Layer Link Active bit in the Link Status register changes. Software must read Data Link Layer Active field to determine the link state before initiating configuration cycles to the hot plugged device. |



| Register: SLTSTS Device: 0 (PICe only) Device: 1-10 Function: 0 Offset: AAh | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 7 | RO | 0h | Electromechanical Latch Status When read this register returns the current state of the Electromechanical Interlock (the EMILS pin) which has the defined encodings as: 0b Electromechanical Interlock Disengaged 1b Electromechanical Interlock Engaged |
| 6 | RO | 0h | Presence Detect State For ports with slots (where the Slot Implemented bit of the PCI Express Capabilities Registers is 1b), this field is the logical OR of the Presence Detect status determined via an in-band mechanism and sideband Present Detect pins. Refer to how PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for how the inband presence detect mechanism works (certain states in the LTSSM constitute "card present" and others don't). 0: Card/Module/Cable slot empty or Cable Slot occupied but not powered 1: Card/module Present in slot (powered or unpowered) or cable present and powered on other end For ports with no slots, IOH hardwires this bit to 1b. Note: OS could get confused when it sees an empty PCI Express root port that is, "no slots + no presence", since this is now disallowed in the spec. So bios must hide all unused root ports devices in IOH config space, via the DEVHIDE register in Intel® QPI CSR space. |
| 5 | RO | 0h | MRL Sensor State This bit reports the status of an MRL sensor if it is implemented. 0: MRL Closed 1: MRL Open |
| 4 | RW1C | 0h | Command Completed This bit is set by the IOH when the hot-plug command has completed and the hot-plug controller is ready to accept a subsequent command. It is subsequently cleared by software after the field has been read and processed. This bit provides no guarantee that the action corresponding to the command is complete. |
| 3 | RW1C | 0h | Presence Detect Changed This bit is set by the IOH when a Presence Detect Changed event is detected. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support out-of-band presence detect. |
| 2 | RW1C | 0h | MRL Sensor Changed This bit is set by the IOH when an MRL Sensor Changed event is detected. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support MRL. |
| 1 | RW1C | 0h | Power Fault Detected This bit is set by the IOH when a power fault event is detected by the power controller. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support power fault detection. |
| 0 | RW1C | 0h | Attention Button Pressed This bit is set by the IOH when the attention button is pressed. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding this bit inactive if the FF/system does not support attention button. IOH silently discards the Attention_Button_Pressed message if received from PCI Express link without updating this bit. |



21.12.4.25 ROOTCON: PCI Express Root Control Register

The PCI Express Root Control register specifies parameters specific to the root complex port.

| Register: ROOTCON Device: 0-10 Function: 0 Offset: ACh | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:5 | RV | 0h | <i>Reserved.</i> |
| 4 | RW | 0h | CRS software visibility Enable This bit, when set, enables the Root Port to return Configuration Request Retry Status (CRS) Completion Status to software. If 0, retry status cannot be returned to software |
| 3 | RW | 0h | PME Interrupt Enable (Applies only to devices 0-7. This bit is a don't care for device 8) This field controls the generation of MSI interrupts for PME messages. 1: Enables interrupt generation upon receipt of a PME message 0: Disables interrupt generation for PME messages. |
| 2 | RW | 0h | System Error on Fatal Error Enable This field enables notifying the internal core error logic of occurrence of an uncorrectable fatal error at the port or below its hierarchy. The internal core error logic of IOH then decides if/how to escalate the error further (pins/message etc). Refer to Chapter 16 for details of how/which system notification is generated for a PCI Express/ESI fatal error. 1: indicates that a internal core error logic notification should be generated if a fatal error (ERR_FATAL) is reported by any of the devices in the hierarchy associated with and including this port. 0: No internal core error logic notification should be generated on a fatal error (ERR_FATAL) reported by any of the devices in the hierarchy associated with and including this port. Note that generation of system notification on a PCI Express/ESI fatal error is orthogonal to generation of an MSI interrupt for the same error. Both a system error and MSI can be generated on a fatal error or software can chose one of the two. Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for details of how this bit is used in conjunction with other error control bits to generate core logic notification of error events in a PCI Express/ESI port. |
| 1 | RW | 0h | System Error on Non-Fatal Error Enable This field enables notifying the internal core error logic of occurrence of an uncorrectable non-fatal error at the port or below its hierarchy. The internal core error logic of IOH then decides if/how to escalate the error further (pins/message etc). Refer to Chapter 16 for details of how/which system notification is generated for a PCI Express/ESI non-fatal error. 1: indicates that a internal core error logic notification should be generated if a non-fatal error (ERR_NONFATAL) is reported by any of the devices in the hierarchy associated with and including this port. 0: No internal core error logic notification should be generated on a non-fatal error (ERR_NONFATAL) reported by any of the devices in the hierarchy associated with and including this port. Note that generation of system notification on a PCI Express/ESI non-fatal error is orthogonal to generation of an MSI interrupt for the same error. Both a system error and MSI can be generated on a non-fatal error or software can chose one of the two. Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for details of how this bit is used in conjunction with other error control bits to generate core logic notification of error events in a PCI Express/ESI port. |



| Register: ROOTCON Device: 0-10 Function: 0 Offset: ACh | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 0 | RW | 0h | System Error on Correctable Error Enable This field controls notifying the internal core error logic of the occurrence of a correctable error in the device or below its hierarchy. The internal core error logic of IOH then decides if/how to escalate the error further (pins/ message etc). Refer to Chapter 16 for details of how/which system notification is generated for a PCI Express correctable error. 1: indicates that an internal core error logic notification should be generated if a correctable error (ERR_COR) is reported by any of the devices in the hierarchy associated with and including this port. 0: No internal core error logic notification should be generated on a correctable error (ERR_COR) reported by any of the devices in the hierarchy associated with and including this port. Note that generation of system notification on a PCI Express correctable error is orthogonal to generation of an MSI interrupt for the same error. Both a system error and MSI can be generated on a correctable error or software can chose one of the two. Refer to PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s for details of how this bit is used in conjunction with other error control bits to generate core logic notification of error events in a PCI Express/ESI port. |

21.12.4.26 ROOTCAP: PCI Express Root Capabilities Register

The PCI Express Root Status register specifies parameters specific to the root complex port.

| Register: ROOTCAP Device: 0-8; N/A for others Function: 0 Offset: AEh | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:1 | RV | 0h | <i>Reserved.</i> |
| 0 | RO | 1 | CRS Software Visibility This bit, when set, indicates that the Root Port is capable of returning Configuration Request Retry Status (CRS) Completion Status to software. IOH supports this capability. |

21.12.4.27 ROOTSTS: PCI Express Root Status Register

The PCI Express Root Status register specifies parameters specific to the root complex port.

| Register: ROOTSTS Device: 1-10 Function: 0 Offset: B0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:18 | RV | 0h | <i>Reserved.</i> |
| 17 | RO | 0h | PME Pending This field indicates that another PME is pending when the PME Status bit is set. When the PME Status bit is cleared by software; the pending PME is delivered by hardware by setting the PME Status bit again and updating the Requestor ID appropriately. The PME pending bit is cleared by hardware if no more PMEs are pending. |



| Register: ROOTSTS Device: 1-10 Function: 0 Offset: B0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 16 | RW1C | 0h | PME Status This field indicates a PM_PME message (either from the link or internally from within that root port) was received at the port. 1: PME was asserted by a requester as indicated by the PMEREQID field This bit is cleared by software by writing a '1'. Note that the root port itself could be the source of a PME event when a hotplug event is observed when the port is in D3hot state. |
| 15:0 | RO | 0000h | PME Requester ID This field indicates the PCI requester ID of the last PME requestor. If the root port itself was the source of the (virtual) PME message, then a RequesterID of IOHBUSNO:DevNo:0 is logged in this field. |

21.12.4.28 DEVCAP2: PCI Express Device Capabilities Register 2

| Register: DEVCAP2 Device: 0-10 Function: 0 Offset: B4h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:6 | RV | 0h | Reserved |
| 5 | RO | 1 | Alternative RID Interpretation (ARI) Capable: This bit is set to 1b indicating Root Port supports this capability |
| 4 | RO | 1 | Completion Timeout Disable Supported: IOH supports disabling Completion Timeout |
| 3:0 | RO | 1110b | Completion Time-Out Values Supported – This field indicates device support for the optional Completion Time-Out programmability mechanism. This mechanism allows system software to modify the Completion Time-Out range. Bits are one-hot encoded and set according to the table below to show time-out value ranges supported. A device that supports the optional capability of Completion Time-Out Programmability must set at least two bits. Ranges are defined: Range A: 50us to 10ms Range B: 10ms to 250ms Range C: 250ms to 4s Range D: 4s to 64s Bits are set according to table below to show timeout value ranges supported. 0000b: Completions Timeout programming not supported -- values is fixed by implementation in the range 50us to 50ms. 0001b: Range A 0010b: Range B 0011b: Range A & B 0110b: Range B & C 0111b: Range A, B, & C 1110b: Range B, C and D 1111b: Range A, B, C & D All other values are reserved. IOH supports time-out values from to 2ms-6s. The values specified above are required by PCIe Specification 2.1 |



21.12.4.29 DEVCTRL2: PCI Express Device Control Register 2

| Register: DEVCTRL2 Device: 0-10 Function: 0 Offset: B8h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:6 | RV | 0h | Reserved |
| 5 | RO | 0 | ARI Enable - this bit is hardwired to 0. |
| 4 | RW | 0 | Completion Time-Out Disable – When set to 1b, this bit disables the Completion Time-Out mechanism for all NP tx that IOH issues on the PCIe/ ESI link. When 0b, completion time-out is enabled. Software can change this field while there is active traffic in the root port. |
| 3:0 | RW | 0000b | Completion Time-Out Value on NP Tx that IOH issues on PCIe/ESI – In Devices that support Completion Time-Out programmability, this field allows system software to modify the Completion Time-Out range. The following encodings and corresponding time-out ranges are defined: 0000b = 2 ms 0001b = Reserved (IOH aliases to 0000b) 0010b = Reserved (IOH aliases to 0000b) 0101b = 4 ms 0110b = 10 ms 1001b = 40 ms 1010b = 210 ms 1101b = 800 ms 1110b = 2 s to 6.5 s Note: These values can deviate +/- 10% When OS selects the highest range, Section 21.13.8, “CTOCTRL: Completion Time-Out Control Register” on page 553 further controls the time-out value within that range. For all other ranges selected by OS, the time-out value within that range is fixed in IOH hardware. Software can change this field while there is active traffic in the root port. |

21.12.4.30 LNKCON2: PCI Express Link Control Register 2

| Register: LNKCON2 Device: 1-10 Function: 0 Offset: C0h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:13 | RV | 0 | Reserved |
| 12 | RWS | 0 | Compliance De-emphasis – This bit sets the de-emphasis level in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. Encodings: 1b -3.5 dB 0b -6 dB |
| 11 | RWS | 0 | Compliance SOS - When set to 1b, the LTSSM is required to send SKP Ordered Sets periodically in between the (modified) compliance patterns. |
| 10 | RWS | 0 | Enter Modified Compliance - When this bit is set to 1b, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.Compliance substate. |
| 9:7 | RWS | 0 | Transmit Margin - This field controls the value of the nondeemphasized voltage level at the Transmitter pins. |



| Register: LNKCON2 Device: 1-10 Function: 0 Offset: C0h | | | |
|---|------|-----------------|---|
| Bit | Attr | Default | Description |
| 6 | RWO | 0 | Selectable De-emphasis - When the Link is operating at 5.0 GT/s speed, this bit selects the level of de-emphasis for an Upstream component. Encodings: 1b -3.5 dB 0b -6 dB When the Link is operating at 2.5 GT/s speed, the setting of this bit has no effect. |
| 5 | RW | 0 | Hardware Autonomous Speed Disable - IOH does not change link speed autonomously other than for reliability reasons. |
| 4 | RWS | 0 | Enter Compliance - Software is permitted to force a link to enter Compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1b in both components on a link and then initiating a hot reset on the link. |
| 3:0 | RWS | See Description | Target Link Speed - This field sets an upper limit on link operational speed by restricting the values advertised by the upstream component in its training sequences. Defined encodings are: 0001b 2.5 Gb/s Target Link Speed 0010b 5 Gb/s Target Link Speed All other encodings are reserved. If a value is written to this field that does not correspond to a speed included in the Supported Link Speeds field, IOH will default to Gen1 speed. This field is also used to set the target compliance mode speed when software is using the Enter Compliance bit to force a link into compliance mode. |

21.12.4.31 PMCAP: Power Management Capabilities Register

The PM Capabilities Register defines the capability ID, next pointer and other power management related support. The following PM registers /capabilities are added for software compliance.

| Register: PMCAP Device: 1-10 Function: 0 Offset: E0h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:27 | RO | 11001 | PME Support Bits 31, 30 and 27 must be set to '1' for PCI-PCI bridge structures representing ports on root complexes. |
| 26 | RO | 0 | D2 Support IOH does not support power management state D2. |
| 25 | RO | 0 | D1 Support IOH does not support power management state D1. |
| 24:22 | RO | 0h | AUX Current |
| 21 | RO | 0 | Device Specific Initialization |
| 20 | RV | 0 | <i>Reserved.</i> |
| 19 | RO | 0 | PME Clock This field is hardwired to 0h as it does not apply to PCI Express. |



| Register: PMCAP Device: 1-10 Function: 0 Offset: E0h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 18:16 | RWO | 011 | Version This field is set to 3h (PM 1.2 compliant) as version number. The bit is RWO to make the version 2h incase legacy OS'es have any issues. |
| 15:8 | RO | 00h | Next Capability Pointer This is the last capability in the chain and hence set to 0. |
| 7:0 | RO | 01h | Capability ID Provides the PM capability ID assigned by PCI-SIG. |

21.12.4.32 PMCSR: Power Management Control and Status Register

This register provides status and control information for PM events in the PCI Express port of the IOH.

| Register: PMCSR Device: 1-10 Function: 0 Offset: E4h | | | |
|---|--------|---------|--|
| Bit | Attr | Default | Description |
| 31:24 | RO | 00h | Data Not relevant for IOH |
| 23 | RO | 0h | Bus Power/Clock Control Enable This field is hardwired to 0h as it does not apply to PCI Express. |
| 22 | RO | 0h | B2/B3 Support This field is hardwired to 0h as it does not apply to PCI Express. |
| 21:16 | RV | 0h | <i>Reserved.</i> |
| 15 | RW1CST | 0h | PME Status Applies only to root ports This PME Status is a sticky bit. This bit is set, independent of the PMEEN bit defined below, on an enabled PCI Express hotplug event provided the root port was in D3hot state. Software clears this bit by writing a '1' when it has been completed. Refer to <i>PCI Express Base Specification, Revision 1.1 and post 1.1 Erratas and EC*s</i> for further details on wake event generation at a root port. |
| 14:13 | RO | 0h | Data Scale Not relevant for IOH |
| 12:9 | RO | 0h | Data Select Not relevant for IOH |
| 8 | RWS | 0h | PME Enable Applies only to root ports. This field is a sticky bit and when set, enables PMEs generated internally on a PCI Express hotplug event to set the appropriate bits in the ROOTSTS register (which can then trigger an MSI or cause a _PMEGPE event). |
| 7:4 | RV | 0h | <i>Reserved.</i> |
| 3 | RWO | 1 | No Soft Reset Has no effect on functionality. |
| 2 | RV | 0h | <i>Reserved.</i> |



| Register: PMCSR Device: 1-10 Function: 0 Offset: E4h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 1:0 | RW | 0h | <p>Power State</p> <p>This 2-bit field is used to determine the current power state of the function and to set a new power state as well.</p> <p>00: D0 01: D1 (not supported by IOH) 10: D2 (not supported by IOH) 11: D3_hot</p> <p>Software should only write values supported in PMCAP (00 & 11). If Software tries to write 01 or 10 to this field, the power state does not change from the existing power state (which is either D0 or D3hot) and nor do these bits1:0 change value.</p> <p>All devices will respond to only Type 0 configuration transactions when in D3hot state (root port will not forward Type 1 accesses to the downstream link) and will not respond to memory/IO transactions (i.e. D3hot state is equivalent to MSE/IOSE bits being clear) as target and will not generate any memory/IO/configuration transactions as initiator on the primary bus (messages are still allowed to pass through).</p> <p>When the rootport is in D3hot state it will block accesses to IOAPIC memory range. That is when rootport is in D3hot OS will not be able to access downstream IOAPIC.</p> <p>Only functionality this field is connected to is in dropping outbound transactions if this power state field indicates D3_hot. This option is through CSR PXPCTRL & by default disabled for D3_hot. If drop option is set for D3-hot, if in D3_hot then next write cannot be anything other than D0.</p> |



21.12.5 PCI Express Enhanced Configuration Space

21.12.5.1 ERRCAPHDR: PCI Express Enhanced Capability Header

This register identifies the capability structure and points to the next structure.

| Register: ERRCAPHDR Device: 0-10 Function: 0 Offset: 100h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:20 | RO | 150h | Next Capability Offset This field points to the next Capability in extended configuration space. |
| 19:16 | RO | 1h | Capability Version Set to 1h for this version of the PCI Express logic |
| 15:0 | RO | 0001h | PCI Express Extended CAP_ID Assigned for advanced error reporting |

21.12.5.2 UNCERRSTS: Uncorrectable Error Status

This register identifies uncorrectable errors detected for PCI Express/ESI port.

| Register: UNCERRSTS Device: 0-10 Function: 0 Offset: 104h | | | |
|--|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:22 | RV | 0h | <i>Reserved</i> |
| 21 | RW1CS | 0 | ACS Violation Status |
| 20 | RW1CS | 0 | Received an Unsupported Request |
| 19 | RV | 0 | <i>Reserved</i> |
| 18 | RW1CS | 0 | Malformed TLP Status |
| 17 | RW1CS | 0 | Receiver Buffer Overflow Status |
| 16 | RW1CS | 0 | Unexpected Completion Status |
| 15 | RW1CS | 0 | Completer Abort Status |
| 14 | RW1CS | 0 | Completion Time-out Status |
| 13 | RW1CS | 0 | Flow Control Protocol Error Status |
| 12 | RW1CS | 0 | Poisoned TLP Status |
| 11:6 | RV | 0h | <i>Reserved</i> |
| 5 | RW1CS | 0 | Surprise Down Error Status <i>Note:</i> For non hot-plug removals, this will be logged only when SLTCON[10] is set to 0. |
| 4 | RW1CS | 0 | Data Link Protocol Error Status |
| 3:0 | RV | 0h | <i>Reserved</i> |



21.12.5.3 UNCERRMSK: Uncorrectable Error Mask

This register masks uncorrectable errors from being signaled.

| Register: UNCERRMSK Device: 0-10 Function: 0 Offset: 108h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:21 | RV | 0 | Reserved |
| 21 | RWS | 0 | ACS Violation Mask |
| 20 | RWS | 0 | Unsupported Request Error Mask |
| 19 | RV | 0 | Reserved |
| 18 | RWS | 0 | Malformed TLP Mask |
| 17 | RWS | 0 | Receiver Buffer Overflow Mask |
| 16 | RWS | 0 | Unexpected Completion Mask |
| 15 | RWS | 0 | Completer Abort Mask |
| 14 | RWS | 0 | Completion Time-out Mask |
| 13 | RWS | 0 | Flow Control Protocol Error Mask |
| 12 | RWS | 0 | Poisoned TLP Mask |
| 11:6 | RV | 0 | Reserved |
| 5 | RWS | 0 | Surprise Down Error Mask |
| 4 | RWS | 0 | Data Link Layer Protocol Error Mask |
| 3:0 | RV | 0 | Reserved |

21.12.5.4 UNCERRSEV: Uncorrectable Error Severity

This register indicates the severity of the uncorrectable errors.

| Register: UNCERRSEV Device: 0-10 Function: 0 Offset: 10Ch | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:22 | RV | 0 | <i>Reserved</i> |
| 21 | RWS | 0 | ACS Violation Severity |
| 20 | RWST | 0 | Unsupported Request Error Severity |
| 19 | RV | 0 | <i>Reserved</i> |
| 18 | RWS | 1 | Malformed TLP Severity |
| 17 | RWS | 1 | Receiver Buffer Overflow Severity |
| 16 | RWS | 0 | Unexpected Completion Severity |
| 15 | RWS | 0 | Completer Abort Severity |
| 14 | RWS | 0 | Completion Time-out Severity |
| 13 | RWS | 1 | Flow Control Protocol Error Severity |
| 12 | RWS | 0 | Poisoned TLP Severity |
| 11:6 | RV | 0 | <i>Reserved</i> |
| 5 | RWS | 1 | Surprise Down Error Severity |



| Register: UNCERRSEV Device: 0-10 Function: 0 Offset: 10Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 4 | RWS | 1 | Data Link Protocol Error Severity |
| 3:0 | RV | 0 | Reserved |

21.12.5.5 CORERRSTS: Correctable Error Status

This register identifies the status of the correctable errors that have been detected by the Express port.

| Register: CORERRSTS Device: 0-10 Function: 0 Offset: 110h | | | |
|--|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:14 | RV | 0h | Reserved |
| 13 | RW1CS | 0 | Advisory Non-fatal Error Status Note: inbound memory writes within address range of 2^{51} to $2^{52}-1$ are considered Advisory non-fatal instead of non-fatal. If severity of UR is set to Non-Fatal and if Advisory reporting is enabled then if a 64-bit Memory write with address in range of 0x8_0000_0000_0000 to 0xF_FFFF_FFFF_FFFF is encountered, then it is logged as Advisory non-Fatal error. i.e. CORERRSTS[13] is set, instead of just non-fatal. Memory writes above 2^{52} (upto 2^{63}) are correctly logged as non-Fatal under similar conditions. No issues if UR severity is set to Fatal or Advisory reporting is not enabled. No issues for Memory reads in the same address range. |
| 12 | RW1CS | 0 | Replay Timer Time-out Status |
| 11:9 | RV | 0 | Reserved |
| 8 | RW1CS | 0 | Replay_Num Rollover Status |
| 7 | RW1CS | 0 | Bad DLLP Status |
| 6 | RW1CS | 0 | Bad TLP Status |
| 5:1 | RV | 0 | Reserved |
| 0 | RW1CS | 0 | Receiver Error Status |

21.12.5.6 CORERRMSK: Correctable Error Mask

This register masks correctable errors from being not signalled.

| Register: CORERRMSK Device: 0-10 Function: 0 Offset: 114h | | | |
|--|------|---------|--------------------------------------|
| Bit | Attr | Default | Description |
| 31:14 | RV | 0h | Reserved |
| 13 | RWS | 1h | Advisory Non-fatal Error Mask |
| 12 | RWS | 0h | Replay Timer Time-out Mask |
| 11:9 | RV | 0h | Reserved |
| 8 | RWS | 0h | Replay_Num Rollover Mask |



| Register: CORERRMSK Device: 0-10 Function: 0 Offset: 114h | | | |
|--|------|---------|---------------------|
| Bit | Attr | Default | Description |
| 7 | RWS | 0h | Bad DLLP Mask |
| 6 | RWS | 0 | Bad TLP Mask |
| 5:1 | RV | 0 | Reserved |
| 0 | RWS | 0 | Receiver Error Mask |

21.12.5.7 ERRCAP: Advanced Error Capabilities and Control Register

| Register: ERRCAP Device: 0-10 Function: 0 Offset: 118h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:9 | RV | 0 | Reserved |
| 8 | RO | 0 | ECRC Check Enable: N/A to IOH |
| 7 | RO | 0 | ECRC Check Capable: N/A to IOH |
| 6 | RO | 0 | ECRC Generation Enable: N/A to IOH |
| 5 | RO | 0 | ECRC Generation Capable: N/A to IOH |
| 4:0 | ROS | 0h | First error pointer The First Error Pointer is a read-only register that identifies the bit position of the first unmasked error reported in the Uncorrectable Error register. In case of two errors happening at the same time, fatal error gets precedence over non-fatal, in terms of being reported as first error. This field is rearmed to capture new errors when the status bit indicated by this field is cleared by software. |

21.12.5.8 HDRLOG: Header Log

This register contains the header log when the first error occurs. Headers of the subsequent errors are not logged.

| Register: HDRLOG Device: 0-10 Function: 0 Offset: 11Ch | | | |
|---|------|---------|-------------------------------------|
| Bit | Attr | Default | Description |
| 127:0 | ROS | 0h | Header of TLP associated with error |



21.12.5.9 RPERRCMD: Root Port Error Command Register

This register controls behavior upon detection of errors. Refer to [Section 16.6.3.5, “PCI Express Error Reporting Specifics”](#) for details of MSI generation for PCIe error events.

| Register: RPERRCMD Device: 0-10 Function: 0 Offset: 12Ch | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:3 | RV | 0 | <i>Reserved</i> |
| 2 | RW | 0 | FATAL Error Reporting Enable Enable MSI interrupt on fatal errors when set. |
| 1 | RW | 0 | Non-FATAL Error Reporting Enable Enable interrupt on a non-fatal error when set. |
| 0 | RW | 0 | Correctable Error Reporting Enable Enable interrupt on correctable error when it set. |

21.12.5.10 RPERRSTS: Root Error Status Register

The Root Error Status register reports status of error Messages (ERR_COR), ERR_NONFATAL, and ERR_FATAL) received by the Root Complex in IOH, and errors detected by the Root Port itself (which are treated conceptually as if the Root Port had sent an error Message to itself). The ERR_NONFATAL and ERR_FATAL Messages are grouped together as uncorrectable. Each correctable and uncorrectable (non-fatal and fatal) error source has a first error bit and a next error bit associated with it respectively. When an error is received by a Root Complex, the respective first error bit is set and the Requestor ID is logged in the Error Source Identification register. A set individual error status bit indicates that a particular error category occurred; software may clear an error status by writing a 1 to the respective bit. If software does not clear the first reported error before another error Message is received of the same category (correctable or uncorrectable), the corresponding next error status bit will be set but the Requestor ID of the subsequent error Message is discarded. The next error status bits may be cleared by software by writing a 1 to the respective bit as well.

| Register: RPERRSTS Device: 0-10 Function: 0 Offset: 130h | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:27 | RO | 0h | Advanced Error Interrupt Message Number Advanced Error Interrupt Message Number offset between base message data and the MSI message if assigned more than one message number. IOH hardware automatically updates this register to 0x1h if the number of messages allocated to the root port is 2. See bit 6:4 for details of the number of messages allocated to a root port. |
| 26:7 | RV | 0 | <i>Reserved</i> |
| 6 | RW1CS | 0 | Fatal Error Messages Received Set when one or more Fatal Uncorrectable error Messages have been received. |
| 5 | RW1CS | 0 | Non-Fatal Error Messages Received Set when one or more Non-Fatal Uncorrectable error Messages have been received. |



| Register: RPERRSTS Device: 0-10 Function: 0 Offset: 130h | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 4 | RW1CS | 0 | First Uncorrectable Fatal Set when bit 2 is set (from being clear) and the message causing bit 2 to be set is an ERR_FATAL message. |
| 3 | RW1CS | 0 | Multiple Error Fatal/Nonfatal Received Set when either a fatal or a non-fatal error message is received and Error Fatal/Nonfatal Received is already set, that is, log from the 2nd Fatal or No fatal error message onwards |
| 2 | RW1CS | 0 | Error Fatal/Nonfatal Received Set when either a fatal or a non-fatal error message is received and this bit is already not set, that is, log the first error message. Note that when this bit is set bit 3 could be either set or clear. |
| 1 | RW1CS | 0 | Multiple Correctable Error Received Set when either a correctable error message is received and Correctable Error Received bit is already set, that is, log from the 2nd Correctable error message onwards |
| 0 | RW1CS | 0 | Correctable Error Received Set when a correctable error message is received and this bit is already not set, that is, log the first error message |

21.12.5.11 ERRSID: Error Source Identification Register

While internally generated error messages in the IOH will have their Requester ID logged correctly in this register, incoming ERR_* messages' Requester ID will not be.

| Register: ERRSID Device: 0-10 Function: 0 Offset: 134h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:16 | ROS | 0h | Fatal Non Fatal Error Source ID Requestor ID of the source when a Fatal or Non Fatal error message is received and the Error Fatal/Nonfatal Received bit is not already set, that is, log ID of the first Fatal or Non Fatal error message. Note that when the root port itself is the cause of the received message (virtual message), then a Source ID of IOHBUSNO:DevNo:0 is logged into this register. |
| 15:0 | ROS | 0h | Correctable Error Source ID Requestor ID of the source when a correctable error message is received and the Correctable Error Received bit is not already set, that is, log ID of the first correctable error message. Note that when the root port itself is the cause of the received message (virtual message), then a Source ID of IOHBUSNO:DevNo:0 is logged into this register. |



21.12.5.12 SSMSK: Stop and Scream Mask Register

This register masks uncorrectable errors from being signaled as Stop and Scream events. Whenever the uncorrectable status bit is set and stop and scream mask is not set for that bit, it will trigger a Stop and Scream event.

| Register: SSMSK Device: 0-10 Function: 0 Offset: 138 | | | |
|---|------|---------|-------------------------------------|
| Bit | Attr | Default | Description |
| 31:22 | RV | 0 | Reserved |
| 21 | RWS | 0 | ACS Violation Mask |
| 20 | RWS | 0 | Unsupported Request Error Mask |
| 19 | RV | 0 | Reserved |
| 18 | RWS | 0 | Malformed TLP Mask |
| 17 | RWS | 0 | Receiver Buffer Overflow Mask |
| 16 | RWS | 0 | Unexpected Completion Mask |
| 15 | RWS | 0 | Completer Abort Mask |
| 14 | RWS | 0 | Completion Time-out Mask |
| 13 | RWS | 0 | Flow Control Protocol Error Mask |
| 12 | RWS | 0 | Poisoned TLP Mask |
| 11:6 | RV | 0 | Reserved |
| 5 | RWS | 0 | Surprise Down Error Mask |
| 4 | RWS | 0 | Data Link Layer Protocol Error Mask |
| 3:0 | RV | 0 | Reserved |

21.12.5.13 APICBASE: APIC Base Register

| Register: APICBASE Device: 0-10 Function: 0 Offset: 140h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:12 | RV | 0h | Reserved |
| 11:1 | RW | 0h | Bits 19:9 of the APIC base Bits 31:20 are assumed to be 0xFECh. Bits 8:0 are a don't care for address decode. Address decoding to the APIC range is done as APIC_BASE[31:8] <= A[31:8] <= APIC_LIMIT[31:8]. |
| 0 | RW | 0h | APIC range enable: enables the decode of the APIC window |



21.12.5.14 APICLIMIT: APIC Limit Register

| Register: APICLIMIT Device: 0-10 Function: 0 Offset: 142h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 15:12 | RV | 0h | Reserved |
| 11:1 | RW | 0h | Bits 19:9 of the APIC limit Bits 31:20 are assumed to be 0xFECh. Bits 8:0 are a don't care for address decode. Address decoding to the APIC range is done as APIC_BASE[31:8] <= A[31:8] <= APIC_LIMIT[31:8]. |
| 0 | RV | 0h | Reserved |

21.12.5.15 ACSCAPHDR: Access Control Services Extended Capability Header

This register identifies the Access Control Services (ACS) capability structure and points to the next structure.

| Register: ACSCAPHDR Device: 0-10 Function: 0 Offset: 150h | | | |
|--|------|---|---|
| Bit | Attr | Default | Description |
| 31:20 | RO | Dev: def 0: 160h 1: 160h 3: 160h 7: 160h else: 00h | Next Capability Offset This field points to the next Capability configuration space. This is set to 160h for Dev# 0, 1, 3 and 7. For other PCI Express devices this is set to 00h indicating that this is the last capability. |
| 19:16 | RO | 1h | Capability Version Set to 1h for this version of the PCI Express logic |
| 15:0 | RO | 000Dh | PCI Express Extended CAP_ID Assigned for Access Control Services capabilities |

21.12.5.16 ACSCAP: Access Control Services Capability Register

This register identifies the Access Control Services (ACS) capabilities.

| Register: ACSCAP Device: 0-10 Function: 0 Offset: 154h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:8 | RO | 00h | Egress Control Vector Size Indicates the number of bits in the Egress Control Vector. This is set to 00h as ACS P2P Egress Control (E) bit in this register is 0b. |
| 7 | RV | 0 | Reserved. |
| 6 | RO | 0 | ACS Direct Translated P2P (T) Indicates that the component does not implement ACS Direct Translated P2P. |
| 5 | RO | 0 | ACS P2P Egress Control (E) Indicates that the component does not implement ACS P2P Egress Control. |
| 4 | RO | 1 | ACS Upstream Forwarding (U) Indicates that the component implements ACS Upstream Forwarding. |



| Register: ACSCAP Device: 0-10 Function: 0 Offset: 154h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 3 | RO | 1 | ACS P2P Completion Redirect (C) Indicates that the component implements ACS P2P Completion Redirect. |
| 2 | RO | 1 | ACS P2P Request Redirect (R) Indicates that the component implements ACS P2P Request Redirect. |
| 1 | RO | 1 | ACS Translation Blocking (B) Indicates that the component implements ACS Translation Blocking. |
| 0 | RO | 1 | ACS Source Validation (V) Indicates that the component implements ACS Source Validation. |

21.12.5.17 ACCTRL: Access Control Services Control Register

This register identifies the Access Control Services (ACS) control bits.

| Register: ACCTRL Device: 0-10 Function: 0 Offset: 156h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 15:7 | RV | 0 | Reserved. |
| 6 | RO | 0 | ACS Direct Translated P2P Enable (T) This is hardwired to 0b as the component does not implement ACS Direct Translated P2P. |
| 5 | RW | 0 | ACS P2P Egress Control Enable (E) This is hardwired to 0b as the component does not implement ACS P2P Egress Control. |
| 4 | RW | 0 | ACS Upstream Forwarding Enable (U) When set, the component forwards upstream any Request or Completion TLPs it receives that were redirected upstream by a component lower in the hierarchy. Note that the U bit only applies to upstream TLPs arriving at a Downstream Port, and whose normal routing targets the same Downstream Port. |
| 3 | RW | 0 | ACS P2P Completion Redirect Enable (C) Determines when the component redirects peer-to-peer Completions upstream; applicable only to Read Completions whose Relaxed Ordering Attribute is clear. |
| 2 | RW | 0 | ACS P2P Request Redirect Enable (R) This bit determines when the component redirects peer-to-peer Requests upstream. |
| 1 | RW | 0 | ACS Translation Blocking Enable (B) When set, the component blocks all upstream Memory Requests whose Address Translation (AT) field is not set to the default value. |
| 0 | RW | 0 | ACS Source Validation Enable (V) When set, the component validates the Bus Number from the Requester ID of upstream Requests against the secondary / subordinate Bus Numbers. |



21.12.5.18 PERFCTRLSTS: Performance Control and Status Register

| Register: PERFCTRLSTS Device: 0-10 Function: 0 Offset: 180h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 63:41 | RV | 0 | Reserved |
| 40 | RW | 0 | DCA Requester ID override: When this CSR bit is set, it indicates that there is no match for DCA Req ID authentication and eventually disables DCA all together so no prefetch hints will be sent. This is independent of how the tag field is programmed. |
| 39:21 | RV | 0 | Reserved |
| 20:16 | RW | 0x18 | Number of outstanding RFOs/pre-allocated non-posted requests for PCI Express Gen1 This register controls the number of outstanding inbound non-posted requests - i/o, config, memory - that a Gen1 PCI Express downstream port can have, for all non-posted requests (peer-to-peer or to main-memory) it pre-allocates buffer space for. This register provides the value for the port when it is operating in Gen1 mode and for a link width of x4. The value of this parameter for the port when operating in Gen1 x8 width is obtained by multiplying this register by 2 and 4 respectively. Software programs this register based on the read/RFO latency to main memory. This register also specifies the number of RFOs that can be kept outstanding on Intel® QPI from a given port. The link speed of the port can change during a PCI Express hotplug event and the port must use this register or the Gen2 register (see bits 13:8) based on the link speed. A value of 1 indicates one outstanding pre-allocated request, 2 indicates 2 outstanding pre-allocated requests and so on. Software can change this register at runtime (in preparation for Intel® QPI quiesce) and hardware should be tolerant of that. |
| 15:14 | RV | 0 | Reserved |
| 13:8 | RW | 0x30 | Number of outstanding pre-allocated non-posted requests for PCI Express Gen2 This register controls the number of outstanding inbound non-posted requests - i/o, config, memory - (maximum length of these requests is a CL) that a Gen2 PCI Express downstream port can have, for all non-posted requests (peer-to-peer or to main-memory) it pre-allocates buffer space for. This register provides the value for the port when it is operating in Gen2 mode and for a link width of x4. The value of this parameter for the port when operating in Gen2 width is obtained by multiplying this register by 2 and 4 respectively. Software programs this register based on the read/RFO latency to main memory. This register also specifies the number of RFOs that can be kept outstanding on Intel® QPI for a given port. The link speed of the port can change during a PCI Express hotplug event and the port must use this register or the Gen1 register (see bits 20:16) based on the link speed. A value of 1 indicates one outstanding pre-allocated request, 2 indicates 2 outstanding pre-allocated requests and so on. Software can change this register at runtime (in preparation for Intel® QPI quiesce) and hardware should be tolerant of that. |
| 7:4 | RV | 0 | Reserved |
| 3 | RW | 0 | Enable No-Snoop Optimization on Writes When set, memory reads with NS=1 will not be snooped on Intel QuickPath Interconnect. |
| 2 | RW | 0h | Enable No-Snoop Optimization on Reads When set, memory reads with NS=1 will not be snooped on Intel QuickPath Interconnect. |
| 1 | RV | 0h | Reserved |



| Register: PERFCTRLSTS Device: 0-10 Function: 0 Offset: 180h | | | |
|--|------|---------|-------------|
| Bit | Attr | Default | Description |
| 0 | RV | 1 | Reserved |

21.12.5.19 MISCCTRLSTS: Misc. Control and Status Register (Dev #0, ESI Only)

| Register: MISCCTRLSTS Device: 0 (ESI Only) Function: 0 Offset: 188h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 63:49 | RV | 0 | Reserved |
| 48 | RW1C | 0 | Received PME_TO_ACK: Indicates that IOH received a PME turn off ack packet or it timed out waiting for the packet |
| 47:37 | RV | 0 | Reserved |
| 36 | RWS | 0 | Form-Factor Indicates what form-factor a particular root port controls 0 - CEM/Cable 1 - SIOM This bit is used to interpret bit 6 in the VPP serial stream for the port as either MRL# (CEM/Cable) input or EMLSTS# (SIOM) input. In case of cable form factor. |
| 35 | RWST | 0 | Override System Error on PCIe Fatal Error Enable: When set, fatal errors on PCI Express (that have been successfully propagated to the primary interface of the port) are sent to the IOH core error logic (for further escalation) regardless of the setting of the equivalent bit in the ROOTCTRL register. When clear, the fatal errors are only propagated to the IOH core error logic if the equivalent bit in ROOTCTRL register is set. |
| 34 | RWST | 0 | Override System Error on PCIe Non-fatal Error Enable: When set, non-fatal errors on PCI Express (that have been successfully propagated to the primary interface of the port) are sent to the IOH core error logic (for further escalation) regardless of the setting of the equivalent bit in the ROOTCTRL register. When clear, the non-fatal errors are only propagated to the IOH core error logic if the equivalent bit in ROOTCTRL register is set. |
| 33 | RWST | 0 | Override System Error on PCIe Correctable Error Enable: When set, correctable errors on PCI Express (that have been successfully propagated to the primary interface of the port) are sent to the IOH core error logic (for further escalation) regardless of the setting of the equivalent bit in the ROOTCTRL register. When clear, the correctable errors are only propagated to the IOH core error logic if the equivalent bit in ROOTCTRL register is set. |
| 32 | RW | 0 | ACPI PME Interrupt Enable: When set, Assert/Deassert_PMEGPE messages are enabled to be generated when ACPI mode is enabled for handling PME messages from PCI Express. When this bit is cleared (from a 1), a Deassert_PMEGPE message is scheduled on behalf of the root port if an Assert_PMEGPE message was sent earlier from the root port. |
| 31 | RWST | 0 | Disable L0s on transmitter: When set, IOH never puts its tx in L0s state, even if OS enables it via the Link Control register. |
| 30 | RWO | 0 | Inbound I/O disable: When set, all inbound I/O are aborted and treated as UR. |
| 29 | RW | 1 | cfg_to_en 1: Disables 0: Enables config timeouts, independently of other timeouts. |



| Register: MISCCTRLSTS Device: 0 (ESI Only) Function: 0 Offset: 188h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 28 | RWST | 0 | Disables Timeouts Completely when set. |
| 27 | RWS | 0 | System Interrupt Only on Link BW/Management Status This bit, when set, will disable generating MSI interrupt on link bandwidth (speed and/or width) and management changes, even if MSI is enabled, that is, will disable generating MSI when LNKSTS bits 15 and 14 are set. Whether or not this condition results in a system event like SMI/PMI/CPEI is dependent is masked or not in the XPCORERRMSK register. |
| 26 | RW | 0 | Disable EOI broadcast to this PCIe link: When set, EOI message will not be broadcast down this PCIe link. When clear, the port is a valid target for EOI broadcast. |
| 25:22 | RV | 0 | Reserved |
| 21 | RV | 1 | Reserved |
| 20:13 | RV | 0 | Reserved |
| 12:11 | RV | 1 | Reserved |
| 10:9 | RV | 0 | Reserved |
| 8:7 | RW | 0 | PME_TO_ACK Timeout Control: |
| 6 | RWST | 0 | Enable timeout for receiving PME_TO_ACK: When set, IOH enables the timeout to receiving the PME_TO_ACK |
| 5 | RW | 0 | Send PME_TURN_OFF Message When this bit is written with a 1b, IOH sends a PME_TURN_OFF message to the PCIe link. Hardware clears this bit when the message has been sent on the link. |
| 4 | RW | 0 | Enable System Error only for AER When this bit is set, the PCI Express errors do not trigger an MSI interrupt, regardless of the whether MSI is enabled or not. Whether or not PCI Express errors result in a system event like NMI/SMI/PMI/CPEI is dependent on whether the appropriate system error enable bits are set or not. See Section 16.6.3, "PCI Express Error Reporting Mechanism" on page 229 for details of how this bit interacts with other control bits in signalling errors to the IOH global error reporting logic. When this bit is clear, PCI Express errors are reported via MSI and/or NMI/SMI/MCA/CPEI. When this bit is clear and if MSI enable bit in the Section 21.12.5.19, "MISCCTRLSTS: Misc. Control and Status Register (Dev #0, ESI Only)" on page 535 is set, then an MSI interrupt is generated for PCI Express errors. When this bit is clear, and 'System Error on Fatal Error Enable' bit in Section 21.12.4.25, "ROOTCON: PCI Express Root Control Register" on page 518 is set, then NMI/SMI/MCA is (also) generated for a PCI Express fatal error. Similar behavior for non-fatal and corrected errors. |



| Register: MISCCTRLSTS Device: 0 (ESI Only) Function: 0 Offset: 188h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 3 | RW | 0 | <p>Enable ACPI mode for Hotplug</p> <p>When this bit is set, all HP events from the PCI Express port are handled via _HPGPE messages to the ICH and no MSI messages are ever generated for HP events (regardless of whether MSI is enabled at the root port or not) at the root port. When this bit is clear, _HPGPE message generation on behalf of root port HP events is disabled and OS can chose to generate MSI interrupt for HP events, by setting the MSI enable bit in root ports. This bit does not apply to the ESI ports.</p> <p>Clearing this bit (from being 1) schedules a Deassert_HPGPE event on behalf of the root port, provided there was any previous Assert_HPGPE message that was sent without an associated Deassert message.</p> |
| 2 | RW | 0 | <p>Enable ACPI mode for PM</p> <p>When this bit is set, all PM events at the PCI Express port are handled via _PMEGPE messages to the ICH, and no MSI interrupts are ever generated for PM events at the root port (regardless of whether MSI is enabled at the root port or not). When clear, _PMEGPE message generation for PM events is disabled and OS can chose to generate MSI interrupts for delivering PM events by setting the MSI enable bit in root ports. This bit does not apply to the ESI ports.</p> <p>Clearing this bit (from being 1) schedules a Deassert_PMEGPE event on behalf of the root port, provided there was any previous Assert_PMEGPE message that was sent without an associated Deassert message.</p> |
| 1 | RWO | 0h | <p>Inbound Configuration enable</p> <p>When clear all inbound configuration transactions are sent a UR response by the receiving PCI Express port. When set, inbound configs are allowed.</p> |
| 0 | RO | 1 | <p>Set Host Bridge Class code</p> <p>When this bit is set, the class code register indicates "Host Bridge".</p> |

21.12.5.20 MISCCTRLSTS: Misc. Control and Status Register (Dev #0, PCIe Only)

| Register: MISCCTRLSTS Device: 0 (PCIe Only) Function: 0 Offset: 188h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 63:49 | RV | 0 | Reserved |
| 48 | RW1C | 0 | <p>Received PME_TO_ACK:</p> <p>Indicates that IOH received a PME turn off ack packet or it timed out waiting for the packet</p> |
| 47:37 | RV | 0 | Reserved |
| 36 | RWS | 0 | <p>Form-Factor</p> <p>Indicates what form-factor a particular root port controls</p> <p>0 - CEM/Cable 1 - SIOM</p> <p>This bit is used to interpret bit 6 in the VPP serial stream for the port as either MRL# (CEM/Cable) input or EMLSTS# (SIOM) input. In case of cable form factor.</p> |
| 35 | RWST | 0 | <p>Override System Error on PCIe Fatal Error Enable: When set, fatal errors on PCI Express (that have been successfully propagated to the primary interface of the port) are sent to the IOH core error logic (for further escalation) regardless of the setting of the equivalent bit in the ROOTCTRL register. When clear, the fatal errors are only propagated to the IOH core error logic if the equivalent bit in ROOTCTRL register is set.</p> |



| Register: MISCCTRLSTS Device: 0 (PCIe Only) Function: 0 Offset: 188h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 34 | RWST | 0 | Override System Error on PCIe Non-fatal Error Enable: When set, non-fatal errors on PCI Express (that have been successfully propagated to the primary interface of the port) are sent to the IOH core error logic (for further escalation) regardless of the setting of the equivalent bit in the ROOTCTRL register. When clear, the non-fatal errors are only propagated to the IOH core error logic if the equivalent bit in ROOTCTRL register is set. |
| 33 | RWST | 0 | Override System Error on PCIe Correctable Error Enable: When set, correctable errors on PCI Express (that have been successfully propagated to the primary interface of the port) are sent to the IOH core error logic (for further escalation) regardless of the setting of the equivalent bit in the ROOTCTRL register. When clear, the correctable errors are only propagated to the IOH core error logic if the equivalent bit in ROOTCTRL register is set. |
| 32 | RW | 0 | ACPI PME Interrupt Enable: When set, Assert/Deassert_PMEGPE messages are enabled to be generated when ACPI mode is enabled for handling PME messages from PCI Express. When this bit is cleared (from a 1), a Deassert_PMEGPE message is scheduled on behalf of the root port if an Assert_PMEGPE message was sent earlier from the root port. |
| 31 | RWST | 0 | Disable L0s on transmitter: When set, IOH never puts its tx in L0s state, even if OS enables it via the Link Control register. |
| 30 | RWO | 0 | Inbound I/O disable: When set, all inbound I/O are aborted and treated as UR. |
| 29 | RW | 1 | cfg_to_en 1: Disables 0: Enables config timeouts, independently of other timeouts. |
| 28 | RWST | 0 | Disables Timeouts Completely when set. |
| 27 | RWS | 0 | System Interrupt Only on Link BW/Management Status This bit, when set, will disable generating MSI interrupt on link bandwidth (speed and/or width) and management changes, even if MSI is enabled, that is, will disable generating MSI when LNKSTS bits 15 and 14 are set. Whether or not this condition results in a system event like SMI/PMI/CPEI is dependent is masked or not in the XPCORERRMSK register. |
| 26 | RW | 0 | Disable EOI broadcast to this PCIe link: When set, EOI message will not be broadcast down this PCIe link. When clear, the port is a valid target for EOI broadcast. |
| 25:22 | RV | 0 | Reserved |
| 21 | RV | 1 | Reserved |
| 20:13 | RV | 0 | Reserved |
| 12:11 | RV | 1 | Reserved |
| 10:9 | RV | 0 | Reserved |
| 8:7 | RW | 0 | PME_TO_ACK Timeout Control: |
| 6 | RWST | 0 | Enable timeout for receiving PME_TO_ACK: When set, IOH enables the timeout to receiving the PME_TO_ACK |
| 5 | RW | 0 | Send PME_TURN_OFF Message When this bit is written with a 1b, IOH sends a PME_TURN_OFF message to the PCIe link. Hardware clears this bit when the message has been sent on the link. |



| Register: MISCCTRLSTS Device: 0 (PCIe Only) Function: 0 Offset: 188h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 4 | RW | 0 | <p>Enable System Error only for AER</p> <p>When this bit is set, the PCI Express errors do not trigger an MSI interrupt, regardless of the whether MSI is enabled or not. Whether or not PCI Express errors result in a system event like NMI/SMI/PMI/CPEI is dependent on whether the appropriate system error enable bits are set or not.</p> <p>See Section 16.6.3, "PCI Express Error Reporting Mechanism" on page 229 for details of how this bit interacts with other control bits in signalling errors to the IOH global error reporting logic.</p> <p>When this bit is clear, PCI Express errors are reported via MSI and/or NMI/SMI/MCA/CPEI. When this bit is clear and if MSI enable bit in the Section 21.12.5.19, "MISCCTRLSTS: Misc. Control and Status Register (Dev #0, ESI Only)" on page 535 is set, then an MSI interrupt is generated for PCI Express errors. When this bit is clear, and 'System Error on Fatal Error Enable' bit in Section 21.12.4.25, "ROOTCON: PCI Express Root Control Register" on page 518 is set, then NMI/SMI/MCA is (also) generated for a PCI Express fatal error. Similar behavior for non-fatal and corrected errors.</p> |
| 3 | RW | 0 | <p>Enable ACPI mode for Hotplug</p> <p>When this bit is set, all HP events from the PCI Express port are handled via _HPGPE messages to the ICH and no MSI messages are ever generated for HP events (regardless of whether MSI is enabled at the root port or not) at the root port. When this bit is clear, _HPGPE message generation on behalf of root port HP events is disabled and OS can chose to generate MSI interrupt for HP events, by setting the MSI enable bit in root ports. This bit does not apply to the ESI ports.</p> <p>Clearing this bit (from being 1) schedules a Deassert_HPGPE event on behalf of the root port, provided there was any previous Assert_HPGPE message that was sent without an associated Deassert message.</p> |
| 2 | RW | 0 | <p>Enable ACPI mode for PM</p> <p>When this bit is set, all PM events at the PCI Express port are handled via _PMEGPE messages to the ICH, and no MSI interrupts are ever generated for PM events at the root port (regardless of whether MSI is enabled at the root port or not). When clear, _PMEGPE message generation for PM events is disabled and OS can chose to generate MSI interrupts for delivering PM events by setting the MSI enable bit in root ports. This bit does not apply to the ESI ports.</p> <p>Clearing this bit (from being 1) schedules a Deassert_PMEGPE event on behalf of the root port, provided there was any previous Assert_PMEGPE message that was sent without an associated Deassert message.</p> |
| 1 | RWO | 0h | <p>Inbound Configuration enable</p> <p>When clear all inbound configuration transactions are sent a UR response by the receiving PCI Express port. When set, inbound configs are allowed.</p> |
| 0 | RO | 1 | <p>Set Host Bridge Class code</p> <p>When this bit is set, the class code register indicates "Host Bridge".</p> |



21.12.5.21 MISCCTRLSTS: Misc. Control and Status Register (Dev #1-10, PCIe Only)

| Register: MISCCTRLSTS Device: 1-10 (PCIe Only) Function: 0 Offset: 188h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 63:49 | RV | 0 | Reserved |
| 48 | RW1C | 0 | Received PME_TO_ACK: Indicates that IOH received a PME turn off ack packet or it timed out waiting for the packet |
| 47:37 | RV | 0 | Reserved |
| 36 | RWS | 0 | Form-Factor Indicates what form-factor a particular root port controls 0 - CEM/Cable 1 - SIOM This bit is used to interpret bit 6 in the VPP serial stream for the port as either MRL# (CEM/Cable) input or EMLSTS# (SIOM) input. In case of cable form factor. |
| 35 | RWST | 0 | Override System Error on PCIe Fatal Error Enable: When set, fatal errors on PCI Express (that have been successfully propagated to the primary interface of the port) are sent to the IOH core error logic (for further escalation) regardless of the setting of the equivalent bit in the ROOTCTRL register. When clear, the fatal errors are only propagated to the IOH core error logic if the equivalent bit in ROOTCTRL register is set. |
| 34 | RWST | 0 | Override System Error on PCIe Non-fatal Error Enable: When set, non-fatal errors on PCI Express (that have been successfully propagated to the primary interface of the port) are sent to the IOH core error logic (for further escalation) regardless of the setting of the equivalent bit in the ROOTCTRL register. When clear, the non-fatal errors are only propagated to the IOH core error logic if the equivalent bit in ROOTCTRL register is set. |
| 33 | RWST | 0 | Override System Error on PCIe Correctable Error Enable: When set, correctable errors on PCI Express (that have been successfully propagated to the primary interface of the port) are sent to the IOH core error logic (for further escalation) regardless of the setting of the equivalent bit in the ROOTCTRL register. When clear, the correctable errors are only propagated to the IOH core error logic if the equivalent bit in ROOTCTRL register is set. |
| 32 | RW | 0 | ACPI PME Interrupt Enable: When set, Assert/Deassert_PMEGPE messages are enabled to be generated when ACPI mode is enabled for handling PME messages from PCI Express. When this bit is cleared (from a 1), a Deassert_PMEGPE message is scheduled on behalf of the root port if an Assert_PMEGPE message was sent earlier from the root port. |
| 31 | RWST | 0 | Disable L0s on transmitter: When set, IOH never puts its tx in L0s state, even if OS enables it via the Link Control register. |
| 30 | RW | 0 | Inbound I/O disable: When set, all inbound I/O are aborted and treated as UR. |
| 29 | RWST | 1 | cfg_to_en Disables/enables config timeouts, independently of other timeouts. |
| 28 | RWST | 0 | to_dis Disables timeouts completely. |



| Register: MISCCTRLSTS Device: 1-10 (PCIe Only) Function: 0 Offset: 188h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 27 | RWST | 0 | System Interrupt Only on Link BW/Management Status This bit, when set, will disable generating MSI interrupt on link bandwidth (speed and/or width) and management changes, even if MSI is enabled, that is, will disable generating MSI when LNKSTS bits 15 and 14 are set. Whether or not this condition results in a system event like SMI/PMI/CPEI is dependent is masked or not in the XPCORERMSK register. |
| 26 | RW | 0 | Disable EOI broadcast to this PCIe link: When set, EOI message will not be broadcast down this PCIe link. When clear, the port is a valid target for EOI broadcast. |
| 25 | RWST | 0 | Peer2peer Memory Write Disable: When set, peer2peer memory writes are master aborted otherwise they are allowed to progress per the peer2peer decoding rules. |
| 24 | RW | 0 | Peer2peer Memory Read Disable: When set, peer2peer memory reads are master aborted otherwise they are allowed to progress per the peer2peer decoding rules. |
| 25:13 | RV | 0 | Reserved |
| 12 | RV | 1 | Reserved |
| 11:9 | RV | 0 | Reserved |
| 8:7 | RW | 0 | PME2ACKTOCTRL |
| 6 | RWST | 0 | Enable timeout for receiving PME_TO_ACK: When set, IOH enables the timeout to receiving the PME_TO_ACK |
| 5 | RW | 0 | Send PME_TURN_OFF message When this bit is written with a 1b, IOH sends a PME_TURN_OFF message to the PCIe link. Hardware clears this bit when the message has been sent on the link. |
| 4 | RW | 0 | Enable System Error only for AER When this bit is set, the PCI Express errors do not trigger an MSI interrupt, regardless of the whether MSI is enabled or not. Whether or not PCI Express errors result in a system event like NMI/SMI/PMI/CPEI is dependent on whether the appropriate system error enable bits are set or not. See Section 16.6.3, "PCI Express Error Reporting Mechanism" on page 229 for details of how this bit interacts with other control bits in signalling errors to the IOH global error reporting logic. When this bit is clear, PCI Express errors are reported via MSI and/or NMI/SMI/MCA/CPEI. When this bit is clear and if MSI enable bit in the Section 21.12.5.19, "MISCCTRLSTS: Misc. Control and Status Register (Dev #0, ESI Only)" on page 535 is set, then an MSI interrupt is generated for PCI Express errors. When this bit is clear, and 'System Error on Fatal Error Enable' bit in Section 21.12.4.25, "ROOTCON: PCI Express Root Control Register" on page 518 is set, then NMI/SMI/MCA is (also) generated for a PCI Express fatal error. Similar behavior for non-fatal and corrected errors. |



| Register: MISCCTRLSTS Device: 1-10 (PCIe Only) Function: 0 Offset: 188h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 3 | RW | 0 | Enable ACPI mode for Hotplug When this bit is set, all HP events from the PCI Express port are handled via _HPGPE messages to the ICH and no MSI messages are ever generated for HP events (regardless of whether MSI is enabled at the root port or not) at the root port. When this bit is clear, _HPGPE message generation on behalf of root port HP events is disabled and OS can chose to generate MSI interrupt for HP events, by setting the MSI enable bit in root ports. This bit does not apply to the ESI ports. Clearing this bit (from being 1) schedules a Deassert_HPGPE event on behalf of the root port, provided there was any previous Assert_HPGPE message that was sent without an associated Deassert message. |
| 2 | RW | 0 | Enable ACPI mode for PM When this bit is set, all PM events at the PCI Express port are handled via _PMEGPE messages to the ICH, and no MSI interrupts are ever generated for PM events at the root port (regardless of whether MSI is enabled at the root port or not). When clear, _PMEGPE message generation for PM events is disabled and OS can chose to generate MSI interrupts for delivering PM events by setting the MSI enable bit in root ports. This bit does not apply to the ESI ports. Clearing this bit (from being 1) schedules a Deassert_PMEGPE event on behalf of the root port, provided there was any previous Assert_PMEGPE message that was sent without an associated Deassert message. |
| 1 | RWO | 0h | Inbound Configuration enable When clear all inbound configuration transactions are sent a UR response by the receiving PCI Express port. When set, inbound configs are allowed. |
| 0 | RW | 0 | Set Host Bridge Class code When this bit is set, the class code register indicates "Host Bridge". |

21.12.5.22 PCIE_IOU0_BIF_CTRL: PCIe IO Unit (IOU)0 Bifurcation Control Register

This control register holds bifurcation control information pertaining to the PCI Express IO Unit 0.

| Register: PCIE_IOU0_BIF_CTRL Device: 3 Function: 0 Offset: 190h | | | |
|--|------|---------|-----------------|
| Bit | Attr | Default | Description |
| 15:4 | RV | 0h | <i>Reserved</i> |



| Register: PCIE_IOU0_BIF_CTRL Device: 3 Function: 0 Offset: 190h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 3 | WO | 0 | <p>IOU0 Start Bifurcation: When software writes a 1 to this bit, IOH starts the port 0 bifurcation process. After writing to this bit, software can poll the Data Link Layer link active bit in the LNKSTS register to determine if a port is up and running. Once a port bifurcation has been initiated by writing a 1 to this bit, software cannot initiate any more write-1 to this bit (write of 0 is ok).</p> <p>Note that this bit can be written to a 1 in the same write that changes values for bits 2:0 in this register and in that case, the new value from the write to bits 2:0 take effect.</p> <p>This bit always reads a 0b.</p> |
| 2:0 | RWS | 000b | <p>IOU0 Bifurcation Control:</p> <p>111 Reserved 110 ¹/ Reserved 101 Reserved 100 x16 011 x8x8 (15:8 operate as x8, 7:0 operate as x8) 010 x8x4x4 (15:8 operate as x8, 7:4 operate as x4 and 3:0 operate as x4) 001 x4x4x8 (15:12 operate as x4, 11:8 operate as x4 and 7:0 operate as x8) 000 x4x4x4x4 (15:12 operate as x4, 11:8 operate as x4, 7:4 operate as x4 and 3:0 operate as x4)</p> <p>To select a port 0 bifurcation, software sets this field and then either a) sets bit 3 in this register to initiate training OR b) resets the entire IOH and on exit from that reset, IOH will bifurcate the ports per the setting in this field.</p> <p>Refer to for further description of the port bifurcation under bios feature.</p> |



21.12.5.23 PCIE_IOU1_BIF_CTRL: PCIe IO Unit (IOU)1 Bifurcation Control Register

This control register holds bifurcation control information pertaining to the PCI Express IO Unit 1.

| Register: PCIE_IOU1_BIF_CTRL Device: 7 Function: 0 Offset: 190h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 15:4 | RV | 0 | <i>Reserved</i> |
| 3 | WO | 0 | IOU1 Start Bifurcation: When software writes a 1 to this bit, IOH starts the port 1 bifurcation process. After writing to this bit, software can poll the Data Link Layer link active bit in the LNKSTS register to determine if a port is up and running. Once a port bifurcation has been initiated by writing a 1 to this bit, software cannot initiate any more write-1 to this bit (write of 0 is ok). Note that this bit can be written to a 1 in the same write that changes values for bits 2:0 in this register and in that case, the new value from the write to bits 2:0 take effect. This bit always reads a 0b. |
| 2:0 | RWS | 000b | IOU1 Bifurcation Control: 111 Use strap settings to determine port bifurcation (default) 110 ¹ / Reserved 101 Reserved 100 x16 011 x8x8 (15:8 operate as x8, 7:0 operate as x8) 010 x8x4x4 (15:8 operate as x8, 7:4 operate as x4 and 3:0 operate as x4) 001 x4x4x8 (15:12 operate as x4, 11:8 operate as x4 and 7:0 operate as x8) 000 x4x4x4x4 (15:12 operate as x4, 11:8 operate as x4, 7:4 operate as x4 and 3:0 operate as x4) To select a port 1 bifurcation, software sets this field and then either a) sets bit 3 in this register to initiate training OR b) resets the entire IOH and on exit from that reset, IOH will bifurcate the ports per the setting in this field. Refer to for further description of the port bifurcation under bios feature. |

21.12.5.24 PCIE_IOU2_BIF_CTRL: PCIe IO Unit (IOU)2 Bifurcation Control Register

This control register holds bifurcation control information pertaining to the PCI Express IO Unit 2.

| Register: PCIE_IOU2_BIF_CTRL Device: 1 Function: 0 Offset: 190h | | | |
|--|------|---------|-----------------|
| Bit | Attr | Default | Description |
| 15:4 | RV | 0h | <i>Reserved</i> |



| Register: PCIE_IOU2_BIF_CTRL Device: 1 Function: 0 Offset: 190h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 3 | WO | 0h | <p>IOU2 Start Bifurcation: When software writes a 1 to this bit, IOH starts the port 2 bifurcation process. After writing to this bit, software can poll the Data Link Layer link active bit in the LNKSTS register to determine if a port is up and running. Once a port bifurcation has been initiated by writing a 1 to this bit, software cannot initiate any more write-1 to this bit (write of 0 is ok).</p> <p>Note that this bit can be written to a 1 in the same write that changes values for bits 1:0 in this register and in that case, the new value from the write to bits 1:0 take effect.</p> <p>This bit always reads a 0b.</p> |
| 2 | RV | 0h | <i>Reserved</i> |
| 1:0 | RWS | 0b | <p>IOU2 Bifurcation Control:</p> <p>111 Reserved 110 Reserved 101 Reserved 100 Reserved 011 Reserved 010 Reserved 001 x4 000 x2x2 (3:2 operate as x2, 1:0 operate as x2)</p> <p>To select a port 2 bifurcation, software sets this field and then either a) sets bit 3 in this register to initiate training OR b) resets the entire IOH and on exit from that reset, IOH will bifurcate the ports per the setting in this field. Refer to further description of the port bifurcation under bios feature.</p> |

21.12.6 XP Common Block Link Control Registers

21.12.6.1 XP[7,3,1,0]DLLCTRL: XP DLL Control Register

| Register: XP[7,3,1,0]DLLCTRL Device: 14 Function: 0 Offset: 4C0h Device: 13 Function: 3 Offset: 4C0h Device: 13 Function: 1 Offset: 4C0h Device: 13 Function: 0 Offset: 4C0h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:24 | RV | 0h | <i>Reserved</i> |
| 23:20 | RWS | 3h | <p>ack_period: This specifies the period after which a packet will be acked. The timer starts running once it is internally received. For X16, the default value is 3h. For X8 or X4, the number is scaled by 2 from the X16 number; for X2, the number is scaled by 4 from the X16 number; and for x1, the number is scaled by 8 from the X16 number. Note: A value of 0 is illegal.</p> |
| 19:14 | RWS | 3h | <p>fc_p_period: This specifies the period after which the flow control packet will be scheduled. This starts from the time at which the credit release is received by the link layer. For X16, the default number is 3h. For X8 or X4, the number is scaled by 2 from the X16 number; for X2, the number is scaled by 4 from the X16 number; and for x1, the number is scaled by 8 from the X16 number. Note: A value of 0 is illegal.</p> |



| Register: XP[7,3,1,0]DLLCTRL Device: 14 Function: 0 Offset: 4C0h Device: 13 Function: 3 Offset: 4C0h Device: 13 Function: 1 Offset: 4C0h Device: 13 Function: 0 Offset: 4C0h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 14:10 | RWS | 3h | fc_np_period: This specifies the period after which the flow control packet will be scheduled. This starts from the time at which the credit release is received by the link layer. For X16, the default number is 3h. For X8 or X4, the number is scaled by 2 from the X16 number; for X2, the number is scaled by 4 from the X16 number; and for x1, the number is scaled by 8 from the X16 number. Note: A value of 0 is illegal. |
| 9:5 | RWS | 3h | fc_c_period: This specifies the period after which the flow control packet will be scheduled. This starts from the time at which the credit release is received by the link layer. For X16, the default number is 3h. For X8 or X4, the number is scaled by 2 from the X16 number; for X2, the number is scaled by 4 from the X16 number; and for x1, the number is scaled by 8 from the X16 number. Note: A value of 0 is illegal. |
| 4:0 | RWS | 1Ch | idle_fc_period: This is the time in which flow control updates are scheduled, if there are no new credits to be received. If LOs is not disabled, this specifies the number of microseconds after which the flow control packet will be sent. For ports where LOs is disabled, the timer value is calculated by multiplying this number by 256ns. Note: A value of 0 is illegal. |

21.12.6.2 XP[7,3,1,0]RETRYCTRL: RetryControl Register

| Register: XP[7,3,1,0]RETRYCTRL Device: 14 Function: 0 Offset: 4C4h Device: 13 Function: 3 Offset: 4C4h Device: 13 Function: 1 Offset: 4C4h Device: 13 Function: 0 Offset: 4C4h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:17 | RV | 0h | <i>Reserved</i> |
| 16:14 | RWS | 4h | reinit_threshold: Specifies the number of retry attempts after which a recovery will be initiated. Note: A value of 0 is an illegal. |
| 13:6 | RWS | 0 | unuseintrntrybuf: The number of unusable entries in the retry buffer. There are 220 entries. Setting it very high may cause the link to not send any TLP's. |
| 5:0 | RWS | 3h | replay_timer: Specifies the time in microseconds after which a replay will be initiated. Setting it to 0 will disable the timer. The timer accuracy is plus or minus 1us |



21.13 IOH Defined PCI Express Error Registers

The contents of the next set of registers - XPCORERRSTS, XPCORERRMSK, XPUNCERRSTS, XPUNCERRMSK, XPUNCERRSEV, XPUNCERRPTR. The architecture model for error logging and escalation of internal errors is similar to that of PCI Express AER, except that these internal errors never trigger an MSI and are always reported to the system software. Mask bits mask the reporting of an error and severity bit controls escalation to either fatal or non-fatal error to the internal core error logic. Note that internal errors detected in the PCI Express cluster are not dependent on any other control bits for error escalation other than the mask bit defined in these registers. All these registers are sticky. Refer to [Figure 16-6](#).

21.13.1 XPCORERRSTS - XP Correctable Error Status Register

| Register: XPCORERRSTS Device: 0-10 Function: 0 Offset: 200h | | | |
|--|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:1 | RV | 0 | Reserved |
| 0 | RW1CS | 0 | PCI link bandwidth changed status Note: This bit is implemented as an OR of LNKSTS[15] (LABS) and LNKSTS[14] (LBMS). The mask for XPCORERRSTS[0] is set to 1 by default. Thus in order to log bandwidth changes, LNKSTS[15:14], XPCORERRSTS[0] and XPCOREDMASK[0] need to be cleared. Once the xpcorerrsts[0] is unmasked and then set to 1 due to a bandwidth change LNKSTS[15:14] need to be cleared before clearing XPCORERRSTS[0]. |

21.13.2 XPCORERRMSK - XP Correctable Error Mask Register

| Register: XPCORERRMSK Device: 0-10 Function: 0 Offset: 204h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:1 | RV | 0 | Reserved |
| 0 | RWS | 0 | PCI link bandwidth Changed mask This bit is only functional when set for Device's 0, 3, and 7. Device 4 gets its mask value from Device 6, and Device 8 gets its mask value from Device 10. |



21.13.3 XPUNCERRSTS - XP Uncorrectable Error Status Register

| Register: XPUNCERRSTS Device: 0-10 Function: 0 Offset: 208h | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:10 | RV | 0 | Reserved |
| 9 | RWICS | 0 | Outbound Poisoned Data: Set when outbound poisoned data (from QPI or peer, write or read completion) is received by this port |
| 8 | RW1CS | 0 | Received MSI writes greater than a DWORD data |
| 7 | RV | 0 | Reserved |
| 6 | RW1CS | 0 | Received PCIe completion with UR status |
| 5 | RW1CS | 0 | Received PCIe completion with CA status |
| 4 | RW1CS | 0 | Sent completion with Unsupported Request |
| 3 | RW1CS | 0 | Sent completion with completion Abort |
| 2 | RV | 0 | Reserved |
| 1 | RW1CS | 0 | Outbound Switch FIFO data parity error detected |
| 0 | RV | 0 | Reserved |

21.13.4 XPUNCERRMSK - XP Uncorrectable Error Mask Register

| Register: XPUNCERRMSK Device: 0-10 Function: 0 Offset: 20Ch | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:10 | RV | 0 | Reserved |
| 9 | RWS | 0 | Outbound Poisoned Data Mask: Masks signaling of stop and scream condition to the core error logic |
| 8 | RWS | 0 | Received MSI writes greater than a DWORD data mask |
| 7 | RV | 0 | Reserved |
| 6 | RWS | 0 | Received PCIe completion with UR status mask |
| 5 | RWS | 0 | Received PCIe completion with CA status mask |
| 4 | RWS | 0 | Sent completion with Unsupported Request mask |
| 3 | RWS | 0 | Sent completion with Completer Abort mask |
| 2 | RV | 0 | Reserved |
| 1 | RWS | 0 | Outbound Switch FIFO data parity error detected mask |
| 0 | RV | 0 | Reserved |



21.13.5 XPUNCERRSEV - XP Uncorrectable Error Severity Register

| Register: XPUNCERRSEV Device: 0-10 Function: 0 Offset: 210h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:10 | RV | 0 | Reserved |
| 9 | RWS | 0 | Outbound Poisoned Data Severity |
| 8 | RWS | 0 | Received MSI writes greater than a DWORD data severity |
| 7 | RV | 1 | Reserved |
| 6 | RWS | 0 | Received PCIe completion with UR status severity |
| 5 | RWS | 0 | Received PCIe completion with CA status severity |
| 4 | RWS | 0 | Sent completion with Unsupported Request severity |
| 3 | RWS | 1 | Sent completion with Completer Abort severity |
| 2 | RV | 0 | Reserved |
| 1 | RWS | 1 | Outbound Switch FIFO data parity error detected severity |
| 0 | RV | 1 | Reserved |

21.13.5.1 XPUNCERRPTR - XP Uncorrectable Error Pointer Register

| Register: XPUNCERRPTR Device: 0-10 Function: 0 Offset: 214h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 7:5 | RV | 0 | Reserved |
| 4:0 | ROS | 0 | XP Uncorrectable First Error Pointer - This field points to which of the unmasked uncorrectable errors happened first. This field is only valid when the corresponding error is unmasked and the status bit is set and this field is rearmed to load again when the status bit indicated to by this pointer is cleared by software from 1 to 0. Value of 0x0 corresponds to bit 0 in XPUNCERRSTS register, value of 0x1 corresponds to bit 1, and so on. |



21.13.5.2 UNCEDMASK: Uncorrectable Error Detect Status Mask

This register masks uncorrectable errors from causing the associated AER status bit to be set.

| Register: UNCEDMASK Device: 0-10 Function: 0 Offset: 218h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:22 | RV | 0h | Reserved |
| 21 | RWST | 1 | ACS Violation Detect Mask |
| 20 | RWST | 1 | Received an Unsupported Request Detect Mask |
| 19 | RV | 0 | Reserved |
| 18 | RWST | 1 | Malformed TLP Detect Mask |
| 17 | RWST | 1 | Receiver Buffer Overflow Detect Mask |
| 16 | RWST | 1 | Unexpected Completion Detect Mask |
| 15 | RWST | 1 | Completer Abort Detect Mask |
| 14 | RWST | 1 | Completion Time-out Detect Mask |
| 13 | RWST | 1 | Flow Control Protocol Error Detect Mask |
| 12 | RWST | 1 | Poisoned TLP Detect Mask |
| 11:6 | RV | 0h | Reserved |
| 5 | RWST | 1 | Surprise Down Error Detect Mask |
| 4 | RWST | 1 | Data Link Layer Protocol Error Detect Mask |
| 3:0 | RV | 0 | Reserved |

21.13.5.3 COREDMASK: Correctable Error Detect Status Mask

This register masks correctable errors from causing the associated status bit in AER status register to be set.

| Register: COREDMASK Device: 0-10 Function: 0 Offset: 21Ch | | | |
|--|------|---------|--------------------------------------|
| Bit | Attr | Default | Description |
| 31:14 | RV | 0 | Reserved |
| 13 | RWST | 1 | Advisory Non-fatal Error Detect Mask |
| 12 | RWST | 1 | Replay Timer Time-out Detect Mask |
| 11:9 | RV | 0 | Reserved |
| 8 | RWST | 1 | Replay_Num Rollover Detect Mask |
| 7 | RWST | 1 | Bad DLLP Detect Mask |
| 6 | RWST | 1 | Bad TLP Detect Mask |
| 5:1 | RV | 0 | Reserved |
| 0 | RWST | 1 | Receiver Error Detect Mask |



21.13.5.4 RPEDMASK - Root Port Error Detect Status Mask

This register masks the associated error messages (received from PCIe link and NOT the virtual ones generated internally), from causing the associated status bits in AER to be set.

| Register: RPEDMASK Device: 0-10 Function: 0 Offset: 220h | | | |
|---|------|---------|--------------------------------------|
| Bit | Attr | Default | Description |
| 31:3 | RV | 0 | Reserved |
| 2 | RWST | 1 | Fatal error detect Detect mask |
| 1 | RWST | 1 | Non-Ffatal error detect Detect mask |
| 0 | RWST | 1 | Correctable error detect status mask |

21.13.5.5 XPUNCEDMASK - XP Uncorrectable Error Detect Mask Register

| Register: XPUNCEDMASK Device: 0-10 Function: 0 Offset: 224h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 31:10 | RV | 0 | Reserved |
| 9 | RWS | 1 | Outbound Poisoned Data Detect Mask |
| 8 | RWS | 1 | Received MSI writes greater than a DWORD data Detect Mask |
| 7 | RV | 0 | Reserved |
| 6 | RWS | 1 | Received PCIe completion with UR Detect Mask |
| 5 | RWS | 1 | Received PCIe completion with CA Detect Mask |
| 4 | RWS | 1 | Sent completion with Unsupported Request Detect Mask |
| 3 | RWS | 1 | Sent completion with Completer Abort Detect Mask |
| 2 | RV | 0 | Reserved |
| 1 | RWS | 1 | Outbound Switch FIFO data parity error Detect Mask |
| 0 | RV | 0 | Reserved |

21.13.5.6 XPCOREDMASK - XP Correctable Error Detect Mask Register

| Register: XPCOREDMASK Device: 0-10 Function: 0 Offset: 228h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:1 | RV | 0 | Reserved |
| 0 | RWS | 1 | PCI link bandwidth changed Detect Mask |



21.13.6 XPGLBERRSTS - XP Global Error Status Register

This register captures if an error is logged in any of two buckets of errors within XP, XP internal core logic, and PCI Express AER.

| Register: XPGLBERRSTS Device: 0-10 Function: 0 Offset: 230h | | | |
|--|-------|---------|--|
| Bit | Attr | Default | Description |
| 15:3 | RV | 0 | Reserved |
| 2 | RW1CS | 0 | <p>PCIe AER Correctable Error - A PCIe correctable error (either internally detected by IOH or a FATAL message received) was detected anew. Note that if that error was masked in the PCIe AER, it is not reported in this field. Software clears this bit by writing a 1 and at that stage, only "subsequent" PCIe unmasked correctable errors will set this bit.</p> <p>Conceptually, per the flow of PCI Express Base Spec 1.1 defined Error message control, this bit is set by the ERR_COR message that is enabled to cause a System Error notification.</p> <p>Refer to section titled PCI Express Error Reporting Specifics in the RAS chapter in IOH RAS for details of how this bit interacts with other control/status bits in signalling errors to the IOH global error reporting logic.</p> |
| 1 | RW1CS | 0 | <p>PCIe AER Non-fatal Error - A PCIe non-fatal error (either internally detected by IOH or a FATAL message received) was detected anew. Note that if that error was masked in the PCIe AER, it is not reported in this field. Software clears this bit by writing a 1 and at that stage only "subsequent" PCIe unmasked non-fatal errors will set this bit again.</p> <p>Refer to section titled PCI Express Error Reporting Specifics in the RAS chapter in IOH RAS for details of how this bit interacts with other control/status bits in signalling errors to the IOH global error reporting logic.</p> |
| 0 | RW1CS | 0 | <p>PCIe AER Fatal Error - A PCIe fatal error (either internally detected by IOH or a FATAL message received) was detected anew. Note that if that error was masked in the PCIe AER, it is not reported in this field. Software clears this bit by writing a 1 and at that stage, only "subsequent" PCIe unmasked fatal errors will set this bit.</p> <p>Refer to section titled PCI Express Error Reporting Specifics in the RAS chapter in IOH RAS for details of how this bit interacts with other control/status bits in signalling errors to the IOH global error reporting logic.</p> |

21.13.7 XPGLBERRPTR - XP Global Error Pointer Register

This register captures if an error is logged in any of three buckets of errors within XP - XP internal and PCI Express AER.

| Register: XPGLBERRPTR Device: 0-7,9 Function: 0 Offset: 232h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 15:3 | RV | 0 | Reserved |
| 2:0 | ROS | 0 | <p>XP Cluster Global First Error Pointer - This field points to which of the 3 errors indicated in the XPGLBERRSTS register happened first. This field is only valid when the corresponding status bit is set and this field is rearmed to load again when the status bit indicated to by this pointer is cleared by software from 1 to 0.</p> <p>Value of 0x0 corresponds to bit 0 in XPGLBERRSTS register, value of 0x1 corresponds to bit 1, and so on.</p> |



21.13.8 CTOCTRL: Completion Time-Out Control Register

| Register: CTOCTRL Device: 0-10 Function: 0 Offset: 1E0h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:10 | RV | 0 | Reserved |
| 9:8 | RW | 0 | XP-to-PCIe time-out select within 2s to 6.5s range: When OS selects a time-out range of 2s to 6.5s for XP (that affect NP tx issued to the PCIe/ESI) using the root port's DEVCON2 register, this field selects the sub-range within that larger range, for additional controllability. 00: 2s 01: 4s 10: 6.5s 11: Reserved Note: These values may vary by +/- 10% |
| 7:0 | RV | 0 | Reserved |

21.13.9 PCIE_LER_SS_CTRLSTS: PCI Express Live Error Recovery/Stop and Scream Control and Status Register

| Register: PCIE_LER_SS_CTRLSTS Device: 0-10 Function: 0 Offset: 1E4h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31 | RO | 0 | LER_SS Status Indicates that an error was detected which caused the PCIe port to go into a live error recovery mode. This bit remains set till all the associated unmasked status bits are cleared. |
| 30:8 | RV | 0 | Reserved |
| 7 | RWS | 0 | XPUNCERRSTS_Received PCIe Completion With UR Status Mask When clear, when the Received PCIe Completion With URstatus bit in the XPUNCERRSTS register is 1b, the LER_SS Status bit in this register will be set. When set, the register Received PCIe Completion With URstatus bit in the XPUNCERRSTS has no impact on the LER_SS Status bit in this register. |
| 6 | RWS | 1 | XPUNCERRSTS_Received PCIe Completion With CA Status Mask When clear, when the Received PCIe Completion With CAstatus bit in the XPUNCERRSTS register is 1b, the LER_SS Status bit in this register will be set. When set, the register Received PCIe Completion With CAstatus bit in the XPUNCERRSTS has no impact on the LER_SS Status bit in this register. |
| 5 | RWS | 0 | XPUNCERRSTS_Stop and Scream Status Mask When clear, when the Outbound Poisoned Data status bit in the XPUNCERRSTS register is 1b, the LER_SS Status bit in this register will be set. When set, the Outbound Poisoned Data status bit in the XPUNCERRSTS register has no impact on the LER_SS Status bit in this register |
| 4 | RV | 0 | Reserved |



| Register: PCIE_LER_SS_CTRLSTS Device: 0-10 Function: 0 Offset: 1E4h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 3 | RWS | 0 | ERRSTS_Fatal Error Messages Received Mask When clear, when the Fatal Error Messages Received status bit in the RPERRSTS register is 1b, the LER_SS Status bit in this register will be set. When set, the Fatal Error Messages Received status bit in the RPERRSTS register has no impact on the LER_SS Status bit in this register |
| 2 | RWS | 0 | ERRSTS_Non-Fatal Error Messages Received Mask When clear, when the Non-Fatal Error Messages Received status bit in the RPERRSTS register is 1b, the LER_SS Status bit in this register will be set. When set, the Non-Fatal Error Messages Received status bit in the RPERRSTS register has no impact on the LER_SS Status bit in this register |
| 1 | RV | 0 | <i>Reserved</i> |
| 0 | RWS | 0 | LER_SS Enable Note : LER_SS should be disabled on the ESI port. When set, as long as the LER_SS Status bit in this register is set, the associated root port will go into LER/Stop and Scream mode. When clear, the root port can never go into LER/Stop and Scream mode. The SSMSK register contains additional mask bits that has an impact on whether or not the LER_SS Status bit will be set when LER_SS Enable is set. When a root port enters the LER_SS mode, it automatically brings the associated PCIe link down and the port behaves per the rules states in the PCI Express base spec for link down condition. Also, if the port enters LER_SS mode because of the Outbound Poisoned Data status bit being set (in the XPUNCERRSTS register), the associated outbound data should never be sent to the device south i.e. the link should go down before the poisoned data escapes to the PCIe link. SW can later clear the associated status bits (in either the XPUNCERRSTS, RPERRSTS, or UNCERRSTS register) and thus cause the LER_SS status bit in this register to clear and that would bring the port out of the LER_SS mode, and the port continues working normally i.e. the PCIe link starts training again and normal operation follows. It is up to software to provide sufficient time for the transactions pending in the inbound and outbound queues of the associated root port to have drained via the normal transaction flows, before causing the LER_SS status bit to clear. This procedure also must be followed if the LER_SS Status bit is set and it is desired to disable LER/Stop and Scream mode, afterwards the LER_SS Enable bit can be cleared Also, note that error logging/escalation as defined in the PCI Express spec, via AER registers and MSI mechanism remain unaffected by this bit. Do not enable on the ESI port. |

21.13.10 XP[10:0]ERRCNTSEL: Error Counter Selection Register

| Register: XP[10:0]ERRCNTSEL Device: 0-10 Function: 0 Offset: 400h | | | |
|--|------|---------|-------------|
| Bit | Attr | Default | Description |
| 31:0 | RV | 0h | Reserved |



21.13.11 XP[10:0]ERRCNT: Error Counter Register

| Register: XP[10:0]ERRCNT Device: 0-10 Function: 0 Offset: 404h | | | |
|---|------|---------|-------------|
| Bit | Attr | Default | Description |
| 7:0 | RV | 0 | Reserved |

21.14 Intel VT-d Memory Mapped Register

Table 21-30. Intel VT-d Memory Mapped Registers - 0x00 - 0xFF, 1000-10FF

| | | | |
|---------------------|-----|-----------------------|-----|
| VTD_VERSION | 00h | INV_QUEUE_HEAD | 80h |
| | 04h | NV_QUEUE_HEAD | 84h |
| VTD_CAP | 08h | INV_QUEUE_TAIL | 88h |
| | 0Ch | | 8Ch |
| EX_VTD_CAP | 10h | INV_QUEUE_ADD | 90h |
| | 14h | | 94h |
| GLBCMD | 18h | | 98h |
| GLBSTS | 1Ch | INV_COMP_STATUS | 9Ch |
| ROOTENTRYADD | 20h | INV_COMP_EVT_CTL | A0h |
| | 24h | INV_COMP_EVT_DATA | A4h |
| CTXCMD | 28h | INV_COMP_EVT_ADDR | A8h |
| | 2Ch | | ACh |
| | 30h | | B0h |
| FLTSTS | 34h | | B4h |
| FLTEVTCTRL | 38h | INTR_REMAP_TABLE_BASE | B8h |
| FLTEVTDATA | 3Ch | | BCh |
| FLTEVTADDR | 40h | | C0h |
| FLTEVTUPRADDR | 44h | | C4h |
| | 48h | | C8h |
| | 4Ch | | CCh |
| | 50h | | D0h |
| | 54h | | D4h |
| | 58h | | D8h |
| | 5Ch | | DCh |
| | 60h | | E0h |
| PMEN | 64h | | E4h |
| PROT_LOW_MEM_BASE | 68h | | E8h |
| PROT_LOW_MEM_LIMIT | 6Ch | | ECh |
| PROT_HIGH_MEM_BASE | 70h | | F0h |
| | 74h | | F4h |
| PROT_HIGH_MEM_LIMIT | 78h | | F8h |
| | 7Ch | | FCh |



Table 21-31. Intel VT-d Memory Mapped Registers - 0x100 - 0x1FF, 0x1100-0x11FF

| | | | | | | |
|---------|--|--|--|------|----------|----------|
| FLTREC0 | | | | 100h | | 180h |
| | | | | 104h | | 184h |
| | | | | 108h | | 188h |
| | | | | 10Ch | | 18Ch |
| FLTREC1 | | | | 110h | | 190h |
| | | | | 114h | | 194h |
| | | | | 118h | | 198h |
| | | | | 11Ch | | 19Ch |
| FLTREC2 | | | | 120h | | 1A0h |
| | | | | 124h | | 1A4h |
| | | | | 128h | | 1A8h |
| | | | | 12Ch | | 1AC h |
| FLTREC3 | | | | 130h | | 1B0h |
| | | | | 134h | | 1B4h |
| | | | | 138h | | 1B8h |
| | | | | 13Ch | | 1BC h |
| FLTREC4 | | | | 140h | | 1C0h |
| | | | | 144h | 1C4h | |
| | | | | 148h | 1C8h | |
| | | | | 14Ch | 1CC h | |
| FLTREC5 | | | | 150h | 1D0 h | |
| | | | | 154h | 1D4 h | |
| | | | | 158h | 1D8 h | |
| | | | | 15Ch | 1DC h | |
| FLTREC6 | | | | 160h | 1E0h | |
| | | | | 164h | 1E4h | |
| | | | | 168h | 1E8h | |
| | | | | 16Ch | 1ECh | |
| FLTREC7 | | | | 170h | 1F0h | |
| | | | | 174h | 1F4h | |
| | | | | 178h | 1F8h | |
| | | | | 17Ch | 1FCh | |

Configuration Register Space

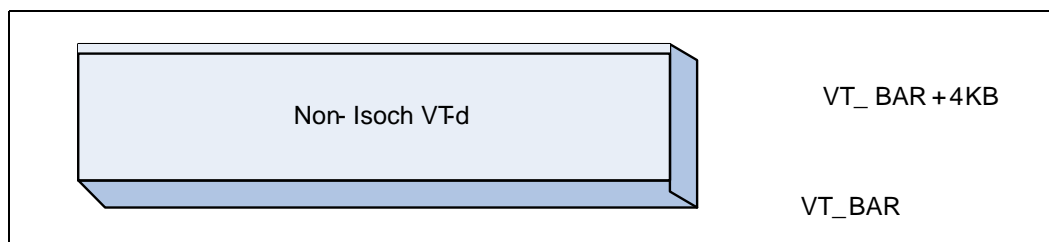


| | | | | | | |
|------------|------|--|--|------|----------|------|
| IOTLBINV | | | | 200h | | 280h |
| | | | | 204h | | 284h |
| INVADDRREG | | | | 208h | | 288h |
| | | | | 20Ch | | 28Ch |
| | | | | 210h | 290h | |
| | | | | 214h | 294h | |
| | | | | 218h | 298h | |
| | | | | 21Ch | 29Ch | |
| | | | | 220h | 2A0h | |
| | | | | 224h | 2A4h | |
| | | | | 228h | 2A8h | |
| | | | | 22Ch | 2AC h | |
| | | | | 230h | 2B0h | |
| | | | | 234h | 2B4h | |
| | | | | 238h | 2B8h | |
| | | | | 23Ch | 2BC h | |
| | | | | 240h | 2C0h | |
| | | | | 244h | 2C4h | |
| | | | | 248h | 2C8h | |
| | | | | 24Ch | 2CC h | |
| | | | | 250h | 2D0 h | |
| | | | | 254h | 2D4 h | |
| | | | | 258h | 2D8 h | |
| | | | | 25Ch | 2DC h | |
| | | | | 260h | 2E0h | |
| | | | | 264h | 2E4h | |
| | | | | 268h | 2E8h | |
| | | | | 26Ch | 2ECh | |
| | | | | 270h | 2F0h | |
| | | | | 274h | 2F4h | |
| | | | | 278h | 2F8h | |
| 27Ch | 2FCh | | | | | |

21.14.1 Intel VT-d Memory Mapped Registers

The Intel VT-d registers are all addressed using aligned DWORD or aligned QWORD accesses. Any combination of BEs is allowed within a DWORD or QWORD access. The Intel VT-d remap engine registers corresponding to the non-Isoch port represented by Device#0, occupy the first 4K of offset starting from the base address defined by VTBAR register.

Figure 21-2. Base Address of Intel VT-D Remap Engines



21.14.1.1 VTD_VERSION: Version Number Register

| Register: VTD_VERSION Addr: MMIO BAR: VTBAR Offset: 00h, 1000h | | | |
|---|------|---------|----------------|
| Bit | Attr | Default | Description |
| 31:8 | RV | 0h | Reserved |
| 7:4 | RO | 1h | Major Revision |
| 3:0 | RO | 0h | Minor Revision |

21.14.1.2 VTD_CAP[0:1]: Intel VT-d Capability Register

| Register: EXT_VTD_CAP[0:1] Addr: MMIO BAR: VTBAR Offset: 08h, 1008h | | | |
|--|------|-----------------------|---|
| Bit | Attr | Default | Description |
| 63:56 | RV | 0h | Reserved |
| 55 | RO | 1h | DMA Read Draining: Supported in IOH |
| 54 | RO | 1h | DMA Write Draining: Supported in IOH |
| 53:48 | RO | 09h | MAMV: IOH support MAMV value of 9h. |
| 47:40 | RO | 8h | Number of Fault Recording Registers: IOH supports 8 fault recording registers. |
| 39 | RO | 1h | Page Selective Invalidation: Supported in IOH |
| 38 | RV | 0h | Reserved |
| 37:34 | RV | 0h | Reserved |
| 33:24 | RO | 10h | Fault Recording Register Offset: Fault registers are at offset 100h |
| 23 | RWO | 0h (Offset 08h) | Reserved |
| 22 | RWO | 0h | Zero length DMA requests to write-only pages supported. |



| Register: EXT_VTD_CAP[0:1] Addr: MMIO BAR: VTBAR Offset: 08h, 1008h | | | |
|--|------|---------------------|--|
| Bit | Attr | Default | Description |
| 21:16 | RO | 2Fh (Offset 08h) | MGAW: This register is set by IOH based on the setting of the Non-Isch GPA_LIMIT field in the VTGENCTRL register. |
| 15:13 | RV | 0h | Reserved |
| 12:8 | RO | 2h (Offset 08h) | SAGAW: IOH supports only 4 level walks on the VT-d engine |
| 7 | RO | 0 | TCM: IOHTB does not cache invalid pages |
| 6 | RO | 1 | PHMR Support: IOH supports protected high memory range |
| 5 | RO | 1 | PLMR Support: IOH supports protected low memory range |
| 4 | RV | 0 | Reserved |
| 3 | RO | 0 | Advanced Fault Logging: IOHTB does not support advanced fault logging |
| 2:0 | RO | 010b | Number of Domains Supported: IOH supports 256 domains with 8 bit domain ID |

21.14.1.3 EXT_VTD_CAP[0:1]: Extended Intel VT-d Capability Register

| Register: EXT_VTD_CAP[0:1] Addr: MMIO BAR: VTBAR Offset: 10h, 1010h | | | |
|--|------|----------------------------------|--|
| Bit | Attr | Default | Description |
| 63:32 | RV | 0h | Reserved |
| 23:20 | RO | Fh | Maximum Handle Mask Value: IOH supports all 16 bits of handle being masked. Note IOH always performs global interrupt entry invalidation on any interrupt cache invalidation command and h/w never really looks at the mask value |
| 19:18 | RV | 0h | Reserved |
| 17:8 | RO | 20h | Invalidation Unit Offset: IOH has the invalidation registers at offset 200h |
| 7 | RWO | 1 (Offset 10h), 0 (Offset 1010h) | 0: Hardware does not support 1-setting of the SNP field in the page-table entries. 1: Hardware supports the 1-setting of the SNP field in the page-table entries. IOH supports snoop override only for the non-isoch Intel VT-d engine |
| 6 | RWO | 1 | IOH supports pass through Note that when this bit is set to 0, Intel VT-d spec requires error checking on the "type" field, that is, "type" field cannot have an encoding of 10b. If software set an encoding of 10b, h/w has to cause a fault |
| 5 | RO | 1 | IOH supports caching hints |
| 4 | RO | See Description | IA32 Extended Interrupt Mode: Default is 1 if DISAPICEXT = 0, else default = 0 |
| 3 | RO | 1 | Interrupt Remapping Support: IOH supports this |
| 2 | RWO | 1 (Offset 10h), 0 (Offset 1010h) | Device TLB support: IOH supports ATS for the non-isoch Intel VT-d engine. Note that when this bit is set to 0, Intel VT-d spec requires error checking on the "type" field, that is, "type" field cannot have an encoding of 01b. If software set an encoding of 01b, h/w has to cause a fault. |
| 1 | RWO | 1 | Queued Invalidation support: IOH supports this |
| 0 | RWO | 0 | Coherency Support: BIOS can write to this bit to indicate to hardware to either snoop or not-snoop the Interrupt table structures in memory (root/context/pd/pt/irt) |



21.14.1.4 GLBCMD[0:1]: Global Command Register

| Register: GLBCMD[0:1] Addr: MMIO BAR: VTBAR Offset: 18h, 1018h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31 | RW | 0h | Translation Enable: This bit is used by software to disable (0) DMA-remapping in hardware or enable(1) in software. |
| 30 | RW | 0h | Set Root Table Pointer: Software sets this field to set/update the root-entry table pointer used by hardware. The root-entry table pointer is specified through the Root-entry Table Address register. Hardware reports the status of the root table pointer set operation through the RTPS field in the Global Status register. |
| 29 | RO | 0 | Set Fault Log Pointer: N/A to IOH |
| 28 | RO | 0 | Enable Advanced Fault Logging: N/A to IOH |
| 27 | RO | 0 | Write Buffer Flush: N/A to IOH |
| 26 | RW | 0 | Queued Invalidation Enable: Software writes to this field to enable queued invalidations. 0: Disable queued invalidations. In this case, invalidations must be performed through the Context Command and IOTLB Invalidation Unit registers. 1: Enable use of queued invalidations. Once enabled, all invalidations must be submitted through the invalidation queue and the invalidation registers can no longer be used without going through an IOH reset. The invalidation queue address register must be initialized before enabling queued invalidations. Also software must make sure that all invalidations submitted prior via the register interface are all completed before enabling the queued invalidation interface. |
| 25 | RW | 0 | Interrupt Remapping Enable: 0: Disable Interrupt Remapping Hardware 1: Enable Interrupt Remapping Hardware Hardware reports the status of the interrupt-remap enable operation through the IRES field in the Global Status register. |
| 24 | RW | 0 | Set Interrupt Remap Table Pointer: Software sets this field to set/update the Interrupt Remap Table Pointer used by hardware. The interrupt remapping Table Pointer is specified through the interrupt Remapping Table Address Register. |
| 23 | RW | 0 | Compatibility Format Interrupt: Software writes to this field to enable / disable Compatibility Format Interrupt. The value reported in this field is reported only when interrupt-remapping is enabled and Legacy interrupt mode is active 0: Compatibility Format Interrupts Blocked 1: Compatibility Format Interrupts Processed as bypassing interrupt remapping. |
| 22:0 | RV | 0 | Reserved |

21.14.1.5 GLBSTS[0:1]: Global Status Register

| Register: GLBSTS[0:1] Addr: MMIO BAR: VTBAR Offset: 1Ch, 101Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31 | RO | 0 | Translation Enable Status: When set, indicates that translation hardware is enabled and when clear indicates the translation hardware is not enabled. |



| Register: GLBSTS[0:1] Addr: MMIO BAR: VTBAR Offset: 1Ch, 101Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 30 | RO | 0 | Set Root Table Pointer Status: This field indicates the status of the root- table pointer in hardware. |
| 29 | RO | 0 | Set Fault Log Pointer: N/A to IOH |
| 28 | RO | 0 | Advanced Fault Logging Status: N/A to IOH |
| 27 | RO | 0 | Write Buffer Flush: N/A to IOH |
| 26 | RO | 0 | Queued Invalidation Interface Status: IOH sets this bit once it has completed the software command to enable the queued invalidation interface. Until then this bit is 0. |
| 25 | RO | 0 | Interrupt Remapping Enable Status: IOH sets this bit once it has completed the software command to enable the interrupt remapping interface. Till then this bit is 0 |
| 24 | RO | | Interrupt Remapping Table Pointer Status: This field indicates the status of the interrupt remapping table pointer in hardware. This field is cleared by hardware when software sets the SIRTTP field in the Global Command register. This field is set by hardware when hardware completes the set interrupt remap table pointer operation using the value provided in the Interrupt Remapping Table Address register |
| 23 | RO | 0 | Compatibility Format Interrupt Status: The value reported in this field is reported only when interrupt-remapping is enabled and Legacy interrupt mode is active 0: Compatibilty Format Interrupts Blocked 1: Compatibilty Format Interrupts Processed as bypassing interrupt remapping. |
| 22:0 | RV | 0 | Reserved |

21.14.1.6 ROOTENTRYADD[0:1]: Root Entry Table Address Register

| Register: ROOTENTRYADD[0:1] Addr: MMIO BAR: VTBAR Offset: 20h, 1020h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 63:12 | RW | 0 | Root Entry Table Base Address: 4K aligned base address for the root entry table. IOH does not utilize bits 63:43 and checks for them to be 0. Software specifies the base address of the root-entry table through this register, and enables it in hardware through the <i>SRTTP</i> field in the <i>Global Command</i> register. Reads of this register returns value that was last programmed to it. |
| 11:0 | RV | 0 | Reserved |



21.14.1.7 CTXCMD[0:1]: Context Command Register

| Register: CTXCMD[0:1] Addr: MMIO BAR: VTBAR Offset: 28h, 1028 | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 63 | RW | 0 | <p>Invalidate Context Entry Cache (ICC): Software requests invalidation of context-cache by setting this field. Software must also set the requested invalidation granularity by programming the CIRG field. Software must read back and check the ICC field to be clear to confirm the invalidation is complete. Software must not update this register when this field is set. Hardware clears the ICC field to indicate the invalidation request is complete. Hardware also indicates the granularity at which the invalidation operation was performed through the CAIG field. Software must not submit another invalidation request through this register while the ICC field is set.</p> <p>Since information from the context-cache may be used by hardware to tag IOTLB entries, software must perform domain-selective (or global) invalidation of IOTLB after the context cache invalidation has completed</p> |
| 62:61 | RW | 0 | <p>Context Invalidation Request Granularity (CIRG): When requesting hardware to invalidate the context-entry cache (by setting the ICC field), software writes the requested invalidation granularity through this field. Following are the encoding for the 2-bit IRG field.</p> <p>00: <i>Reserved</i></p> <p>01: Global Invalidation request. IOH supports this.</p> <p>10: Domain-selective invalidation request. The target domain-id must be specified in the DID field. IOH supports this.</p> <p>11: Device-selective invalidation request. The target SID must be specified in the SID field, and the domain-id (programmed in the context-entry for this device) must be provided in the DID field. IOH does not support this and alias this request to a domain-selective invalidation request. IOH supports this.</p> <p>Hardware indicates completion of the invalidation request by clearing the ICC field. At this time, hardware also indicates the granularity at which the actual invalidation was performed through the CAIG field.</p> |
| 60:59 | RO | 0 | <p>Context Actual Invalidation Granularity (CAIG): Hardware reports the granularity at which an invalidation request was processed through the CAIG field at the time of reporting invalidation completion (by clearing the ICC field). The following are the encoding for the 2-bit CAIG field.</p> <p>00: <i>Reserved</i>. This is the value on reset.</p> <p>01: Global Invalidation performed. IOH sets this in response to a global invalidation request.</p> <p>10: Domain-selective invalidation performed using the domain-id that was specified by software in the DID field. IOH sets this in response to a domain-selective or device-selective invalidation request. IOH set this in response to a domain-selective invalidation request.</p> <p>11: Device-selective invalidation performed.</p> |
| 58:34 | RV | 0 | Reserved |
| 33:32 | RW | 0 | Function Mask: Since IOH does not perform any device selective invalidation, this field is a don't care. Used by IOH when performing device selective invalidation |
| 31:16 | RW | 0 | Source ID: IOH ignores this field. Used by IOH when performing device selective context cache invalidation. |
| 15:0 | RW | 0 | Domain ID: Indicates the id of the domain whose context-entries needs to be selectively invalidated. S/W needs to program this for both domain and device selective invalidates. IOH ignores bits 15:8 since it supports only a 8 bit Domain ID. |



21.14.1.8 FLTSTS[0:1]: Fault Status Register

| Register: FLTSTS[0:1] Addr: MMIO BAR: VTBAR Offset: 34h, 1034h | | | |
|---|-------|---------|--|
| Bit | Attr | Default | Description |
| 31:16 | RV | 0 | Reserved |
| 15:8 | ROS | 0 | Fault Record Index: This field is valid only when the Primary Fault Pending field is set. This field indicates the index (from base) of the fault recording register to which the first pending fault was recorded when the Primary Fault pending field was set by hardware. |
| 7 | RV | 0 | Reserved |
| 6 | RW1CS | 0 | Invalidation Timeout Error: Hardware detected a Device-IOTLB invalidation completion time-out. At this time, a fault event may be generated based on the programming of the Fault Event Control register. Hardware implementations not supporting Device-IOTLBs may implement this bit as RO. |
| 5 | RW1CS | 0 | Invalidation Completion Timeout: Hardware received no ATS invalidation completions during an invalidation completion timeout period, while there are one or more pending ATS invalidation requests waiting for invalidation completions. At this time, a fault event is generated based on the programming of the Fault Event Control register. |
| 4 | RW1CS | 0 | Invalidation Queue Error: Hardware detected an error associated with the invalidation queue. For example, hardware detected an erroneous or unsupported Invalidation Descriptor in the Invalidation Queue. At this time, a fault event is generated based on the programming of the Fault Event Control register. |
| 3:2 | RV | 0 | Reserved |
| 1 | ROS | 0 | Primary Fault Pending: This field indicates if there are one or more pending faults logged in the fault recording registers. 0: No pending faults in any of the fault recording registers 1: One or more fault recording registers has pending faults. The fault recording index field is updated by hardware whenever this field is set by hardware. Also, depending on the programming of fault event control register, a fault event is generated when hardware sets this field. |
| 0 | RW1CS | 0 | Primary Fault Overflow: Hardware sets this bit to indicate overflow of fault recording registers |



21.14.1.9 FLTEVTCTRL[0:1]: Fault Event Control Register

| Register: FLTEVTCTRL[0:1] Addr: MMIO BAR: VTBAR Offset: 38h, 1038h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31 | RW | 1 | Interrupt Message Mask: 1: Hardware is prohibited from issuing interrupt message requests. 0: Software has cleared this bit to indicate interrupt service is available. When a faulting condition is detected, hardware may issue a interrupt request (using the fault event data and fault event address register values) depending on the state of the interrupt mask and interrupt pending bits. |
| 30 | RO | 0 | Interrupt Pending: Hardware sets the IP field whenever it detects an interrupt condition. Interrupt condition is defined as when an interrupt condition occurs when hardware records a fault through one of the Fault Recording registers and sets the PPF field in Fault Status register. - Hardware detected error associated with the Invalidation Queue, setting the IQE field in the Fault Status register. - Hardware detected invalidation completion timeout error, setting the ICT field in the Fault Status register. - If any of the above status fields in the Fault Status register was already set at the time of setting any of these fields, it is not treated as a new interrupt condition. The IP field is kept set by hardware while the interrupt message is held pending. The interrupt message could be held pending due to interrupt mask (IM field) being set, or due to other transient hardware conditions. The IP field is cleared by hardware as soon as the interrupt message pending condition is serviced. This could be due to either (a) Hardware issuing the interrupt message due to either change in the transient hardware condition that caused interrupt message to be held pending or due to software clearing the IM field. (b) Software servicing all the pending interrupt status fields in the Fault Status register. - PPF field is cleared by hardware when it detects all the Fault Recording registers have Fault (F) field clear. - Other status fields in the Fault Status register is cleared by software writing back the value read from the respective fields. |
| 29:0 | RV | 0 | Reserved |

21.14.1.10 FLTEVTDATA[0:1]: Fault Event Data Register

| Register: FLTEVTDATA[0:1] Addr: MMIO BAR: VTBAR Offset: 3Ch, 103Ch | | | |
|---|------|---------|----------------|
| Bit | Attr | Default | Description |
| 31:16 | RV | 0 | Reserved |
| 15:0 | RW | 0 | Interrupt Data |



21.14.1.11 FLTEVTADDR[0:1]: Fault Event Address Register

| Register: FLTEVTADDR[0:1] Addr: MMIO BAR: VTBAR Offset: 40h, 1040h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31:2 | RO | 0 | Interrupt Address: The interrupt address is interpreted as the address of any other interrupt from a PCI Express port. |
| 1:0 | RV | 0 | Reserved |

21.14.1.12 FLTEVTUPADDR[0:1]: Fault Event Upper Address Register

| Register: FLTEVTUPADDR[0:1] Addr: MMIO BAR: VTBAR Offset: 44h, 1044h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:0 | RW | 0 | Address : IOH supports extended interrupt mode |

21.14.1.13 PMEN[0:1] : Protected Memory Enable Register

| Register: PMEN[0:1] Addr: MMIO BAR: VTBAR Offset: 64h, 1064h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31 | RWL | 0 | Enable Protected Memory as defined by the PROT_LOW(HIGH)_BASE and PROT_LOW(HIGH)_LIMIT registers This bit is RO when LT.CMD.LOCK.PMRC (OFFSET 0000h: LT.STS[12]=1) and RW when LT.CMD.LOCK.PMRC (OFFSET 0000h: LT.STS[12]=0) |
| 30:1 | RV | 0 | Reserved |
| 0 | RO | 0 | Protected Region Status: This bit is set by IOH whenever it has completed enabling the protected memory region per the rules stated in the Intel VT-d spec |

21.14.1.14 PROT_LOW_MEM_BASE[0:1] : Protected Memory Low Base Register

| Register: PROT_LOW_MEM_BASE[0:1] Addr: MMIO BAR: VTBAR Offset: 68h, 1068h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 31:21 | RWL | 0 | 2 MB aligned base address of the low protected dram region This bit is RO when LT.CMD.LOCK.PMRC (OFFSET 0000h: LT.STS[12]=1) and RW when LT.CMD.LOCK.PMRC (OFFSET 0000h: LT.STS[12]=0) Note that Intel VT-d engine generated reads/writes (page walk, interrupt queue, invalidation queue read, invalidation status) themselves are allowed toward this region |
| 20:0 | RV | 0 | Reserved |



21.14.1.15 PROT_LOW_MEM_LIMIT[0:1] : Protected Memory Low Limit Register

| Register: PROT_LOW_MEM_LIMIT[0:1] Addr: MMIO BAR: VTBAR Offset: 6Ch, 106Ch | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 31:24 | RWL | 0 | 2 MB aligned limit address of the low protected dram region This bit is RO when LT.CMD.LOCK.PMRC (OFFSET 0000h: LT.STS[12]=1) and RW when LT.CMD.LOCK.PMRC (OFFSET 0000h: LT.STS[12]=0) |
| 23:0 | RV | 0 | Reserved |

21.14.1.16 PROT_HIGH_MEM_BASE[0:1] : Protected Memory High Base Register

| Register: PROT_HIGH_MEM_BASE[0:1] Addr: MMIO BAR: VTBAR Offset: 70h, 1070h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 63:24 | RWL | 0 | 2 MB aligned base address of the high protected dram region This bit is RO when LT.CMD.LOCK.PMRC (OFFSET 0000h: LT.STS[12]=1) and RW when LT.CMD.LOCK.PMRC (OFFSET 0000h: LT.STS[12]=0) Note that Intel VT-d engine generated reads/writes (page walk, interrupt queue, invalidation queue read, invalidation status) themselves are allowed toward this region |
| 23:0 | RV | 0 | Reserved |

21.14.1.17 PROT_HIGH_MEM_LIMIT[0:1] : Protected Memory Limit Base Register

| Register: PROT_HIGH_MEM_LIMIT[0:1] Addr: MMIO BAR: VTBAR Offset: 78h, 1078h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 63:21 | RWL | 0 | 2 MB aligned limit address of the high protected dram region This bit is RO when LT.CMD.LOCK.PMRC (OFFSET 0000h: LT.STS[12]=1) and RW when LT.CMD.LOCK.PMRC (OFFSET 0000h: LT.STS[12]=0) Note that Intel VT-d engine generated reads/writes (page walk, interrupt queue, invalidation queue read, invalidation status) themselves are allowed toward this region |
| 20:0 | RV | 0 | Reserved |



21.14.1.18 INV_QUEUE_HEAD[0:1]: Invalidation Queue Header Pointer Register

| Register: INV_QUEUE_HEAD[0:1] Addr: MMIO BAR: VTBAR Offset: 80h, 1080h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 63:19 | RV | 0 | Reserved |
| 18:4 | RO | 0 | Queue Head: Specifies the offset (128-bit aligned) to the invalidation queue for the command that will be fetched next by hardware. This field is incremented after the command has been fetched successfully and has been verified to be a valid/supported command. |
| 3:0 | RV | 0 | Reserved |

21.14.1.19 INV_QUEUE_TAIL[0:1]: Invalidation Queue Tail Pointer Register

| Register: INV_QUEUE_TAIL[0:1] Addr: MMIO BAR: VTBAR Offset: 88h, 1088h | | | |
|---|------|---------|--|
| Bit | Attr | Default | Description |
| 63:19 | RV | 0 | Reserved |
| 18:4 | RW | 0 | Queue Tail: Specifies the offset (128-bit aligned) to the invalidation queue for the command that will be written next by software. |
| 3:0 | RV | 0 | Reserved |

21.14.1.20 INV_QUEUE_ADD[0:1]: Invalidation Queue Address Register

| Register: INV_QUEUE_ADD[0:1] Addr: MMIO BAR: VTBAR Offset: 90h, 1090h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 63:12 | RW | 0 | This field points to the base of size-aligned invalidation request queue. |
| 11:3 | RV | 0 | Reserved |
| 2:0 | RW | 0 | Queue Size: This field specifies the length of the invalidation request queue. The number of entries in the invalidation queue is defined as $2^{(X + 8)}$, where X is the value programmed in this field. |



21.14.1.21 INV_COMP_STATUS[0:1]: Invalidation Completion Status Register

| Register: INV_COMP_STATUS[0:1] Addr: MMIO BAR: VTBAR Offset: 9Ch, 109Ch | | | |
|--|-------|---------|---|
| Bit | Attr | Default | Description |
| 31:1 | RV | 0 | Reserved |
| 0 | RW1CS | 0 | Invalidation Wait Descriptor Complete: Indicates completion of Invalidation Wait Descriptor with Interrupt Flag (IF) field set |

21.14.1.22 INV_COMP_EVT_CTL[0:1]: Invalidation Completion Event Control Register

| Register: INV_COMP_EVT_CTL[0:1] Addr: MMIO BAR: VTBAR Offset: A0h, 10A0h | | | |
|---|------|---------|---|
| Bit | Attr | Default | Description |
| 31 | RW | 0 | Interrupt Mask: 0: No masking of interrupt. When a invalidation event condition is detected, hardware issues an interrupt message (using the Invalidation Event Data & Invalidation Event Address register values). 1: This is the value on reset. Software may mask interrupt message generation by setting this field. Hardware is prohibited from sending the interrupt message when this field is set. |
| 30 | RO | 0 | Interrupt Pending: Hardware sets the IP field whenever it detects an interrupt condition. Interrupt condition is defined as: - An Invalidation Wait Descriptor with Interrupt Flag (IF) field set completed, setting the IWC field in the Fault Status register. - If the IWC field in the Invalidation Event Status register was already set at the time of setting this field, it is not treated as a new interrupt condition. The IP field is kept set by hardware while the interrupt message is held pending. The interrupt message could be held pending due to interrupt mask (IM field) being set, or due to other transient hardware conditions. The IP field is cleared by hardware as soon as the interrupt message pending condition is serviced. This could be due to either: (a) Hardware issuing the interrupt message due to either change in the transient hardware condition that caused interrupt message to be held pending or due to software clearing the IM field. (b) Software servicing the IWC field in the Fault Status register. |
| 29:0 | RV | 0 | Reserved |

21.14.1.23 INV_COMP_EVT_DATA[0:1]: Invalidation Completion Event Data Register

| Register: INV_COMP_EVT_DATA[0:1] Addr: MMIO BAR: VTBAR Offset: A4h, 10A4h | | | |
|--|------|---------|-------------|
| Bit | Attr | Default | Description |
| 31:16 | RV | 0 | Reserved |



| Register: INV_COMP_EVT_DATA[0:1] Addr: MMIO BAR: VTBAR Offset: A4h, 10A4h | | | |
|--|------|---------|----------------|
| Bit | Attr | Default | Description |
| 15:0 | RW | 0 | Interrupt Data |

21.14.1.24 INV_COMP_EVT_ADDR[0:1]: Invalidation Completion Event Address Register

| Register: INV_COMP_EVT_ADDR[0:1] Addr: MMIO BAR: VTBAR Offset: A8h, 10A8h | | | |
|--|------|---------|-------------------|
| Bit | Attr | Default | Description |
| 31:2 | RW | 0 | Interrupt Address |
| 1:0 | RV | 0 | Reserved |

21.14.1.25 INTR_REMAP_TABLE_BASE[0:1]: Interrupt Remapping Table Base Address Register

| Register: INTR_REMAP_TABLE_BASE[0:1] Addr: MMIO BAR: VTBAR Offset: B8h, 10B8h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 63:12 | RW | 0 | Intr Remap Base: This field points to the base of page-aligned interrupt remapping table. If the Interrupt Remapping Table is larger than 4KB in size, it must be size-aligned. Reads of this field returns value that was last programmed to it. |
| 11 | RWL | 0 | IA32 Extended Interrupt Enable 0: IA32 system is operating in legacy IA32 interrupt mode. Hardware interprets only 8-bit APICID in the Interrupt Remapping Table entries. 1: IA32 system is operating in extended IA32 interrupt mode. Hardware interprets 32-bit APICID in the Interrupt Remapping Table entries. |
| 10:4 | RV | 0 | Reserved |
| 3:0 | RW | 0 | Size: This field specifies the size of the interrupt remapping table. The number of entries in the interrupt remapping table is $2^{(X+1)}$, where X is the value programmed in this field. |



21.14.1.26 FLTREC: Fault Record Register

| Register: FLTREC Addr: MMIO BAR: VTBAR Offset: [170h:100h], [1170:1100h] | | | |
|---|-------|---------|---|
| Bit | Attr | Default | Description |
| 127 | RW1CS | 0 | Fault (F): Hardware sets this field to indicate a fault is logged in this fault recording register. The F field is set by hardware after the details of the fault is recorded in the PADDR, SID, FR and T fields. When this field is set, hardware may collapse additional faults from the same requestor (SID). Software writes the value read from this field to clear it. |
| 126 | ROS | 0 | Type of faulted DMA Request 0: DMA Write 1: DMA Read Request Field valid only when F bit is set |
| 125:124 | ROS | 0 | Address Type: This field captures the AT field from the faulted DMA request. Field valid only when F bit is set |
| 123:104 | RV | 0 | Reserved |
| 103:96 | ROS | 0 | Fault Reason: Reason for the first translation fault. See Intel VT-d spec for details. This field is only valid when Fault bit is set. |
| 95:80 | RV | 0 | Reserved |
| 79:64 | ROS | 0 | Source Identifier: Requester ID that faulted. Valid only when F bit is set |
| 63:12 | ROS | 0 | GPA: 4k aligned GPA for the faulting transaction. Valid only when F field is set |
| 11:0 | RV | 0 | Reserved |

21.14.1.27 IOTLBINV[0:1] : IOTLB Invalidate Register

| Register: IOTLBINV[0:1] Addr: MMIO BAR: VTBAR Offset: 208h, 1208h | | | |
|--|------|---------|--|
| Bit | Attr | Default | Description |
| 63 | RW | 0 | Invalidate IOTLB cache (IVT): Software requests IOTLB invalidation by setting this field. Software must also set the requested invalidation granularity by programming the IIRG field. Hardware clears the IVT field to indicate the invalidation request is complete. Hardware also indicates the granularity at which the invalidation operation was performed through the IAIG field. Software must read back and check the IVT field to be clear to confirm the invalidation is complete. When IVT field is set, software must not update the contents of this register (and Invalidate Address register, i if it is being used), nor submit new IOTLB invalidation requests. |
| 62 | RV | 0 | Reserved |



| Register: IOTLBINV[0:1] Addr: MMIO BAR: VTBAR Offset: 208h, 1208h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 61:60 | RW | 0 | IOTLB Invalidation Request Granularity (IIRG): When requesting hardware to invalidate the I/OTLB (by setting the IVT field), software writes the requested invalidation granularity through this IIRG field. Following are the encoding for the 2-bit IIRG field. 00: Reserved. Hardware ignores the invalidation request and reports invalidation complete by clearing the IVT field and reporting 00 in the AIG field. 01: Global Invalidation request. IOH supports this. 10: Domain-selective invalidation request. The target domain-id must be specified in the DID field. IOH supports this. 11: Page-selective invalidation request. The target address, mask and invalidation hint must be specified in the Invalidate Address register, the domain-id must be provided in the DID field. IOH supports this. |
| 59 | RV | 0 | Reserved |
| 58:57 | RO | 0 | IOTLB Actual Invalidation Granularity (IAIG): Hardware reports the granularity at which an invalidation request was proceed through the AIG field at the time of reporting invalidation completion (by clearing the IVT field). The following are the encoding for the 2-bit IAIG field. 00: Reserved. 01: Global Invalidation performed. IOH sets this in response to a global IOTLB invalidation request. 010: Domain-selective invalidation performed using the domain-id that was specified by software in the DID field. IOH sets this in response to a domain-selective IOTLB invalidation request. 11: IOH sets this in response to a page selective invalidation request. |
| 56:50 | RV | 0 | Reserved |
| 49 | RW | 0 | Drain Reads: IOH uses this to drain or not drain reads on an invalidation request. |
| 48 | RW | 0 | Drain Writes: IOH uses this to drain or not drain writes on an invalidation request |
| 47:32 | RW | 0 | Domain ID: Domain to be invalidated and is programmed by software for both page and domain selective invalidation requests. IOH ignores the bits 47:40 since it supports only an 8 bit Domain ID |
| 31:0 | RV | 0 | Reserved |

21.14.1.28 INVADDRREG[1:0]: Invalidate Address Register

| Register: INVADDRREG[1:0] Addr: MMIO BAR: VTBAR Offset: 200h, 1200h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 63:12 | RW | 0 | Address To request a page-specific invalidation request to hardware, software must first write the corresponding guest physical address to this register, and then issue a page-specific invalidate command through the IOTLB_REG. |
| 11:7 | RV | 0 | Reserved |



| Register: INVADDRREG[1:0] Addr: MMIO BAR: VTBAR Offset: 200h, 1200h | | | |
|--|------|---------|---|
| Bit | Attr | Default | Description |
| 6 | RW | 0 | Invalidation Hint: The field provides hint to hardware to preserve or flush the respective non-leaf page-table entries that may be cached in hardware. 0: Software may have modified both leaf and non-leaf page-table entries corresponding to mappings specified in the ADDR and AM fields. On a page-selective invalidation request, IOH must flush both the cached leaf and nonleaf page-table entries corresponding to mappings specified by ADDR and AM fields. IOH performs a domain-level invalidation on non-leaf entries and page-selective-domain-level invalidation at the leaf level 1: Software has not modified any non-leaf page-table entries corresponding to mappings specified in the ADDR and AM fields. On a page-selective invalidation request, IOH preserves the cached non-leaf page-table entries corresponding to mappings specified by ADDR and AM fields and performs only a page-selective invalidation at the leaf level |
| 5:0 | RW | 0 | Address Mask: IOH supports values of 0-9. All other values result in undefined results.. |

§

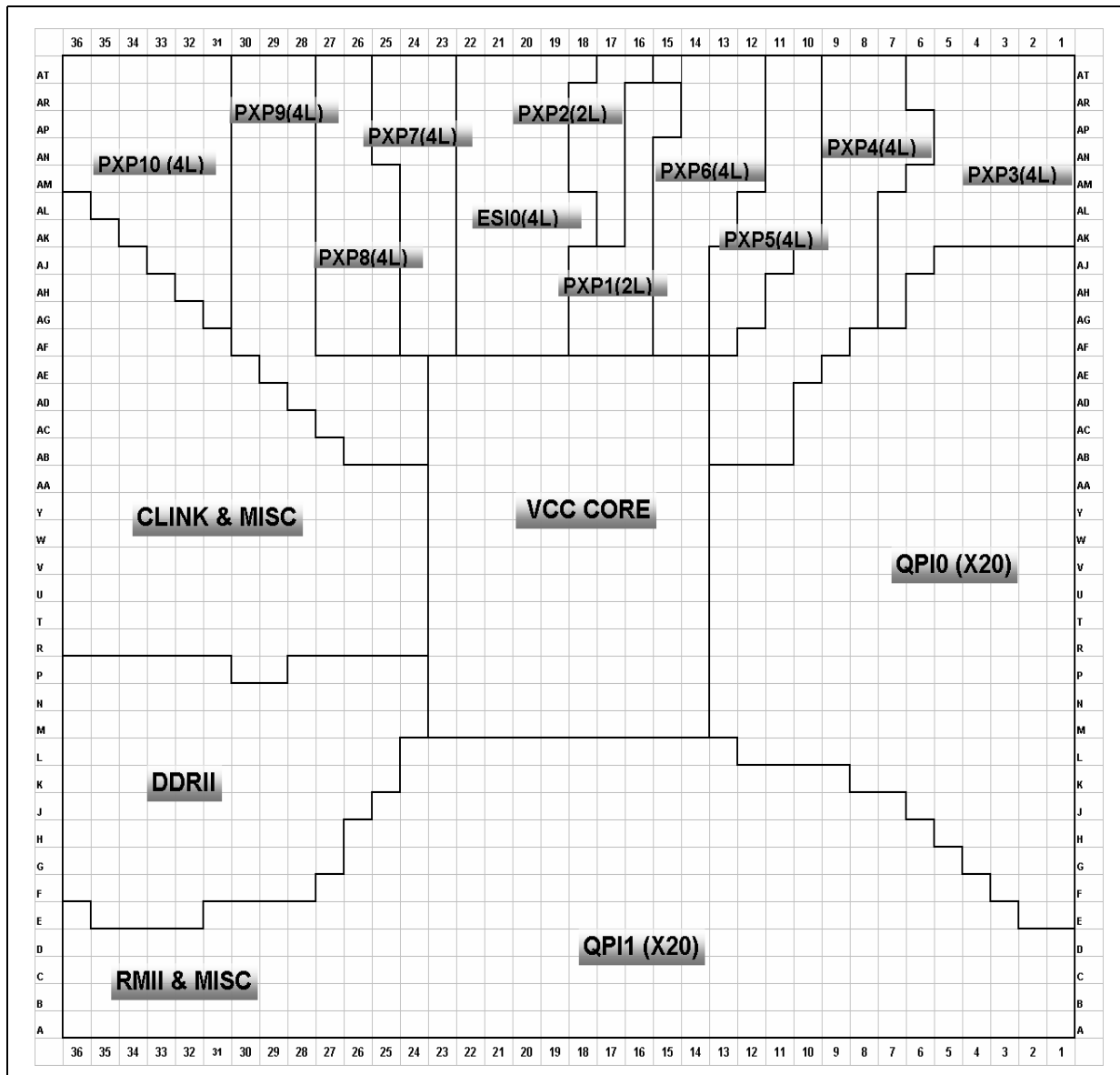


22 Ballout and Package Information

22.1 I/O Hub (IOH) Ballout

The following section presents preliminary ballout information for the IOH. This ballout is subject to change and is to be used for informational purposes only.

Figure 22-1. IOH Quadrant Map





22.2 IOH Pin List and Ballout

Figure 22-2. IOH Ballout Left Side (Top View) (Sheet 1 of 2)

| | 36 | 35 | 34 | 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 |
|----|------------|------------|-----------|------------------|-----------|-------------|-------------|----------------|--------------|------------|--------------|-----------|
| AT | TEST[4] | VSS | VSS | VSS | PE10TN[0] | PE10TP[0] | VSS | PE9TN[1] | PE9TP[1] | PE8TN[3] | PE8TP[3] | VSS |
| AR | VSS | VSS | VCCPE1VRM | PE10TP[2] | VSS | PE10TN[1] | PE9TN[2] | PE9TP[2] | PE7TN[0] | VSS | PE8TN[2] | PE8TN[1] |
| AP | VSS | PE1RCOMP | VSS | PE10TN[2] | PE10TP[3] | PE10TP[1] | PE9TP[3] | VSS | PE7TP[0] | PE7TP[1] | PE8TP[2] | PE7TN[2] |
| AN | PE11COMP | PE11COMPI | PE1RBIAS | RSVD | PE10TN[3] | VSS | PE9TN[3] | PE9TN[0] | PE9TP[0] | PE7TN[1] | VSS | PE7TP[2] |
| AM | VSS | RSVD | PE10RP[1] | VSS | VSS | RSVD | PE1JCLKN | PE1JCLKP | VSS | RSVD | PE8RN[0] | RSVD |
| AL | FERR_N | VSS | PE10RN[1] | PE10RP[0] | PE9RP[2] | PE9RN[2] | VSS | PE9RN[0] | PE8RP[2] | PE8RN[2] | PE8RP[0] | VSS |
| AK | EXTSYSTRIG | PESBLCSEL | VSS | PE10RN[0] | PE10RN[2] | PE9RP[1] | PE9RN[1] | PE9RP[0] | PE9RN[3] | VSS | VSS | PE7RP[3] |
| AJ | VSS | QPISBLCSEL | TESTLO5 | VSS | PE10RP[2] | VSS | PE10RP[3] | VSS | PE9RP[3] | PE8RP[1] | PE8RN[1] | PE8RP[3] |
| AH | XDPCLK1XP | VSS | NODEID[2] | TESTHI[4] | VSS | RSVD | PE10RN[3] | RSVD | RSVD | VSS | RSVD | PE8RN[3] |
| AG | XDPCLK1XN | QPIFREQSEL | VSS | DUALIOH_QPIPRSEL | SMI_N | VSS | RSVD | PE1CLKN | VSS | RSVD | RSVD | VSS |
| AF | VSS | TESTLO8 | TESTLO9 | VSS | TESTLO10 | BMCINIT | VSS | PE1CLKP | ERR_N[1] | VSS | RSVD | VSS |
| AE | XDPDQ[12] | VSS | TESTLO11 | TESTLO25 | VSS | SMBUSID | INIT_N | VSS | ERR_N[2] | LTRESET_N | VSS | VCCAPE1 |
| AD | XDPDQ[8] | XDPDQSP[1] | VSS | TESTLO21 | NMI | VSS | PEHPSDA | SMBSDA | VSS | ERR_N[0] | THERMALERT_N | VCCAPE1 |
| AC | VSS | XDPDQSN[1] | XDPDQ[13] | VSS | TESTLO12 | A20M_N | VSS | TESTLO23 | COREPLLWRDET | VSS | THERMTRIP_N | RSVD |
| AB | XDPDQ[14] | VSS | XDPDQ[15] | XDPDQ[9] | VSS | INTR | TESTLO13 | VSS | PEHPSCL | SMBSCCL | VSS | VCCMISC33 |
| AA | XDPDQ[10] | XDPDQSP[0] | VSS | XDPDQ[11] | RSVD | VSS | QPIFREQSEL1 | QPICKFALL | VSS | RSVD | TESTLO24 | VSS |
| Y | VSS | XDPDQSN[0] | XDPDQ[4] | VSS | RSVD | XDPRDYACK_N | VSS | VRMEN | TESTLO15 | VSS | VCC | VCC |
| W | XDPDQ[1] | VSS | XDPDQ[6] | XDPDQ[0] | VSS | XDPRDYREQ_N | RSVD | VSS | RSVD | TESTLO16 | VSS | VSS |
| V | XDPDQ[3] | XDPDQ[5] | VSS | XDPDQ[2] | TESTLO17 | VSS | TDO | VCCEPW | VSS | RSVD | TDI | VSS |
| U | VSS | XDPDQ[7] | CLCLK | VSS | CLRST_N | TMS | VSS | DDRFREQ[2] | TESTHI[2] | VSS | VCCXDP18 | VTTXDP |
| T | VREFCL | VSS | CLDATA | TCK | VSS | DDRFREQ[3] | PEWIDTH[1] | VSS | PEWIDTH[4] | TESTLO18 | VSS | VCCXDP18 |
| R | PEWIDTH[2] | TESTLO19 | VSS | PEWIDTH[0] | LEGACYIOH | VSS | TRST_N | TESTHI[3] | VSS | VCCCLPWRP | VCCEPW | VSS |
| P | VSS | DDRD[1] | DDRD[3] | VSS | DDRD[7] | DUALIOH | VSS | FWAGNT_ESIMODE | PEWIDTH[5] | VSS | VCCDDR18 | VCCDDR18 |
| N | VCCADDRPLL | VSS | DDRD[2] | DDRD[5] | VSS | DDREDQSN | DDREDQSP | VSS | PEWIDTH[3] | ME_CLK_SRC | VSS | VCCDDR18 |
| M | DDRD[4] | DDRDM | VSS | DDRDM_N | DDRD[0] | VSS | DDRD[6] | DDRDRVCRES | VSS | DDRRES[0] | VCCDDR18 | VSS |
| L | VSS | RSVD | RSVD | VSS | RSVD | RSVD | VSS | DDRRCRES | DDRSLEWCRRES | VSS | RSVD | VCCDDR18 |
| K | DDRBA[1] | VSS | DDRCLKN | DDRCLKP | VSS | DDRBA[2] | DDRRES[1] | VSS | DDRCOMPX | RSVD | VSS | RSVD |



Figure 22-2. IOH Ballout Left Side (Top View) (Sheet 2 of 2)

| | 36 | 35 | 34 | 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 |
|---|----------|---------------|-------------------|-------------------|---------------|---------------|-------------------|-------------------|-------------------|-------------------|--------------------|-------------------|
| J | DDRA[14] | DDRA[12] | VSS | DDRBA[0] | DDRA[13] | VSS | DDRA[2] | DDRA[7] | VSS | RMIITXD[1] | RMIICLKRE FOUT | VSS |
| H | VSS | RSVD | RSVD | VSS | DDRA[4] | DDRA[5] | VSS | DDRA[11] | DDRCAS_N | VSS | QPI1VRMV REFRX1 | VSS |
| G | DDRA[1] | VSS | DDRPLLRE FCLKN | DDRPLLRE FCLKP | VSS | DDRA[8] | DDRA[10] | VSS | RMIICLK | RMIITXEN | VSS | QPI1RNDAT T[1] |
| F | DDRA[0] | DDRA[3] | VSS | DDRA[9] | DDRWE_N | VSS | DDRODT | DDRRAS_N | VSS | VSS | QPI1RNDAT T[2] | QPI1RPDAT T[2] |
| E | VSS | DDRA[6] | DDRCAS_N | VSS | DDRCCKE | RMIITXD[0] | VSS | RMIIRXD[1] | VSS | QPI1RNDAT T[3] | QPI1RPDAT T[3] | VSS |
| D | TESTLO26 | VSS | COREPWR GOOD | CORERST_N | VSS | RMIIMDIO | RMIIMDC | VSS | QPI1RPDAT T[5] | QPI1RNDAT T[5] | VSS | QPI1RNDAT T[4] |
| C | VSS | PLLPCRDE T | VSS | TESTLO22 | AUXPWG OOD | VSS | VSS | VCCQPI1 VRMRX1 | VCCQPI1V RMRX2 | VSS | QPI1RPDAT T[8] | QPI1RNDAT T[8] |
| B | VSS | VSS | VSS | VSS | RMIIRXD[0] | VSS | VCCQPI1V RMRX0 | VCCQPI1 VRMRX3 | VSS | QPI1RPDAT T[7] | QPI1RNDAT T[7] | VSS |
| A | TEST[1] | VSS | VSS | VSS | VSS | RMIICRS DV | VCCAQPI1 RXBG | VSS | QPI1RPDAT T[6] | QPI1RNDAT T[6] | VSS | QPI1RPDAT T[9] |
| | 36 | 35 | 34 | 33 | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 |



Figure 22-3. IOH Ballout Center Side (Top View)

| | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 |
|----|--------------|----------------|-----------------|-----------------|---------------|---------------|---------------|----------------|-----------------|---------------|---------------|---------------|
| AT | RSVD | RSVD | VCCDPE1PLL | VSS | VCCAPE1PLL | VCCAPE1BLG | VCCPEVRM | PE2TP[1] | VSS | PE6TN[1] | PE6TP[1] | PEOJCLKP |
| AR | PE8TP[1] | PE7TP[3] | VSS | ESITP[2] | ESITN[2] | ESITN[0] | VSS | PE2TN[1] | PE1TN[1] | PE1TP[1] | PE6TN[2] | VSS |
| AP | VSS | PE7TN[3] | VSS | RSVD | VSS | ESITP[0] | PE2TN[0] | PE2TP[0] | PE1TN[0] | VSS | PE6TP[2] | PE5TN[3] |
| AN | PE8TP[0] | PE8TN[0] | VSS | ESITN[3] | ESITN[1] | ESITP[1] | RSVD | VSS | PE1TP[0] | PE6TN[0] | PE6TP[0] | PEORBIAS |
| AM | RSVD | VSS | RSVD | ESITP[3] | RSVD | VSS | PE2RP[1] | PE2RN[1] | PE6TN[3] | PE6TP[3] | VSS | PE5RN[2] |
| AL | PE7RP[2] | PE7RN[2] | VSS | VSS | ESIRN[1] | ESIRP[0] | ESIRN[0] | PE2RN[0] | VSS | PE6RN[1] | PE5RP[3] | PE5RN[3] |
| AK | PE7RN[3] | PE7RP[1] | VSS | ESIRN[2] | ESIRP[1] | ESIRP[3] | VSS | PE2RP[0] | PE1RP[0] | PE6RP[1] | PE6RP[2] | VSS |
| AJ | VSS | PE7RN[1] | VSS | ESIRP[2] | VSS | ESIRN[3] | PE1RN[1] | PE1RP[1] | PE1RN[0] | VSS | PE6RN[2] | PE6RP[3] |
| AH | PE7RN[0] | PE7RP[0] | VSS | RSVD | RSVD | RSVD | RSVD | VSS | RSVD | RSVD | RSVD | RSVD |
| AG | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| AF | VSS | VCCAPE1 | VCCAPE1 | VCCAPE1 | VCCAPE1 | VCCAPE | VSS | VCCAPE | VSS | VCCAPE | VSS | VCCAPE |
| AE | VCCAPE1 | VCCAPE1 | VCCAPE1 | VCCAPE1 | VCCAPE1 | VCCAPE | VCCAPE | VCCAPE | VCCAPE | VCCAPE | VCCAPE | VCCAPE |
| AD | VCCAPE1 | VCCAPE1 | VCCAPE1 | VCCAPE1 | VCCAPE1 | VCCAPE | VCCAPE | VCCAPE | VCCAPE | VCCAPE | VCCAPE | VCCAPE |
| AC | VSS | VSS | VCC | VSS | VCC | VCCAPE | VSS | VCCAPE | VSS | VCCAPE | VSS | VCCAPE |
| AB | VCCMISC33 | VCC | VSS | VCC | VSS | VCC | VSS | VCC | VSS | VCC | VSS | VSS |
| AA | VCC | VSS | VCC | VSS | VCC | VSS | VCC | VSS | VCC | VCCAQPI0 | VCCAQPI0 | VCCAQPI0 |
| Y | VSS | VCC | VSS | VCC | VSS | VCC | VSS | VCC | VSS | VCCAQPI0 | VCCAQPI0 | VSS |
| W | VCC | VSS | VCC | VSS | VCC | VSS | VCC | VSS | VCC | VCCAQPI0 | VCCAQPI0 | VCCAQPI0 |
| V | VCC | VCC | VSS | VCC | VSS | VCC | VSS | VCC | VSS | VCCAQPI0 | VCCAQPI0 | VSS |
| U | VSS | VSS | VCC | VSS | VCC | VSS | VCC | VSS | VCC | VCCAQPI0 | VCCAQPI0 | VCCAQPI0 |
| T | VCCEPW | VCCEPW | VSS | VCCEPW | VSS | VCC | VSS | VCC | VSS | VCCAQPI0 | VCCAQPI0 | VSS |
| R | VTTDDR | VSS | VCCEPW | VSS | VCC | VSS | VCC | VSS | VCC | VSS | VCCAQPI0 | VCCAQPI0 |
| P | VSS | VCCDDR18 | VSS | VCCEPW | VSS | VCC | VSS | VCC | VSS | VCC | VSS | VCCAQPI0 |
| N | VCCDDR18 | VSS | VCCEPW | VSS | VSS | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 |
| M | VCCDDR18 | VCCDDR18 | VSS | VCCMISC33EPW | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 | VCCAQPI1 |
| L | VSS | VSS | VCCEPW | VSS | VCCAQPI1 | VCCAQPI1 | VSS | VCCAQPI1 | VSS | VCCAQPI1 | VSS | VCCAQPI1 |
| K | RSVD | VCCEPW | VCCEPW | VCCMISC33EPW | VCCMISC33EPW | VCCAQPI1 | VSS | VCCAQPI1 | VSS | VCCAQPI1 | VSS | RSVD |
| J | RSVD | QPI1VRMVREFRX0 | VSS | RSVD | VSS | RSVD | VSS | VSS | RSVD | VSS | QPI1ICOMP | QPI1RCOMP |
| H | QPI1RNDAT[0] | QPI1RPDAT[0] | VSS | RSVD | RSVD | VSS | RSVD | RSVD | QPI1TPDAT[16] | QPI1TNDAT[16] | VSS | QPI1TXBG[0] |
| G | QPI1RPDAT[1] | VSS | QPI1RXBG[1] | QPI1RXBG[0] | VSS | QPI1RNDAT[18] | QPI1RPDAT[18] | VSS | VSS | QPI1TNDAT[17] | QPI1TPDAT[17] | VSS |
| F | VSS | RSVD | VCCQPI1VRMRXOP1 | VSS | QPI1RNDAT[19] | QPI1RPDAT[19] | VSS | RSVD | QPI1TPDAT[15] | VSS | QPI1TNDAT[18] | QPI1TPDAT[18] |
| E | QPI1RPCLK[0] | QPI1RNCLK[0] | VSS | QPI1RPDAT[17] | QPI1RNDAT[17] | VSS | QPI1RNDAT[16] | QPI1RPDAT[16] | RSVD | QPI1TNDAT[15] | VSS | QPI1TNDAT[19] |
| D | QPI1RPDAT[4] | VSS | RSVD | VCCQPI1VRMRXOP0 | VSS | QPI1RNDAT[15] | QPI1RPDAT[15] | VSS | VSS | QPI1TNDAT[14] | QPI1TPDAT[14] | VSS |
| C | VSS | QPI1RNDAT[11] | QPI1RPDAT[11] | VSS | QPI1RNDAT[14] | QPI1RPDAT[14] | VSS | QPI1VRMVREFRX3 | VCCQPI1VRMRXOP3 | VSS | QPI1TNDAT[13] | QPI1TPDAT[13] |



Figure 22-3. IOH Ballout Center Side (Top View)

| | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 |
|---|------------------|------|-------------------|-------------------|-------------------|-------------------|--------------------|------------------|-------------------|-------------------|-------------------|-------------------|
| B | RSVD | RSVD | VSS | QPI1RNDA T[12] | QPI1RPDAT [12] | VSS | QPI1VRMV REFRX2 | VCCAQPI1 TXBG | QPI1TPDAT [12] | QPI1TNDA T[12] | VSS | QPI1TNDA T[10] |
| A | QPI1RNDA T[9] | VSS | QPI1RPDAT [10] | QPI1RNDA T[10] | VSS | QPI1RPDAT [13] | QPI1RNDA T[13] | VSS | VSS | QPI1TPDAT [11] | QPI1TNDA T[11] | VSS |
| | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13 |



Figure 22-4. IOH Ballout Right (Top View) (Sheet 1 of 2)

| | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|----|----------|----------|-----------------|-----------|---------------|---------------|-----------------|---------------|-----------------|-----------------|---------------|---------------|
| AT | PE0JCLKN | VSS | PE5TN[1] | PE4TN[3] | PE4TP[3] | PE4TP[1] | VSS | PE3TN[3] | PE3TP[3] | VSS | VSS | TEST[3] |
| AR | TESTLO1 | PE5TP[2] | PE5TP[1] | PE5TN[0] | VSS | PE4TN[1] | PE4TP[0] | PE4TN[0] | PE3TN[1] | VSS | VSS | VSS |
| AP | PE5TP[3] | PE5TN[2] | VSS | PE5TP[0] | PE4TP[2] | PE4TN[2] | PE3TN[2] | VSS | PE3TP[1] | PE0ICOMPI | VCCDPEPLL | VSS |
| AN | VSS | PE0CLKP | PE0CLKN | TESTLO2 | TESTLO3 | VSS | PE3TP[2] | PE3TP[0] | PE3TN[0] | VSS | VCCAPEBG | VCCAPEPLL |
| AM | PE5RP[2] | RSVD | RSVD | VSS | PE4RP[1] | PE4RN[1] | TESTLO4 | RSVD | VSS | RSVD | RSVD | PE0RCOMPO |
| AL | PE6RP[0] | VSS | PE5RP[0] | PE4RP[2] | PE4RN[2] | PE3RP[3] | VSS | PE3RN[2] | PE3RN[1] | PE3RP[1] | RESETO_N | PE0ICOMPO |
| AK | PE6RN[0] | PE5RN[1] | PE5RN[0] | PE4RP[3] | VSS | PE3RN[3] | PE3RP[0] | PE3RP[2] | VSS | RSVD | VCCTS | VSS |
| AJ | PE6RN[3] | PE5RP[1] | VSS | PE4RN[3] | PE4RP[0] | PE4RN[0] | PE3RN[0] | VSS | QPI0TNDAT[2] | TSIREF | VSS | QPI0TNDAT[4] |
| AH | VSS | RSVD | RSVD | RSVD | RSVD | VSS | VSS | QPI0TPDAT[1] | QPI0TPDAT[2] | VSS | QPI0TNDAT[5] | QPI0TPDAT[4] |
| AG | VSS | VSS | VSS | RSVD | RSVD | VSS | QPI0TNDAT[0] | QPI0TNDAT[1] | VSS | QPI0TNDAT[3] | QPI0TPDAT[5] | VSS |
| AF | VCCAPE | VSS | RSVD | RSVD | VSS | RSVD | QPI0TPDAT[0] | VSS | QPI0TNDAT[6] | QPI0TPDAT[3] | VSS | QPI0TNDAT[7] |
| AE | VCCAPE | VSS | RSVD | VSS | RSVD | RSVD | VSS | VCCQPI0VRMTX | QPI0TPDAT[6] | VSS | QPI0TNDAT[8] | QPI0TPDAT[7] |
| AD | VCCAPE | VSS | RSVD | RSVD | RSVD | VSS | VCCQPI0VRMRXOP2 | VCCAQPI0PLL | VSS | QPI0TNDAT[9] | QPI0TPDAT[8] | VSS |
| AC | VCCAPE | VSS | VCCQPI0VRMTXOP0 | RSVD | VSS | QPI0REFCLKN | RSVD | VSS | RSVD | QPI0TPDAT[9] | VSS | QPI0TNCLK[0] |
| AB | VSS | VSS | QPI0VRMVREF4 | VSS | QPI0TXBG[1] | QPI0REFCLKP | VSS | QPI0TPDAT[19] | RSVD | VSS | QPI0TPDAT[10] | QPI0TPCLK[0] |
| AA | VCCAQPI0 | VSS | VSS | QPI0RCOMP | QPI0TXBG[0] | VSS | QPI0TPDAT[18] | QPI0TNDAT[19] | VSS | QPI0TPDAT[13] | QPI0TNDAT[10] | VSS |
| Y | VSS | VSS | RSVD | QPI0ICOMP | VSS | QPI0TPDAT[17] | QPI0TNDAT[18] | VSS | QPI0TPDAT[14] | QPI0TNDAT[13] | VSS | QPI0TNDAT[11] |
| W | VCCAQPI0 | VCCAQPI0 | RSVD | VSS | QPI0TNDAT[16] | QPI0TNDAT[17] | VSS | QPI0TNDAT[15] | QPI0TNDAT[14] | VSS | QPI0TNDAT[12] | QPI0TPDAT[11] |
| V | VSS | VSS | VSS | RSVD | QPI0TPDAT[16] | VSS | QPI0TPDAT[15] | RSVD | VSS | VCCQPI0VRMRXOP3 | QPI0TPDAT[12] | VSS |
| U | VCCAQPI0 | VCCAQPI0 | RSVD | RSVD | RSVD | VSS | RSVD | QPI0RPDAT[16] | VSS | QPI0VRMVREF3 | VCCAQPI0TXBG | VSS |
| T | VSS | VSS | RSVD | VSS | RSVD | QPI0RPDAT[18] | VSS | QPI0RNDAT[16] | QPI0RPDAT[15] | VSS | QPI0VRMVREF2 | QPI0RNDAT[13] |
| R | VCCAQPI0 | VSS | VSS | RSVD | VSS | QPI0RNDAT[18] | QPI0RPDAT[19] | VSS | QPI0RNDAT[15] | QPI0RPDAT[14] | VSS | QPI0RPDAT[13] |
| P | VCCAQPI0 | VSS | RSVD | RSVD | RSVD | VSS | QPI0RNDAT[19] | QPI0RNDAT[17] | VSS | QPI0RNDAT[14] | QPI0RPDAT[12] | VSS |
| N | VCCAQPI0 | RSVD | VSS | VSS | RSVD | QPI0RXBG[0] | VSS | QPI0RPDAT[17] | VCCQPI0VRMRXOP0 | VSS | QPI0RNDAT[12] | QPI0RNDAT[10] |



Figure 22-4. IOH Ballout Right (Top View) (Sheet 2 of 2)

| | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |
|---|---------------|---------------|-----------------|--------------|--------------|--------------|-----------------|--------------|--------------|---------------|---------------|---------------|
| M | VCCAQPI1 | QPIOVRMVREF1 | VSS | RSVD | VSS | QPIORXBG[1] | VCCQPI0VRMRXOP1 | VSS | RSVD | QPIORPDAT[11] | VSS | QPIORPDAT[10] |
| L | VSS | VSS | QPIOVRMVREF0 | RSVD | QPIORPDAT[0] | VSS | RSVD | QPIORNCLK[0] | VSS | QPIORNDAT[11] | RSVD | VSS |
| K | RSVD | QPI1VRMVREFTX | VCCQPI1VRMTXOPO | VSS | QPIORNDAT[0] | QPIORPDAT[1] | VSS | QPIORPCLK[0] | QPIORPDAT[4] | VSS | RSVD | QPIORNDAT[9] |
| J | VSS | RSVD | RSVD | VSS | VSS | QPIORNDAT[1] | QPIORPDAT[2] | VSS | QPIORNDAT[4] | QPIORNDAT[8] | VSS | QPIORPDAT[9] |
| H | QPI1TXBG[1] | VSS | RSVD | RSVD | VSS | VSS | QPIORNDAT[2] | QPIORPDAT[3] | VSS | QPIORPDAT[8] | QPIORNDAT[7] | VSS |
| G | QPI1REFCLKP | QPI1REFCLKN | VSS | RSVD | RSVD | VSS | VSS | QPIORNDAT[3] | QPIORNDAT[5] | VSS | QPIORPDAT[7] | QPIORNDAT[6] |
| F | VSS | RSVD | VCCQPI1VRMRXOP2 | VSS | QPI1TPDAT[0] | QPI1TNDAT[0] | VSS | VSS | QPIORPDAT[5] | VCCQPI0VRMRX2 | VSS | QPIORPDAT[6] |
| E | QPI1TPDAT[19] | VSS | VCCAQPI1PLL | VCCQPI1VRMTX | VSS | QPI1TNDAT[1] | QPI1TPDAT[1] | VSS | VSS | VCCQPI0VRMRX1 | VCCQPI0VRMRX3 | VSS |
| D | RSVD | RSVD | VSS | QPI1TPDAT[6] | QPI1TNDAT[6] | VSS | QPI1TPDAT[2] | QPI1TNDAT[2] | VSS | VSS | VCCQPI0VRMRX0 | VCCAQPI0RXBG |
| C | VSS | QPI1TPDAT[9] | QPI1TNDAT[9] | VSS | QPI1TPDAT[3] | QPI1TNDAT[3] | VSS | RSVD | RSVD | VSS | VSS | VSS |
| B | QPI1TPDAT[10] | VSS | QPI1TPDAT[8] | QPI1TNDAT[8] | VSS | QPI1TPDAT[5] | QPI1TNDAT[5] | VSS | RSVD | VSS | VSS | TEST[2] |
| A | QPI1TPCLK[0] | QPI1TNDCLK[0] | VSS | QPI1TPDAT[7] | QPI1TNDAT[7] | VSS | QPI1TPDAT[4] | QPI1TNDAT[4] | VSS | VSS | TEST[0] | |
| | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 |



**Table 22-1. Pin Listing by Pin Name
(Sheet 1 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AT36 | TEST[4] | No Connect | I/O |
| AT35 | VSS | Analog | PWR |
| AT34 | VSS | Analog | PWR |
| AT33 | VSS | Analog | PWR |
| AT32 | PE10TN[0] | PCIEX2 | O |
| AT31 | PE10TP[0] | PCIEX2 | O |
| AT30 | VSS | Analog | PWR |
| AT29 | PE9TN[1] | PCIEX2 | O |
| AT28 | PE9TP[1] | PCIEX2 | O |
| AT27 | PE8TN[3] | PCIEX2 | O |
| AT26 | PE8TP[3] | PCIEX2 | O |
| AT25 | VSS | Analog | PWR |
| AT24 | RSVD | No Connect | |
| AT23 | RSVD | No Connect | |
| AT22 | VCCDPE1PLL | Analog | PWR |
| AT21 | VSS | Analog | PWR |
| AT20 | VCCAPE1PLL | Analog | PWR |
| AT19 | VCCAPE1BG | Analog | PWR |
| AT18 | VCCPEVRM | Analog | PWR |
| AT17 | PE2TP[1] | PCIEX2 | O |
| AT16 | VSS | Analog | PWR |
| AT15 | PE6TN[1] | PCIEX2 | O |
| AT14 | PE6TP[1] | PCIEX2 | O |
| AT13 | PE0JCLKP | HCSL | I |
| AT12 | PE0JCLKN | HCSL | I |
| AT11 | VSS | Analog | PWR |
| AT10 | PE5TN[1] | PCIEX2 | O |
| AT9 | PE4TN[3] | PCIEX2 | O |
| AT8 | PE4TP[3] | PCIEX2 | O |
| AT7 | PE4TP[1] | PCIEX2 | O |
| AT6 | VSS | Analog | PWR |
| AT5 | PE3TN[3] | PCIEX2 | O |
| AT4 | PE3TP[3] | PCIEX2 | O |
| AT3 | VSS | Analog | PWR |
| AT2 | VSS | Analog | PWR |
| AT1 | TEST[3] | No Connect | I/O |
| AR36 | VSS | Analog | PWR |
| AR35 | VSS | Analog | PWR |
| AR34 | VCCPE1VRM | Analog | PWR |
| AR33 | PE10TP[2] | PCIEX2 | O |

**Table 22-1. Pin Listing by Pin Name
(Sheet 2 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AR32 | VSS | Analog | PWR |
| AR31 | PE10TN[1] | PCIEX2 | O |
| AR30 | PE9TN[2] | PCIEX2 | O |
| AR29 | PE9TP[2] | PCIEX2 | O |
| AR28 | PE7TN[0] | PCIEX2 | O |
| AR27 | VSS | Analog | PWR |
| AR26 | PE8TN[2] | PCIEX2 | O |
| AR25 | PE8TN[1] | PCIEX2 | O |
| AR24 | PE8TP[1] | PCIEX2 | O |
| AR23 | PE7TP[3] | PCIEX2 | O |
| AR22 | VSS | Analog | PWR |
| AR21 | ESITP[2] | PCIEX | O |
| AR20 | ESITN[2] | PCIEX | O |
| AR19 | ESITN[0] | PCIEX | O |
| AR18 | VSS | Analog | PWR |
| AR17 | PE2TN[1] | PCIEX2 | O |
| AR16 | PE1TN[1] | PCIEX2 | O |
| AR15 | PE1TP[1] | PCIEX2 | O |
| AR14 | PE6TN[2] | PCIEX2 | O |
| AR13 | VSS | Analog | PWR |
| AR12 | TESTLO1 | Analog | I/O |
| AR11 | PE5TP[2] | PCIEX2 | O |
| AR10 | PE5TP[1] | PCIEX2 | O |
| AR9 | PE5TN[0] | PCIEX2 | O |
| AR8 | VSS | Analog | PWR |
| AR7 | PE4TN[1] | PCIEX2 | O |
| AR6 | PE4TP[0] | PCIEX2 | O |
| AR5 | PE4TN[0] | PCIEX2 | O |
| AR4 | PE3TN[1] | PCIEX2 | O |
| AR3 | VSS | Analog | PWR |
| AR2 | VSS | Analog | PWR |
| AR1 | VSS | Analog | PWR |
| AP36 | VSS | Analog | PWR |
| AP35 | PE1RCOMPO | Analog | I/O |
| AP34 | VSS | Analog | PWR |
| AP33 | PE10TN[2] | PCIEX2 | O |
| AP32 | PE10TP[3] | PCIEX2 | O |
| AP31 | PE10TP[1] | PCIEX2 | O |
| AP30 | PE9TP[3] | PCIEX2 | O |
| AP29 | VSS | Analog | PWR |



**Table 22-1. Pin Listing by Pin Name
(Sheet 3 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AP28 | PE7TP[0] | PCIEX2 | O |
| AP27 | PE7TP[1] | PCIEX2 | O |
| AP26 | PE8TP[2] | PCIEX2 | O |
| AP25 | PE7TN[2] | PCIEX2 | O |
| AP24 | VSS | Analog | PWR |
| AP23 | PE7TN[3] | PCIEX2 | O |
| AP22 | VSS | Analog | PWR |
| AP21 | RSVD | No Connect | |
| AP20 | VSS | Analog | PWR |
| AP19 | ESITP[0] | PCIEX | O |
| AP18 | PE2TN[0] | PCIEX2 | O |
| AP17 | PE2TP[0] | PCIEX2 | O |
| AP16 | PE1TN[0] | PCIEX2 | O |
| AP15 | VSS | Analog | PWR |
| AP14 | PE6TP[2] | PCIEX2 | O |
| AP13 | PE5TN[3] | PCIEX2 | O |
| AP12 | PE5TP[3] | PCIEX2 | O |
| AP11 | PE5TN[2] | PCIEX2 | O |
| AP10 | VSS | Analog | PWR |
| AP9 | PE5TP[0] | PCIEX2 | O |
| AP8 | PE4TP[2] | PCIEX2 | O |
| AP7 | PE4TN[2] | PCIEX2 | O |
| AP6 | PE3TN[2] | PCIEX2 | O |
| AP5 | VSS | Analog | PWR |
| AP4 | PE3TP[1] | PCIEX2 | O |
| AP3 | PE0ICOMPI | Analog | I/O |
| AP2 | VCCDPEPLL | Analog | PWR |
| AP1 | VSS | Analog | PWR |
| AN36 | PE11COMPO | Analog | I/O |
| AN35 | PE11COMPI | Analog | I/O |
| AN34 | PE1RBIAS | Analog | I/O |
| AN33 | RSVD | No Connect | |
| AN32 | PE10TN[3] | PCIEX2 | O |
| AN31 | VSS | Analog | PWR |
| AN30 | PE9TN[3] | PCIEX2 | O |
| AN29 | PE9TN[0] | PCIEX2 | O |
| AN28 | PE9TP[0] | PCIEX2 | O |
| AN27 | PE7TN[1] | PCIEX2 | O |
| AN26 | VSS | Analog | PWR |
| AN25 | PE7TP[2] | PCIEX2 | O |

**Table 22-1. Pin Listing by Pin Name
(Sheet 4 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AN24 | PE8TP[0] | PCIEX2 | O |
| AN23 | PE8TN[0] | PCIEX2 | O |
| AN22 | VSS | Analog | PWR |
| AN21 | ESITN[3] | PCIEX | O |
| AN20 | ESITN[1] | PCIEX | O |
| AN19 | ESITP[1] | PCIEX | O |
| AN18 | RSVD | No Connect | |
| AN17 | VSS | Analog | PWR |
| AN16 | PE1TP[0] | PCIEX2 | O |
| AN15 | PE6TN[0] | PCIEX2 | O |
| AN14 | PE6TP[0] | PCIEX2 | O |
| AN13 | PEORBIAS | Analog | I/O |
| AN12 | VSS | Analog | PWR |
| AN11 | PE0CLKP | HCSL | I |
| AN10 | PE0CLKN | HCSL | I |
| AN9 | TESTLO2 | Analog | I/O |
| AN8 | TESTLO3 | Analog | I/O |
| AN7 | VSS | Analog | PWR |
| AN6 | PE3TP[2] | PCIEX2 | O |
| AN5 | PE3TP[0] | PCIEX2 | O |
| AN4 | PE3TN[0] | PCIEX2 | O |
| AN3 | VSS | Analog | PWR |
| AN2 | VCCAPEBG | Analog | PWR |
| AN1 | VCCAPEPLL | Analog | PWR |
| AM36 | VSS | Analog | PWR |
| AM35 | RSVD | No Connect | |
| AM34 | PE10RP[1] | PCIEX2 | I |
| AM33 | VSS | Analog | PWR |
| AM32 | VSS | Analog | PWR |
| AM31 | RSVD | No Connect | |
| AM30 | PE1JCLKN | HCSL | I |
| AM29 | PE1JCLKP | HCSL | I |
| AM28 | VSS | Analog | PWR |
| AM27 | RSVD | No Connect | |
| AM26 | PE8RN[0] | PCIEX2 | I |
| AM25 | RSVD | No Connect | |
| AM24 | RSVD | No Connect | |
| AM23 | VSS | Analog | PWR |
| AM22 | RSVD | No Connect | |
| AM21 | ESITP[3] | PCIEX | O |



**Table 22-1. Pin Listing by Pin Name
(Sheet 5 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AM20 | RSVD | No Connect | |
| AM19 | VSS | Analog | PWR |
| AM18 | PE2RP[1] | PCIEX2 | I |
| AM17 | PE2RN[1] | PCIEX2 | I |
| AM16 | PE6TN[3] | PCIEX2 | O |
| AM15 | PE6TP[3] | PCIEX2 | O |
| AM14 | VSS | Analog | PWR |
| AM13 | PE5RN[2] | PCIEX2 | I |
| AM12 | PE5RP[2] | PCIEX2 | I |
| AM11 | RSVD | No Connect | |
| AM10 | RSVD | No Connect | |
| AM9 | VSS | Analog | PWR |
| AM8 | PE4RP[1] | PCIEX2 | I |
| AM7 | PE4RN[1] | PCIEX2 | I |
| AM6 | TESTLO4 | Analog | I/O |
| AM5 | RSVD | No Connect | |
| AM4 | VSS | Analog | PWR |
| AM3 | RSVD | No Connect | |
| AM2 | RSVD | No Connect | |
| AM1 | PE0RCOMPO | Analog | I/O |
| AL36 | FERR_N | GPIO | O |
| AL35 | VSS | Analog | PWR |
| AL34 | PE10RN[1] | PCIEX2 | I |
| AL33 | PE10RP[0] | PCIEX2 | I |
| AL32 | PE9RP[2] | PCIEX2 | I |
| AL31 | PE9RN[2] | PCIEX2 | I |
| AL30 | VSS | Analog | PWR |
| AL29 | PE9RN[0] | PCIEX2 | I |
| AL28 | PE8RP[2] | PCIEX2 | I |
| AL27 | PE8RN[2] | PCIEX2 | I |
| AL26 | PE8RP[0] | PCIEX2 | I |
| AL25 | VSS | Analog | PWR |
| AL24 | PE7RP[2] | PCIEX2 | I |
| AL23 | PE7RN[2] | PCIEX2 | I |
| AL22 | VSS | Analog | PWR |
| AL21 | VSS | Analog | PWR |
| AL20 | ESIRN[1] | PCIEX | I |
| AL19 | ESIRP[0] | PCIEX | I |
| AL18 | ESIRN[0] | PCIEX | I |
| AL17 | PE2RN[0] | PCIEX2 | I |

**Table 22-1. Pin Listing by Pin Name
(Sheet 6 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AL16 | VSS | Analog | PWR |
| AL15 | PE6RN[1] | PCIEX2 | I |
| AL14 | PE5RP[3] | PCIEX2 | I |
| AL13 | PE5RN[3] | PCIEX2 | I |
| AL12 | PE6RP[0] | PCIEX2 | I |
| AL11 | VSS | Analog | PWR |
| AL10 | PE5RP[0] | PCIEX2 | I |
| AL9 | PE4RP[2] | PCIEX2 | I |
| AL8 | PE4RN[2] | PCIEX2 | I |
| AL7 | PE3RP[3] | PCIEX2 | I |
| AL6 | VSS | Analog | PWR |
| AL5 | PE3RN[2] | PCIEX2 | I |
| AL4 | PE3RN[1] | PCIEX2 | I |
| AL3 | PE3RP[1] | PCIEX2 | I |
| AL2 | RESETO_N | GPIO | I/O |
| AL1 | PE0ICOMPO | Analog | I/O |
| AK36 | EXTSYSTRIG | GPIO | I/O |
| AK35 | PESBLCSEL | GPIO | I |
| AK34 | VSS | Analog | PWR |
| AK33 | PE10RN[0] | PCIEX2 | I |
| AK32 | PE10RN[2] | PCIEX2 | I |
| AK31 | PE9RP[1] | PCIEX2 | I |
| AK30 | PE9RN[1] | PCIEX2 | I |
| AK29 | PE9RP[0] | PCIEX2 | I |
| AK28 | PE9RN[3] | PCIEX2 | I |
| AK27 | VSS | Analog | PWR |
| AK26 | VSS | Analog | PWR |
| AK25 | PE7RP[3] | PCIEX2 | I |
| AK24 | PE7RN[3] | PCIEX2 | I |
| AK23 | PE7RP[1] | PCIEX2 | I |
| AK22 | VSS | Analog | PWR |
| AK21 | ESIRN[2] | PCIEX | I |
| AK20 | ESIRP[1] | PCIEX | I |
| AK19 | ESIRP[3] | PCIEX | I |
| AK18 | VSS | Analog | PWR |
| AK17 | PE2RP[0] | PCIEX2 | I |
| AK16 | PE1RP[0] | PCIEX2 | I |
| AK15 | PE6RP[1] | PCIEX2 | I |
| AK14 | PE6RP[2] | PCIEX2 | I |
| AK13 | VSS | Analog | PWR |



**Table 22-1. Pin Listing by Pin Name
(Sheet 7 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AK12 | PE6RN[0] | PCIEX2 | I |
| AK11 | PE5RN[1] | PCIEX2 | I |
| AK10 | PE5RN[0] | PCIEX2 | I |
| AK9 | PE4RP[3] | PCIEX2 | I |
| AK8 | VSS | Analog | PWR |
| AK7 | PE3RN[3] | PCIEX2 | I |
| AK6 | PE3RP[0] | PCIEX2 | I |
| AK5 | PE3RP[2] | PCIEX2 | I |
| AK4 | VSS | Analog | PWR |
| AK3 | RSVD | No Connect | |
| AK2 | VCCTS | Analog | PWR |
| AK1 | VSS | Analog | PWR |
| AJ36 | VSS | Analog | PWR |
| AJ35 | QPISBLCESEL | GPIO | I |
| AJ34 | TESTLO5 | GPIO | I |
| AJ33 | VSS | Analog | PWR |
| AJ32 | PE10RP[2] | PCIEX2 | I |
| AJ31 | VSS | Analog | PWR |
| AJ30 | PE10RP[3] | PCIEX2 | I |
| AJ29 | VSS | Analog | PWR |
| AJ28 | PE9RP[3] | PCIEX2 | I |
| AJ27 | PE8RP[1] | PCIEX2 | I |
| AJ26 | PE8RN[1] | PCIEX2 | I |
| AJ25 | PE8RP[3] | PCIEX2 | I |
| AJ24 | VSS | Analog | PWR |
| AJ23 | PE7RN[1] | PCIEX2 | I |
| AJ22 | VSS | Analog | PWR |
| AJ21 | ESIRP[2] | PCIEX | I |
| AJ20 | VSS | Analog | PWR |
| AJ19 | ESIRN[3] | PCIEX | I |
| AJ18 | PE1RN[1] | PCIEX2 | I |
| AJ17 | PE1RP[1] | PCIEX2 | I |
| AJ16 | PE1RN[0] | PCIEX2 | I |
| AJ15 | VSS | Analog | PWR |
| AJ14 | PE6RN[2] | PCIEX2 | I |
| AJ13 | PE6RP[3] | PCIEX2 | I |
| AJ12 | PE6RN[3] | PCIEX2 | I |
| AJ11 | PE5RP[1] | PCIEX2 | I |
| AJ10 | VSS | Analog | PWR |
| AJ9 | PE4RN[3] | PCIEX2 | I |

**Table 22-1. Pin Listing by Pin Name
(Sheet 8 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|--------------|--------------------|-----------|
| AJ8 | PE4RP[0] | PCIEX2 | I |
| AJ7 | PE4RN[0] | PCIEX2 | I |
| AJ6 | PE3RN[0] | PCIEX2 | I |
| AJ5 | VSS | Analog | PWR |
| AJ4 | QPIOTNDAT[2] | Intel QPI | O |
| AJ3 | TSIREF | Analog | I |
| AJ2 | VSS | Analog | PWR |
| AJ1 | QPIOTNDAT[4] | Intel QPI | O |
| AH36 | XDCLK1XP | DDR | O |
| AH35 | VSS | Analog | PWR |
| AH34 | NODEID[2] | GPIO | I |
| AH33 | TESTHI[4] | GPIO | I |
| AH32 | VSS | Analog | PWR |
| AH31 | RSVD | No Connect | |
| AH30 | PE10RN[3] | PCIEX2 | I |
| AH29 | RSVD | No Connect | |
| AH28 | RSVD | No Connect | |
| AH27 | VSS | Analog | PWR |
| AH26 | RSVD | No Connect | |
| AH25 | PE8RN[3] | PCIEX2 | I |
| AH24 | PE7RN[0] | PCIEX2 | I |
| AH23 | PE7RP[0] | PCIEX2 | I |
| AH22 | VSS | Analog | PWR |
| AH21 | RSVD | No Connect | |
| AH20 | RSVD | No Connect | |
| AH19 | RSVD | No Connect | |
| AH18 | RSVD | No Connect | |
| AH17 | VSS | Analog | PWR |
| AH16 | RSVD | No Connect | |
| AH15 | RSVD | No Connect | |
| AH14 | RSVD | No Connect | |
| AH13 | RSVD | No Connect | |
| AH12 | VSS | Analog | PWR |
| AH11 | RSVD | No Connect | |
| AH10 | RSVD | No Connect | |
| AH9 | RSVD | No Connect | |
| AH8 | RSVD | No Connect | |
| AH7 | VSS | Analog | PWR |
| AH6 | VSS | Analog | PWR |
| AH5 | QPIOTPDAT[1] | Intel QPI | O |



**Table 22-1. Pin Listing by Pin Name
(Sheet 9 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|------------------|--------------------|-----------|
| AH4 | QPI0TPDAT[2] | Intel QPI | O |
| AH3 | VSS | Analog | PWR |
| AH2 | QPI0TNDAT[5] | Intel QPI | O |
| AH1 | QPI0TPDAT[4] | Intel QPI | O |
| AG36 | XDPCLK1XN | DDR | O |
| AG35 | QPIFREQSELO | GPIO | I |
| AG34 | VSS | Analog | PWR |
| AG33 | DUALIOH_QPIPRSEL | GPIO | I |
| AG32 | SMI_N | GPIO | I |
| AG31 | VSS | Analog | PWR |
| AG30 | RSVD | No Connect | |
| AG29 | PE1CLKN | HCSL | I |
| AG28 | VSS | Analog | PWR |
| AG27 | RSVD | No Connect | |
| AG26 | RSVD | No Connect | |
| AG25 | VSS | Analog | PWR |
| AG24 | VSS | Analog | PWR |
| AG23 | VSS | Analog | PWR |
| AG22 | VSS | Analog | PWR |
| AG21 | VSS | Analog | PWR |
| AG20 | VSS | Analog | PWR |
| AG19 | VSS | Analog | PWR |
| AG18 | VSS | Analog | PWR |
| AG17 | VSS | Analog | PWR |
| AG16 | VSS | Analog | PWR |
| AG15 | VSS | Analog | PWR |
| AG14 | VSS | Analog | PWR |
| AG13 | VSS | Analog | PWR |
| AG12 | VSS | Analog | PWR |
| AG11 | VSS | Analog | PWR |
| AG10 | VSS | Analog | PWR |
| AG9 | RSVD | No Connect | |
| AG8 | RSVD | No Connect | |
| AG7 | VSS | Analog | PWR |
| AG6 | QPI0TNDAT[0] | Intel QPI | O |
| AG5 | QPI0TNDAT[1] | Intel QPI | O |
| AG4 | VSS | Analog | PWR |
| AG3 | QPI0TNDAT[3] | Intel QPI | O |
| AG2 | QPI0TPDAT[5] | Intel QPI | O |
| AG1 | VSS | Analog | PWR |

**Table 22-1. Pin Listing by Pin Name
(Sheet 10 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|--------------|--------------------|-----------|
| AF36 | VSS | Analog | PWR |
| AF35 | TESTLO8 | GPIO | I |
| AF34 | TESTLO9 | GPIO | I |
| AF33 | VSS | Analog | PWR |
| AF32 | TESTLO10 | GPIO | I |
| AF31 | BMCINIT | GPIO | I |
| AF30 | VSS | Analog | PWR |
| AF29 | PE1CLKP | HCSL | I |
| AF28 | ERR_N[1] | GPIO | O |
| AF27 | VSS | Analog | PWR |
| AF26 | RSVD | No Connect | |
| AF25 | VSS | Analog | PWR |
| AF24 | VSS | Analog | PWR |
| AF23 | VCCAPE1 | Analog | PWR |
| AF22 | VCCAPE1 | Analog | PWR |
| AF21 | VCCAPE1 | Analog | PWR |
| AF20 | VCCAPE1 | Analog | PWR |
| AF19 | VCCAPE | Analog | PWR |
| AF18 | VSS | Analog | PWR |
| AF17 | VCCAPE | Analog | PWR |
| AF16 | VSS | Analog | PWR |
| AF15 | VCCAPE | Analog | PWR |
| AF14 | VSS | Analog | PWR |
| AF13 | VCCAPE | Analog | PWR |
| AF12 | VCCAPE | Analog | PWR |
| AF11 | VSS | Analog | PWR |
| AF10 | RSVD | No Connect | |
| AF9 | RSVD | No Connect | |
| AF8 | VSS | Analog | PWR |
| AF7 | RSVD | No Connect | |
| AF6 | QPI0TPDAT[0] | Intel QPI | O |
| AF5 | VSS | Analog | PWR |
| AF4 | QPI0TNDAT[6] | Intel QPI | O |
| AF3 | QPI0TPDAT[3] | Intel QPI | O |
| AF2 | VSS | Analog | PWR |
| AF1 | QPI0TNDAT[7] | Intel QPI | O |
| AE36 | XDPDQ[12] | DDR | I/O |
| AE35 | VSS | Analog | PWR |
| AE34 | TESTLO11 | GPIO | I/O |
| AE33 | TESTLO25 | GPIO | I/O |



**Table 22-1. Pin Listing by Pin Name
(Sheet 11 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|--------------|--------------------|-----------|
| AE32 | VSS | Analog | PWR |
| AE31 | SMBUSID | GPIO | I |
| AE30 | INIT_N | GPIO | I |
| AE29 | VSS | Analog | PWR |
| AE28 | ERR_N[2] | GPIO | O |
| AE27 | LTRESET_N | GPIO | O |
| AE26 | VSS | Analog | PWR |
| AE25 | VCCAPE1 | Analog | PWR |
| AE24 | VCCAPE1 | Analog | PWR |
| AE23 | VCCAPE1 | Analog | PWR |
| AE22 | VCCAPE1 | Analog | PWR |
| AE21 | VCCAPE1 | Analog | PWR |
| AE20 | VCCAPE1 | Analog | PWR |
| AE19 | VCCAPE | Analog | PWR |
| AE18 | VCCAPE | Analog | PWR |
| AE17 | VCCAPE | Analog | PWR |
| AE16 | VCCAPE | Analog | PWR |
| AE15 | VCCAPE | Analog | PWR |
| AE14 | VCCAPE | Analog | PWR |
| AE13 | VCCAPE | Analog | PWR |
| AE12 | VCCAPE | Analog | PWR |
| AE11 | VSS | Analog | PWR |
| AE10 | RSVD | No Connect | |
| AE9 | VSS | Analog | PWR |
| AE8 | RSVD | No Connect | |
| AE7 | RSVD | No Connect | |
| AE6 | VSS | Analog | PWR |
| AE5 | VCCQPI0VRMTX | Analog | I/O |
| AE4 | QPI0TPDAT[6] | Intel QPI | O |
| AE3 | VSS | Analog | PWR |
| AE2 | QPI0TNDAT[8] | Intel QPI | O |
| AE1 | QPI0TPDAT[7] | Intel QPI | O |
| AD36 | XDPDQ[8] | DDR | O |
| AD35 | XDPDQSP[1] | DDR | I/O |
| AD34 | VSS | Analog | PWR |
| AD33 | TESTLO21 | GPIO | I/O |
| AD32 | NMI | GPIO | I |
| AD31 | VSS | Analog | PWR |
| AD30 | PEHPSDA | GPIO | I/O |
| AD29 | SMBSDA | GPIO | I/O |

**Table 22-1. Pin Listing by Pin Name
(Sheet 12 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-----------------|--------------------|-----------|
| AD28 | VSS | Analog | PWR |
| AD27 | ERR_N[0] | GPIO | O |
| AD26 | THERMALERT_N | GPIO | O |
| AD25 | VCCAPE1 | Analog | PWR |
| AD24 | VCCAPE1 | Analog | PWR |
| AD23 | VCCAPE1 | Analog | PWR |
| AD22 | VCCAPE1 | Analog | PWR |
| AD21 | VCCAPE1 | Analog | PWR |
| AD20 | VCCAPE1 | Analog | PWR |
| AD19 | VCCAPE | Analog | PWR |
| AD18 | VCCAPE | Analog | PWR |
| AD17 | VCCAPE | Analog | PWR |
| AD16 | VCCAPE | Analog | PWR |
| AD15 | VCCAPE | Analog | PWR |
| AD14 | VCCAPE | Analog | PWR |
| AD13 | VCCAPE | Analog | PWR |
| AD12 | VCCAPE | Analog | PWR |
| AD11 | VSS | Analog | PWR |
| AD10 | RSVD | No Connect | |
| AD9 | RSVD | No Connect | |
| AD8 | RSVD | No Connect | |
| AD7 | VSS | Analog | PWR |
| AD6 | VCCQPI0VRMRXOP2 | Analog | I/O |
| AD5 | VCCAQPI0PLL | Analog | I/O |
| AD4 | VSS | Analog | PWR |
| AD3 | QPI0TNDAT[9] | Intel QPI | O |
| AD2 | QPI0TPDAT[8] | Intel QPI | O |
| AD1 | VSS | Analog | PWR |
| AC36 | VSS | Analog | PWR |
| AC35 | XDPDQSN[1] | DDR | I/O |
| AC34 | XDPDQ[13] | DDR | I/O |
| AC33 | VSS | Analog | PWR |
| AC32 | TESTLO12 | GPIO | I |
| AC31 | A20M_N | GPIO | I |
| AC30 | VSS | Analog | PWR |
| AC29 | TESTLO23 | GPIO | I/O |
| AC28 | CORELLPWRDET | GPIO | I |
| AC27 | VSS | Analog | PWR |
| AC26 | THERMTRIP_N | GPIO | I/O |
| AC25 | RSVD | No Connect | |



**Table 22-1. Pin Listing by Pin Name
(Sheet 13 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-----------------|--------------------|-----------|
| AC24 | VSS | Analog | PWR |
| AC23 | VSS | Analog | PWR |
| AC22 | VCC | Analog | PWR |
| AC21 | VSS | Analog | PWR |
| AC20 | VCC | Analog | PWR |
| AC19 | VCCAPE | Analog | PWR |
| AC18 | VSS | Analog | PWR |
| AC17 | VCCAPE | Analog | PWR |
| AC16 | VSS | Analog | PWR |
| AC15 | VCCAPE | Analog | PWR |
| AC14 | VSS | Analog | PWR |
| AC13 | VCCAPE | Analog | PWR |
| AC12 | VCCAPE | Analog | PWR |
| AC11 | VSS | Analog | PWR |
| AC10 | VCCQPI0VRMTXOPO | Analog | I/O |
| AC9 | RSVD | No Connect | |
| AC8 | VSS | Analog | PWR |
| AC7 | QPIOREFCLKN | HCSL | I |
| AC6 | RSVD | No Connect | |
| AC5 | VSS | Analog | PWR |
| AC4 | RSVD | No Connect | |
| AC3 | QPI0TPDAT[9] | Intel QPI | O |
| AC2 | VSS | Analog | PWR |
| AC1 | QPI0TNCLK[0] | Intel QPI | O |
| AB36 | XDPDQ[14] | DDR | I/O |
| AB35 | VSS | Analog | PWR |
| AB34 | XDPDQ[15] | DDR | I/O |
| AB33 | XDPDQ[9] | DDR | I/O |
| AB32 | VSS | Analog | PWR |
| AB31 | INTR | GPIO | I |
| AB30 | TESTLO13 | GPIO | I |
| AB29 | VSS | Analog | PWR |
| AB28 | PEHPSCL | GPIO | O |
| AB27 | SMBSCCL | GPIO | I/O |
| AB26 | VSS | Analog | PWR |
| AB25 | VCCMISC33 | Analog | PWR |
| AB24 | VCCMISC33 | Analog | PWR |
| AB23 | VCC | Analog | PWR |
| AB22 | VSS | Analog | PWR |
| AB21 | VCC | Analog | PWR |

**Table 22-1. Pin Listing by Pin Name
(Sheet 14 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| AB20 | VSS | Analog | PWR |
| AB19 | VCC | Analog | PWR |
| AB18 | VSS | Analog | PWR |
| AB17 | VCC | Analog | PWR |
| AB16 | VSS | Analog | PWR |
| AB15 | VCC | Analog | PWR |
| AB14 | VSS | Analog | PWR |
| AB13 | VSS | Analog | PWR |
| AB12 | VSS | Analog | PWR |
| AB11 | VSS | Analog | PWR |
| AB10 | QPI0VRMVREF4 | Cmos | I |
| AB9 | VSS | Analog | PWR |
| AB8 | QPI0TXBG[1] | Analog | I/O |
| AB7 | QPI0REFCLKP | HCSL | I |
| AB6 | VSS | Analog | PWR |
| AB5 | QPI0TPDAT[19] | Intel QPI | O |
| AB4 | RSVD | No Connect | |
| AB3 | VSS | Analog | PWR |
| AB2 | QPI0TPDAT[10] | Intel QPI | O |
| AB1 | QPI0TPCLK[0] | Intel QPI | O |
| AA36 | XDPDQ[10] | DDR | I/O |
| AA35 | XDPDQSP[0] | DDR | I/O |
| AA34 | VSS | Analog | PWR |
| AA33 | XDPDQ[11] | DDR | I/O |
| AA32 | RSVD | No Connect | |
| AA31 | VSS | Analog | PWR |
| AA30 | QPIFREQSEL1 | GPIO | I |
| AA29 | QPICKFAIL | GPIO | I |
| AA28 | VSS | Analog | PWR |
| AA27 | RSVD | No Connect | |
| AA26 | TESTLO24 | GPIO | I/O |
| AA25 | VSS | Analog | PWR |
| AA24 | VCC | Analog | PWR |
| AA23 | VSS | Analog | PWR |
| AA22 | VCC | Analog | PWR |
| AA21 | VSS | Analog | PWR |
| AA20 | VCC | Analog | PWR |
| AA19 | VSS | Analog | PWR |
| AA18 | VCC | Analog | PWR |
| AA17 | VSS | Analog | PWR |



**Table 22-1. Pin Listing by Pin Name
(Sheet 15 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| AA16 | VCC | Analog | PWR |
| AA15 | VCCAQPI0 | Analog | PWR |
| AA14 | VCCAQPI0 | Analog | PWR |
| AA13 | VCCAQPI0 | Analog | PWR |
| AA12 | VCCAQPI0 | Analog | PWR |
| AA11 | VSS | Analog | PWR |
| AA10 | VSS | Analog | PWR |
| AA9 | QPIORCOMP | Analog | I/O |
| AA8 | QPI0TXBG[0] | Analog | I/O |
| AA7 | VSS | Analog | PWR |
| AA6 | QPI0TPDAT[18] | Intel QPI | O |
| AA5 | QPI0TNDAT[19] | Intel QPI | O |
| AA4 | VSS | Analog | PWR |
| AA3 | QPI0TPDAT[13] | Intel QPI | O |
| AA2 | QPI0TNDAT[10] | Intel QPI | O |
| AA1 | VSS | Analog | PWR |
| Y36 | VSS | Analog | PWR |
| Y35 | XDPDQSN[0] | DDR | I/O |
| Y34 | XDPDQ[4] | DDR | I/O |
| Y33 | VSS | Analog | PWR |
| Y32 | RSVD | No Connect | |
| Y31 | XDPRDYACK_N | DDR | O |
| Y30 | VSS | Analog | PWR |
| Y29 | VRMEN | GPIO | I |
| Y28 | TESTLO15 | GPIO | I |
| Y27 | VSS | Analog | PWR |
| Y26 | VCC | Analog | PWR |
| Y25 | VCC | Analog | PWR |
| Y24 | VSS | Analog | PWR |
| Y23 | VCC | Analog | PWR |
| Y22 | VSS | Analog | PWR |
| Y21 | VCC | Analog | PWR |
| Y20 | VSS | Analog | PWR |
| Y19 | VCC | Analog | PWR |
| Y18 | VSS | Analog | PWR |
| Y17 | VCC | Analog | PWR |
| Y16 | VSS | Analog | PWR |
| Y15 | VCCAQPI0 | Analog | PWR |
| Y14 | VCCAQPI0 | Analog | PWR |
| Y13 | VSS | Analog | PWR |

**Table 22-1. Pin Listing by Pin Name
(Sheet 16 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| Y12 | VSS | Analog | PWR |
| Y11 | VSS | Analog | PWR |
| Y10 | RSVD | No Connect | |
| Y9 | QPI0ICOMP | Analog | I/O |
| Y8 | VSS | Analog | PWR |
| Y7 | QPI0TPDAT[17] | Intel QPI | O |
| Y6 | QPI0TNDAT[18] | Intel QPI | O |
| Y5 | VSS | Analog | PWR |
| Y4 | QPI0TPDAT[14] | Intel QPI | O |
| Y3 | QPI0TNDAT[13] | Intel QPI | O |
| Y2 | VSS | Analog | PWR |
| Y1 | QPI0TNDAT[11] | Intel QPI | O |
| W36 | XDPDQ[1] | DDR | I/O |
| W35 | VSS | Analog | PWR |
| W34 | XDPDQ[6] | DDR | I/O |
| W33 | XDPDQ[0] | DDR | I/O |
| W32 | VSS | Analog | PWR |
| W31 | XDPRDYREQ_N | DDR | I |
| W30 | RSVD | No Connect | |
| W29 | VSS | Analog | PWR |
| W28 | RSVD | No Connect | |
| W27 | TESTLO16 | GPIO | I |
| W26 | VSS | Analog | PWR |
| W25 | VSS | Analog | PWR |
| W24 | VCC | Analog | PWR |
| W23 | VSS | Analog | PWR |
| W22 | VCC | Analog | PWR |
| W21 | VSS | Analog | PWR |
| W20 | VCC | Analog | PWR |
| W19 | VSS | Analog | PWR |
| W18 | VCC | Analog | PWR |
| W17 | VSS | Analog | PWR |
| W16 | VCC | Analog | PWR |
| W15 | VCCAQPI0 | Analog | PWR |
| W14 | VCCAQPI0 | Analog | PWR |
| W13 | VCCAQPI0 | Analog | PWR |
| W12 | VCCAQPI0 | Analog | PWR |
| W11 | VCCAQPI0 | Analog | PWR |
| W10 | RSVD | No Connect | |
| W9 | VSS | Analog | PWR |



**Table 22-1. Pin Listing by Pin Name
(Sheet 17 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| W8 | QPIOTNDAT[16] | Intel QPI | O |
| W7 | QPIOTNDAT[17] | Intel QPI | O |
| W6 | VSS | Analog | PWR |
| W5 | QPIOTNDAT[15] | Intel QPI | O |
| W4 | QPIOTNDAT[14] | Intel QPI | O |
| W3 | VSS | Analog | PWR |
| W2 | QPIOTNDAT[12] | Intel QPI | O |
| W1 | QPIOTPDAT[11] | Intel QPI | O |
| V36 | XDPDQ[3] | DDR | I/O |
| V35 | XDPDQ[5] | DDR | I/O |
| V34 | VSS | Analog | PWR |
| V33 | XDPDQ[2] | DDR | I/O |
| V32 | TESTLO17 | GPIO | I |
| V31 | VSS | Analog | PWR |
| V30 | TDO | GPIO | O |
| V29 | VCCEPW | Analog | I/O |
| V28 | VSS | Analog | PWR |
| V27 | RSVD | No Connect | |
| V26 | TDI | GPIO | I |
| V25 | VSS | Analog | PWR |
| V24 | VCC | Analog | PWR |
| V23 | VCC | Analog | PWR |
| V22 | VSS | Analog | PWR |
| V21 | VCC | Analog | PWR |
| V20 | VSS | Analog | PWR |
| V19 | VCC | Analog | PWR |
| V18 | VSS | Analog | PWR |
| V17 | VCC | Analog | PWR |
| V16 | VSS | Analog | PWR |
| V15 | VCCAQPIO | Analog | PWR |
| V14 | VCCAQPIO | Analog | PWR |
| V13 | VSS | Analog | PWR |
| V12 | VSS | Analog | PWR |
| V11 | VSS | Analog | PWR |
| V10 | VSS | Analog | PWR |
| V9 | RSVD | No Connect | |
| V8 | QPIOTPDAT[16] | Intel QPI | O |
| V7 | VSS | Analog | PWR |
| V6 | QPIOTPDAT[15] | Intel QPI | O |
| V5 | RSVD | No Connect | |

**Table 22-1. Pin Listing by Pin Name
(Sheet 18 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-----------------|--------------------|-----------|
| V4 | VSS | Analog | PWR |
| V3 | VCCQPIOVRMRXOP3 | Analog | I/O |
| V2 | QPIOTPDAT[12] | Intel QPI | O |
| V1 | VSS | Analog | PWR |
| U36 | VSS | Analog | PWR |
| U35 | XDPDQ[7] | DDR | I/O |
| U34 | CLCLK | Cmos | I/O |
| U33 | VSS | Analog | PWR |
| U32 | CLRST_N | Cmos | I |
| U31 | TMS | GPIO | I |
| U30 | VSS | Analog | PWR |
| U29 | DDRFREQ[2] | GPIO | I |
| U28 | TESTHI[2] | GPIO | I |
| U27 | VSS | Analog | PWR |
| U26 | VCCXDP18 | Analog | PWR |
| U25 | VTTXDP | Analog | PWR |
| U24 | VSS | Analog | PWR |
| U23 | VSS | Analog | PWR |
| U22 | VCC | Analog | PWR |
| U21 | VSS | Analog | PWR |
| U20 | VCC | Analog | PWR |
| U19 | VSS | Analog | PWR |
| U18 | VCC | Analog | PWR |
| U17 | VSS | Analog | PWR |
| U16 | VCC | Analog | PWR |
| U15 | VCCAQPIO | Analog | PWR |
| U14 | VCCAQPIO | Analog | PWR |
| U13 | VCCAQPIO | Analog | PWR |
| U12 | VCCAQPIO | Analog | PWR |
| U11 | VCCAQPIO | Analog | PWR |
| U10 | RSVD | No Connect | |
| U9 | RSVD | No Connect | |
| U8 | RSVD | No Connect | |
| U7 | VSS | Analog | PWR |
| U6 | RSVD | No Connect | |
| U5 | QPIORPDAT[16] | Intel QPI | I |
| U4 | VSS | Analog | PWR |
| U3 | QPIOVRMVREF3 | Cmos | I |
| U2 | VCCAQPIOTXBG | Analog | I/O |
| U1 | VSS | Analog | PWR |



**Table 22-1. Pin Listing by Pin Name
(Sheet 19 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| T36 | VREFCL | Cmos | I/O |
| T35 | VSS | Analog | PWR |
| T34 | CLDATA | Cmos | I/O |
| T33 | TCK | GPIO | I |
| T32 | VSS | Analog | PWR |
| T31 | DDRFREQ[3] | GPIO | I |
| T30 | PEWIDTH[1] | GPIO | I/O |
| T29 | VSS | Analog | PWR |
| T28 | PEWIDTH[4] | GPIO | I/O |
| T27 | TESTLO18 | GPIO | I |
| T26 | VSS | Analog | PWR |
| T25 | VCCXDP18 | Analog | PWR |
| T24 | VCCEPW | Analog | PWR |
| T23 | VCCEPW | Analog | PWR |
| T22 | VSS | Analog | PWR |
| T21 | VCCEPW | Analog | PWR |
| T20 | VSS | Analog | PWR |
| T19 | VCC | Analog | PWR |
| T18 | VSS | Analog | PWR |
| T17 | VCC | Analog | PWR |
| T16 | VSS | Analog | PWR |
| T15 | VCCAQPI0 | Analog | PWR |
| T14 | VCCAQPI0 | Analog | PWR |
| T13 | VSS | Analog | PWR |
| T12 | VSS | Analog | PWR |
| T11 | VSS | Analog | PWR |
| T10 | RSVD | No Connect | |
| T9 | VSS | Analog | PWR |
| T8 | RSVD | No Connect | |
| T7 | QPIORPDAT[18] | Intel QPI | I |
| T6 | VSS | Analog | PWR |
| T5 | QPIORNDAT[16] | Intel QPI | I |
| T4 | QPIORPDAT[15] | Intel QPI | I |
| T3 | VSS | Analog | PWR |
| T2 | QPIOVRMREF2 | Cmos | I |
| T1 | QPIORNDAT[13] | Intel QPI | I |
| R36 | PEWIDTH[2] | GPIO | I/O |
| R35 | TESTLO19 | GPIO | I/O |
| R34 | VSS | Analog | PWR |
| R33 | PEWIDTH[0] | GPIO | I/O |

**Table 22-1. Pin Listing by Pin Name
(Sheet 20 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|----------------|--------------------|-----------|
| R32 | LEGACYIOH | GPIO | I/O |
| R31 | VSS | Analog | PWR |
| R30 | TRST_N | GPIO | I |
| R29 | TESTHI[3] | GPIO | I |
| R28 | VSS | Analog | PWR |
| R27 | VCCCLPWRP | Analog | I/O |
| R26 | VCCEPW | Analog | PWR |
| R25 | VSS | Analog | PWR |
| R24 | VTTDDR | Analog | PWR |
| R23 | VSS | Analog | PWR |
| R22 | VCCEPW | Analog | PWR |
| R21 | VSS | Analog | PWR |
| R20 | VCC | Analog | PWR |
| R19 | VSS | Analog | PWR |
| R18 | VCC | Analog | PWR |
| R17 | VSS | Analog | PWR |
| R16 | VCC | Analog | PWR |
| R15 | VSS | Analog | PWR |
| R14 | VCCAQPI0 | Analog | PWR |
| R13 | VCCAQPI0 | Analog | PWR |
| R12 | VCCAQPI0 | Analog | PWR |
| R11 | VSS | Analog | PWR |
| R10 | VSS | Analog | PWR |
| R9 | RSVD | No Connect | |
| R8 | VSS | Analog | PWR |
| R7 | QPIORNDAT[18] | Intel QPI | I |
| R6 | QPIORPDAT[19] | Intel QPI | I |
| R5 | VSS | Analog | PWR |
| R4 | QPIORNDAT[15] | Intel QPI | I |
| R3 | QPIORPDAT[14] | Intel QPI | I |
| R2 | VSS | Analog | PWR |
| R1 | QPIORPDAT[13] | Intel QPI | I |
| P36 | VSS | Analog | PWR |
| P35 | DDR[1] | DDR | I/O |
| P34 | DDR[3] | DDR | I/O |
| P33 | VSS | Analog | PWR |
| P32 | DDR[7] | DDR | I/O |
| P31 | DUALIOH | GPIO | I/O |
| P30 | VSS | Analog | PWR |
| P29 | FWAGNT_ESIMODE | GPIO | I |



**Table 22-1. Pin Listing by Pin Name
(Sheet 21 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| P28 | PEWIDTH[5] | GPIO | I/O |
| P27 | VSS | Analog | PWR |
| P26 | VCCDDR18 | Analog | PWR |
| P25 | VCCDDR18 | Analog | PWR |
| P24 | VSS | Analog | PWR |
| P23 | VCCDDR18 | Analog | PWR |
| P22 | VSS | Analog | PWR |
| P21 | VCCEPW | Analog | PWR |
| P20 | VSS | Analog | PWR |
| P19 | VCC | Analog | PWR |
| P18 | VSS | Analog | PWR |
| P17 | VCC | Analog | PWR |
| P16 | VSS | Analog | PWR |
| P15 | VCC | Analog | PWR |
| P14 | VSS | Analog | PWR |
| P13 | VCCAQPI0 | Analog | PWR |
| P12 | VCCAQPI0 | Analog | PWR |
| P11 | VSS | Analog | PWR |
| P10 | RSVD | No Connect | |
| P9 | RSVD | No Connect | |
| P8 | RSVD | No Connect | |
| P7 | VSS | Analog | PWR |
| P6 | QPIORNDAT[19] | Intel QPI | I |
| P5 | QPIORNDAT[17] | Intel QPI | I |
| P4 | VSS | Analog | PWR |
| P3 | QPIORNDAT[14] | Intel QPI | I |
| P2 | QPIORPDAT[12] | Intel QPI | I |
| P1 | VSS | Analog | PWR |
| N36 | VCCADDRPLL | Analog | PWR |
| N35 | VSS | Analog | PWR |
| N34 | DDRD[2] | DDR | I/O |
| N33 | DDRD[5] | DDR | I/O |
| N32 | VSS | Analog | PWR |
| N31 | DDREDQSN | DDR | O |
| N30 | DDREDQSP | DDR | O |
| N29 | VSS | Analog | PWR |
| N28 | PEWIDTH[3] | GPIO | I/O |
| N27 | ME_CLK_SRC | GPIO | I |
| N26 | VSS | Analog | PWR |
| N25 | VCCDDR18 | Analog | PWR |

**Table 22-1. Pin Listing by Pin Name
(Sheet 22 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|----------------|--------------------|-----------|
| N24 | VCCDDR18 | Analog | PWR |
| N23 | VSS | Analog | PWR |
| N22 | VCCEPW | Analog | PWR |
| N21 | VSS | Analog | PWR |
| N20 | VSS | Analog | PWR |
| N19 | VCCAQPI1 | Analog | PWR |
| N18 | VCCAQPI1 | Analog | PWR |
| N17 | VCCAQPI1 | Analog | PWR |
| N16 | VCCAQPI1 | Analog | PWR |
| N15 | VCCAQPI1 | Analog | PWR |
| N14 | VCCAQPI1 | Analog | PWR |
| N13 | VCCAQPI1 | Analog | PWR |
| N12 | VCCAQPI0 | Analog | PWR |
| N11 | RSVD | No Connect | |
| N10 | VSS | Analog | PWR |
| N9 | VSS | Analog | PWR |
| N8 | RSVD | No Connect | |
| N7 | QPI0RXBG[0] | Analog | I/O |
| N6 | VSS | Analog | PWR |
| N5 | QPIORPDAT[17] | Intel QPI | I |
| N4 | VCCQPI0VMRXOP0 | Analog | I/O |
| N3 | VSS | Analog | PWR |
| N2 | QPIORNDAT[12] | Intel QPI | I |
| N1 | QPIORNDAT[10] | Intel QPI | I |
| M36 | DDRD[4] | DDR | I/O |
| M35 | DDRDM | DDR | O |
| M34 | VSS | Analog | PWR |
| M33 | DDRDM_N | DDR | O |
| M32 | DDRD[0] | DDR | I/O |
| M31 | VSS | Analog | PWR |
| M30 | DDRD[6] | DDR | I/O |
| M29 | DDRDRVCRES | Analog | I/O |
| M28 | VSS | Analog | PWR |
| M27 | DDRRES[0] | Analog | I/O |
| M26 | VCCDDR18 | Analog | PWR |
| M25 | VSS | Analog | PWR |
| M24 | VCCDDR18 | Analog | PWR |
| M23 | VCCDDR18 | Analog | PWR |
| M22 | VSS | Analog | PWR |
| M21 | VCCMISC33EPW | Analog | PWR |



**Table 22-1. Pin Listing by Pin Name
(Sheet 23 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-----------------|--------------------|-----------|
| M20 | VCCAQPI1 | Analog | PWR |
| M19 | VCCAQPI1 | Analog | PWR |
| M18 | VCCAQPI1 | Analog | PWR |
| M17 | VCCAQPI1 | Analog | PWR |
| M16 | VCCAQPI1 | Analog | PWR |
| M15 | VCCAQPI1 | Analog | PWR |
| M14 | VCCAQPI1 | Analog | PWR |
| M13 | VCCAQPI1 | Analog | PWR |
| M12 | VCCAQPI1 | Analog | PWR |
| M11 | QPIOVRMVREF1 | Cmos | I |
| M10 | VSS | Analog | PWR |
| M9 | RSVD | No Connect | |
| M8 | VSS | Analog | PWR |
| M7 | QPIORXBG[1] | Analog | I/O |
| M6 | VCCQPIOVRMRXOP1 | Analog | I/O |
| M5 | VSS | Analog | PWR |
| M4 | RSVD | No Connect | |
| M3 | QPIORPDAT[11] | Intel QPI | I |
| M2 | VSS | Analog | PWR |
| M1 | QPIORPDAT[10] | Intel QPI | I |
| L36 | VSS | Analog | PWR |
| L35 | RSVD | No Connect | |
| L34 | RSVD | No Connect | |
| L33 | VSS | Analog | PWR |
| L32 | RSVD | No Connect | |
| L31 | RSVD | No Connect | |
| L30 | VSS | Analog | PWR |
| L29 | DDRCRES | Analog | I/O |
| L28 | DDRSLEWCRES | Analog | I/O |
| L27 | VSS | Analog | PWR |
| L26 | RSVD | No Connect | |
| L25 | VCCDDR18 | Analog | PWR |
| L24 | VSS | Analog | PWR |
| L23 | VSS | Analog | PWR |
| L22 | VCCEPW | Analog | PWR |
| L21 | VSS | Analog | PWR |
| L20 | VCCAQPI1 | Analog | PWR |
| L19 | VCCAQPI1 | Analog | PWR |
| L18 | VSS | Analog | PWR |
| L17 | VCCAQPI1 | Analog | PWR |

**Table 22-1. Pin Listing by Pin Name
(Sheet 24 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| L16 | VSS | Analog | PWR |
| L15 | VCCAQPI1 | Analog | PWR |
| L14 | VSS | Analog | PWR |
| L13 | VCCAQPI1 | Analog | PWR |
| L12 | VSS | Analog | PWR |
| L11 | VSS | Analog | PWR |
| L10 | QPIOVRMVREF0 | Cmos | I |
| L9 | RSVD | No Connect | |
| L8 | QPIORPDAT[0] | Intel QPI | I |
| L7 | VSS | Analog | PWR |
| L6 | RSVD | No Connect | |
| L5 | QPIORNCLK[0] | Intel QPI | I |
| L4 | VSS | Analog | PWR |
| L3 | QPIORNDAT[11] | Intel QPI | I |
| L2 | RSVD | No Connect | |
| L1 | VSS | Analog | PWR |
| K36 | DDRBA[1] | DDR | O |
| K35 | VSS | Analog | PWR |
| K34 | DDRCLKN | DDR | O |
| K33 | DDRCLKP | DDR | O |
| K32 | VSS | Analog | PWR |
| K31 | DDRBA[2] | DDR | O |
| K30 | DDRRES[1] | Analog | I/O |
| K29 | VSS | Analog | PWR |
| K28 | DDRCOMPX | Analog | I/O |
| K27 | RSVD | No Connect | |
| K26 | VSS | Analog | PWR |
| K25 | RSVD | No Connect | |
| K24 | RSVD | No Connect | |
| K23 | VCCEPW | Analog | PWR |
| K22 | VCCEPW | Analog | PWR |
| K21 | VCCMISC33EPW | Analog | PWR |
| K20 | VCCMISC33EPW | Analog | PWR |
| K19 | VCCAQPI1 | Analog | PWR |
| K18 | VSS | Analog | PWR |
| K17 | VCCAQPI1 | Analog | PWR |
| K16 | VSS | Analog | PWR |
| K15 | VCCAQPI1 | Analog | PWR |
| K14 | VSS | Analog | PWR |
| K13 | RSVD | No Connect | |



**Table 22-1. Pin Listing by Pin Name
(Sheet 25 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-----------------|--------------------|-----------|
| K12 | RSVD | No Connect | |
| K11 | QPI1VRMVREFTX | Cmos | I |
| K10 | VCCQPI1VRMTXOP0 | Analog | I/O |
| K9 | VSS | Analog | PWR |
| K8 | QPIORNDAT[0] | Intel QPI | I |
| K7 | QPIORPDAT[1] | Intel QPI | I |
| K6 | VSS | Analog | PWR |
| K5 | QPIORPCLK[0] | Intel QPI | I |
| K4 | QPIORPDAT[4] | Intel QPI | I |
| K3 | VSS | Analog | PWR |
| K2 | RSVD | No Connect | |
| K1 | QPIORNDAT[9] | Intel QPI | I |
| J36 | DDRA[14] | DDR | O |
| J35 | DDRA[12] | DDR | O |
| J34 | VSS | Analog | PWR |
| J33 | DDRBA[0] | DDR | O |
| J32 | DDRA[13] | DDR | O |
| J31 | VSS | Analog | PWR |
| J30 | DDRA[2] | DDR | O |
| J29 | DDRA[7] | DDR | O |
| J28 | VSS | Analog | PWR |
| J27 | RMIITXD[1] | GPIO | O |
| J26 | RMIICLKREFOUT | GPIO | O |
| J25 | VSS | Analog | PWR |
| J24 | RSVD | No Connect | |
| J23 | QPI1VRMVREFRX0 | Cmos | I |
| J22 | VSS | Analog | PWR |
| J21 | RSVD | No Connect | |
| J20 | VSS | Analog | PWR |
| J19 | RSVD | No Connect | |
| J18 | VSS | Analog | PWR |
| J17 | VSS | Analog | PWR |
| J16 | RSVD | No Connect | |
| J15 | VSS | Analog | PWR |
| J14 | QPI1ICOMP | Analog | I/O |
| J13 | QPI1RCOMP | Analog | I/O |
| J12 | VSS | Analog | PWR |
| J11 | RSVD | No Connect | |
| J10 | RSVD | No Connect | |
| J9 | VSS | Analog | PWR |

**Table 22-1. Pin Listing by Pin Name
(Sheet 26 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|----------------|--------------------|-----------|
| J8 | VSS | Analog | PWR |
| J7 | QPIORNDAT[1] | Intel QPI | I |
| J6 | QPIORPDAT[2] | Intel QPI | I |
| J5 | VSS | Analog | PWR |
| J4 | QPIORNDAT[4] | Intel QPI | I |
| J3 | QPIORNDAT[8] | Intel QPI | I |
| J2 | VSS | Analog | PWR |
| J1 | QPIORPDAT[9] | Intel QPI | I |
| H36 | VSS | Analog | PWR |
| H35 | RSVD | No Connect | |
| H34 | RSVD | No Connect | |
| H33 | VSS | Analog | PWR |
| H32 | DDRA[4] | DDR | O |
| H31 | DDRA[5] | DDR | O |
| H30 | VSS | Analog | PWR |
| H29 | DDRA[11] | DDR | O |
| H28 | DDRCAS_N | DDR | O |
| H27 | VSS | Analog | PWR |
| H26 | QPI1VRMVREFRX1 | Cmos | I |
| H25 | VSS | Analog | PWR |
| H24 | QPI1RNDAT[0] | Intel QPI | I |
| H23 | QPI1RPDAT[0] | Intel QPI | I |
| H22 | VSS | Analog | PWR |
| H21 | RSVD | No Connect | |
| H20 | RSVD | No Connect | |
| H19 | VSS | Analog | PWR |
| H18 | RSVD | No Connect | |
| H17 | RSVD | No Connect | |
| H16 | QPI1TPDAT[16] | Intel QPI | O |
| H15 | QPI1TNDAT[16] | Intel QPI | O |
| H14 | VSS | Analog | PWR |
| H13 | QPI1TXBG[0] | Analog | I/O |
| H12 | QPI1TXBG[1] | Analog | I/O |
| H11 | VSS | Analog | PWR |
| H10 | RSVD | No Connect | |
| H9 | RSVD | No Connect | |
| H8 | VSS | Analog | PWR |
| H7 | VSS | Analog | PWR |
| H6 | QPIORNDAT[2] | Intel QPI | I |
| H5 | QPIORPDAT[3] | Intel QPI | I |



**Table 22-1. Pin Listing by Pin Name
(Sheet 27 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| H4 | VSS | Analog | PWR |
| H3 | QPIORPDAT[8] | Intel QPI | I |
| H2 | QPIORNDAT[7] | Intel QPI | I |
| H1 | VSS | Analog | PWR |
| G36 | DDRA[1] | DDR | O |
| G35 | VSS | Analog | PWR |
| G34 | DDRPLLREFCLKN | DDR | I |
| G33 | DDRPLLREFCLKP | DDR | O |
| G32 | VSS | Analog | PWR |
| G31 | DDRA[8] | DDR | O |
| G30 | DDRA[10] | DDR | O |
| G29 | VSS | Analog | PWR |
| G28 | RMIICLK | GPIO | I |
| G27 | RMIITXEN | GPIO | O |
| G26 | VSS | Analog | PWR |
| G25 | QPI1RNDAT[1] | Intel QPI | I |
| G24 | QPI1RPDAT[1] | Intel QPI | I |
| G23 | VSS | Analog | PWR |
| G22 | QPI1RXBG[1] | Analog | I/O |
| G21 | QPI1RXBG[0] | Analog | I/O |
| G20 | VSS | Analog | PWR |
| G19 | QPI1RNDAT[18] | Intel QPI | I |
| G18 | QPI1RPDAT[18] | Intel QPI | I |
| G17 | VSS | Analog | PWR |
| G16 | VSS | Analog | PWR |
| G15 | QPI1TNDAT[17] | Intel QPI | O |
| G14 | QPI1TPDAT[17] | Intel QPI | O |
| G13 | VSS | Analog | PWR |
| G12 | QPI1REFCLKP | HCSL | I |
| G11 | QPI1REFCLKN | HCSL | I |
| G10 | VSS | Analog | PWR |
| G9 | RSVD | No Connect | |
| G8 | RSVD | No Connect | |
| G7 | VSS | Analog | PWR |
| G6 | VSS | Analog | PWR |
| G5 | QPIORNDAT[3] | Intel QPI | I |
| G4 | QPIORNDAT[5] | Intel QPI | I |
| G3 | VSS | Analog | PWR |
| G2 | QPIORPDAT[7] | Intel QPI | I |
| G1 | QPIORNDAT[6] | Intel QPI | I |

**Table 22-1. Pin Listing by Pin Name
(Sheet 28 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-----------------|--------------------|-----------|
| F36 | DDRA[0] | DDR | O |
| F35 | DDRA[3] | DDR | O |
| F34 | VSS | Analog | PWR |
| F33 | DDRA[9] | DDR | O |
| F32 | DDRWE_N | DDR | O |
| F31 | VSS | Analog | PWR |
| F30 | DDRODT | DDR | O |
| F29 | DDRRAS_N | DDR | O |
| F28 | VSS | Analog | PWR |
| F27 | VSS | Analog | PWR |
| F26 | QPI1RNDAT[2] | Intel QPI | I |
| F25 | QPI1RPDAT[2] | Intel QPI | I |
| F24 | VSS | Analog | PWR |
| F23 | RSVD | No Connect | |
| F22 | VCCQPI1VRMRXOP1 | Analog | I/O |
| F21 | VSS | Analog | PWR |
| F20 | QPI1RNDAT[19] | Intel QPI | I |
| F19 | QPI1RPDAT[19] | Intel QPI | I |
| F18 | VSS | Analog | PWR |
| F17 | RSVD | No Connect | |
| F16 | QPI1TPDAT[15] | Intel QPI | O |
| F15 | VSS | Analog | PWR |
| F14 | QPI1TNDAT[18] | Intel QPI | O |
| F13 | QPI1TPDAT[18] | Intel QPI | O |
| F12 | VSS | Analog | PWR |
| F11 | RSVD | No Connect | |
| F10 | VCCQPI1VRMRXOP2 | Analog | I/O |
| F9 | VSS | Analog | PWR |
| F8 | QPI1TPDAT[0] | Intel QPI | O |
| F7 | QPI1TNDAT[0] | Intel QPI | O |
| F6 | VSS | Analog | PWR |
| F5 | VSS | Analog | PWR |
| F4 | QPIORPDAT[5] | Intel QPI | I |
| F3 | VCCQPI10VRMRX2 | Analog | I/O |
| F2 | VSS | Analog | PWR |
| F1 | QPIORPDAT[6] | Intel QPI | I |
| E36 | VSS | Analog | PWR |
| E35 | DDRA[6] | DDR | O |
| E34 | DDRCS_N | DDR | O |
| E33 | VSS | Analog | PWR |



**Table 22-1. Pin Listing by Pin Name
(Sheet 29 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| E32 | DDRCKE | DDR | O |
| E31 | RMIITXD[0] | GPIO | O |
| E30 | VSS | Analog | PWR |
| E29 | RMIIRXD[1] | GPIO | I |
| E28 | VSS | Analog | PWR |
| E27 | QPI1RNDAT[3] | Intel QPI | I |
| E26 | QPI1RPDAT[3] | Intel QPI | I |
| E25 | VSS | Analog | PWR |
| E24 | QPI1RPCLK[0] | Intel QPI | I |
| E23 | QPI1RNCLK[0] | Intel QPI | I |
| E22 | VSS | Analog | PWR |
| E21 | QPI1RPDAT[17] | Intel QPI | I |
| E20 | QPI1RNDAT[17] | Intel QPI | I |
| E19 | VSS | Analog | PWR |
| E18 | QPI1RNDAT[16] | Intel QPI | I |
| E17 | QPI1RPDAT[16] | Intel QPI | I |
| E16 | RSVD | No Connect | |
| E15 | QPI1TNDAT[15] | Intel QPI | I |
| E14 | VSS | Analog | PWR |
| E13 | QPI1TNDAT[19] | Intel QPI | O |
| E12 | QPI1TPDAT[19] | Intel QPI | O |
| E11 | VSS | Analog | PWR |
| E10 | VCCAQPI1PLL | Analog | PWR |
| E9 | VCCQPI1VRMTX | Analog | I/O |
| E8 | VSS | Analog | PWR |
| E7 | QPI1TNDAT[1] | Intel QPI | O |
| E6 | QPI1TPDAT[1] | Intel QPI | O |
| E5 | VSS | Analog | PWR |
| E4 | VSS | Analog | PWR |
| E3 | VCCQPI0VRMRX1 | Analog | I/O |
| E2 | VCCQPI0VRMRX3 | Analog | I/O |
| E1 | VSS | Analog | PWR |
| D36 | TESTLO26 | GPIO | I/O |
| D35 | VSS | Analog | PWR |
| D34 | COREPWRGOOD | GPIO | I |
| D33 | CORERST_N | GPIO | I |
| D32 | VSS | Analog | PWR |
| D31 | RMIIMDIO | GPIO | I/O |
| D30 | RMIIMDC | GPIO | O |
| D29 | VSS | Analog | PWR |

**Table 22-1. Pin Listing by Pin Name
(Sheet 30 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-----------------|--------------------|-----------|
| D28 | QPI1RPDAT[5] | Intel QPI | I |
| D27 | QPI1RNDAT[5] | Intel QPI | I |
| D26 | VSS | Analog | PWR |
| D25 | QPI1RNDAT[4] | Intel QPI | I |
| D24 | QPI1RPDAT[4] | Intel QPI | I |
| D23 | VSS | Analog | PWR |
| D22 | RSVD | No Connect | |
| D21 | VCCQPI1VRMRXOP0 | Analog | I/O |
| D20 | VSS | Analog | PWR |
| D19 | QPI1RNDAT[15] | Intel QPI | I |
| D18 | QPI1RPDAT[15] | Intel QPI | I |
| D17 | VSS | Analog | PWR |
| D16 | VSS | Analog | PWR |
| D15 | QPI1TNDAT[14] | Intel QPI | O |
| D14 | QPI1TPDAT[14] | Intel QPI | O |
| D13 | VSS | Analog | PWR |
| D12 | RSVD | No Connect | |
| D11 | RSVD | No Connect | |
| D10 | VSS | Analog | PWR |
| D9 | QPI1TPDAT[6] | Intel QPI | O |
| D8 | QPI1TNDAT[6] | Intel QPI | O |
| D7 | VSS | Analog | PWR |
| D6 | QPI1TPDAT[2] | Intel QPI | O |
| D5 | QPI1TNDAT[2] | Intel QPI | O |
| D4 | VSS | Analog | PWR |
| D3 | VSS | Analog | PWR |
| D2 | VCCQPI0VRMRX0 | Analog | I/O |
| D1 | VCCAQPI0RXBG | Analog | I/O |
| C36 | VSS | Analog | PWR |
| C35 | PLLWRDET | GPIO | I |
| C34 | VSS | Analog | PWR |
| C33 | TESTLO22 | GPIO | I/O |
| C32 | AUXPWRGOOD | GPIO | I |
| C31 | VSS | Analog | PWR |
| C30 | VSS | Analog | PWR |
| C29 | VCCQPI1VRMRX1 | Analog | I/O |
| C28 | VCCQPI1VRMRX2 | Analog | I/O |
| C27 | VSS | Analog | PWR |
| C26 | QPI1RPDAT[8] | Intel QPI | I |
| C25 | QPI1RNDAT[8] | Intel QPI | I |



**Table 22-1. Pin Listing by Pin Name
(Sheet 31 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-----------------|--------------------|-----------|
| C24 | VSS | Analog | PWR |
| C23 | QPI1RNDAT[11] | Intel QPI | I |
| C22 | QPI1RPDAT[11] | Intel QPI | I |
| C21 | VSS | Analog | PWR |
| C20 | QPI1RNDAT[14] | Intel QPI | I |
| C19 | QPI1RPDAT[14] | Intel QPI | I |
| C18 | VSS | Analog | PWR |
| C17 | QPI1VRMVREFRX3 | Cmos | I |
| C16 | VCCQPI1VRMRXOP3 | Analog | I/O |
| C15 | VSS | Analog | PWR |
| C14 | QPI1TNDAT[13] | Intel QPI | O |
| C13 | QPI1TPDAT[13] | Intel QPI | O |
| C12 | VSS | Analog | PWR |
| C11 | QPI1TPDAT[9] | Intel QPI | O |
| C10 | QPI1TNDAT[9] | Intel QPI | O |
| C9 | VSS | Analog | PWR |
| C8 | QPI1TPDAT[3] | Intel QPI | O |
| C7 | QPI1TNDAT[3] | Intel QPI | O |
| C6 | VSS | Analog | PWR |
| C5 | RSVD | No Connect | |
| C4 | RSVD | No Connect | |
| C3 | VSS | Analog | PWR |
| C2 | VSS | Analog | PWR |
| C1 | VSS | Analog | PWR |
| B36 | VSS | Analog | PWR |
| B35 | VSS | Analog | PWR |
| B34 | VSS | Analog | PWR |
| B33 | VSS | Analog | PWR |
| B32 | RMII RXD[0] | GPIO | I |
| B31 | VSS | Analog | PWR |
| B30 | VCCQPI1VRMRX0 | Analog | I/O |
| B29 | VCCQPI1VRMRX3 | Analog | I/O |
| B28 | VSS | Analog | PWR |
| B27 | QPI1RPDAT[7] | Intel QPI | I |
| B26 | QPI1RNDAT[7] | Intel QPI | I |
| B25 | VSS | Analog | PWR |
| B24 | RSVD | No Connect | |
| B23 | RSVD | No Connect | |
| B22 | VSS | Analog | PWR |
| B21 | QPI1RNDAT[12] | Intel QPI | I |

**Table 22-1. Pin Listing by Pin Name
(Sheet 32 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|----------------|--------------------|-----------|
| B20 | QPI1RPDAT[12] | Intel QPI | I |
| B19 | VSS | Analog | PWR |
| B18 | QPI1VRMVREFRX2 | Cmos | I |
| B17 | VCCAQPI1TXBG | Analog | I/O |
| B16 | QPI1TPDAT[12] | Intel QPI | O |
| B15 | QPI1TNDAT[12] | Intel QPI | O |
| B14 | VSS | Analog | PWR |
| B13 | QPI1TNDAT[10] | Intel QPI | O |
| B12 | QPI1TPDAT[10] | Intel QPI | O |
| B11 | VSS | Analog | PWR |
| B10 | QPI1TPDAT[8] | Intel QPI | O |
| B9 | QPI1TNDAT[8] | Intel QPI | O |
| B8 | VSS | Analog | PWR |
| B7 | QPI1TPDAT[5] | Intel QPI | O |
| B6 | QPI1TNDAT[5] | Intel QPI | O |
| B5 | VSS | Analog | PWR |
| B4 | RSVD | No Connect | |
| B3 | VSS | Analog | PWR |
| B2 | VSS | Analog | PWR |
| B1 | TEST[2] | No Connect | I/O |
| A36 | TEST[1] | No Connect | I/O |
| A35 | VSS | Analog | PWR |
| A34 | VSS | Analog | PWR |
| A33 | VSS | Analog | PWR |
| A32 | VSS | Analog | PWR |
| A31 | RMII CRS DV | GPIO | I |
| A30 | VCCAQPI1RXBG | Analog | I/O |
| A29 | VSS | Analog | PWR |
| A28 | QPI1RPDAT[6] | Intel QPI | I |
| A27 | QPI1RNDAT[6] | Intel QPI | I |
| A26 | VSS | Analog | PWR |
| A25 | QPI1RPDAT[9] | Intel QPI | I |
| A24 | QPI1RNDAT[9] | Intel QPI | I |
| A23 | VSS | Analog | PWR |
| A22 | QPI1RPDAT[10] | Intel QPI | I |
| A21 | QPI1RNDAT[10] | Intel QPI | I |
| A20 | VSS | Analog | PWR |
| A19 | QPI1RPDAT[13] | Intel QPI | I |
| A18 | QPI1RNDAT[13] | Intel QPI | I |
| A17 | VSS | Analog | PWR |



**Table 22-1. Pin Listing by Pin Name
(Sheet 33 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| A16 | VSS | Analog | PWR |
| A15 | QPI1TPDAT[11] | Intel QPI | O |
| A14 | QPI1TNDAT[11] | Intel QPI | O |
| A13 | VSS | Analog | PWR |
| A12 | QPI1TPCLK[0] | Intel QPI | O |
| A11 | QPI1TNCLK[0] | Intel QPI | O |
| A10 | VSS | Analog | PWR |
| A9 | QPI1TPDAT[7] | Intel QPI | O |
| A8 | QPI1TNDAT[7] | Intel QPI | O |
| A7 | VSS | Analog | PWR |
| A6 | QPI1TPDAT[4] | Intel QPI | O |
| A5 | QPI1TNDAT[4] | Intel QPI | O |
| A4 | VSS | Analog | PWR |
| A3 | VSS | Analog | PWR |
| A2 | TEST[0] | No Connect | I/O |

Table 22-2. Pin Listing by Signal Name
(Sheet 1 of 33)

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|--------------|--------------------|-----------|
| AH2 | QPI0TNDAT[5] | Intel QPI | O |
| AC31 | A20M_N | GPIO | I |
| C32 | AUXPWRGOOD | GPIO | I |
| AF31 | BMCINIT | GPIO | I |
| U34 | CLCLK | Cmos | I/O |
| T34 | CLDATA | Cmos | I/O |
| U32 | CLRST_N | Cmos | I |
| AC28 | CORELLPWRDET | GPIO | I |
| D34 | COREPWRGOOD | GPIO | I |
| D33 | CORERST_N | GPIO | I |
| F36 | DDRA[0] | DDR | O |
| G36 | DDRA[1] | DDR | O |
| G30 | DDRA[10] | DDR | O |
| H29 | DDRA[11] | DDR | O |
| J35 | DDRA[12] | DDR | O |
| J32 | DDRA[13] | DDR | O |
| J36 | DDRA[14] | DDR | O |
| J30 | DDRA[2] | DDR | O |
| F35 | DDRA[3] | DDR | O |
| H32 | DDRA[4] | DDR | O |
| H31 | DDRA[5] | DDR | O |
| E35 | DDRA[6] | DDR | O |
| J29 | DDRA[7] | DDR | O |
| G31 | DDRA[8] | DDR | O |
| F33 | DDRA[9] | DDR | O |
| J33 | DDRBA[0] | DDR | O |
| K36 | DDRBA[1] | DDR | O |
| K31 | DDRBA[2] | DDR | O |
| H28 | DDRCAS_N | DDR | O |
| E32 | DDRCKE | DDR | O |
| K34 | DDRCLKN | DDR | O |
| K33 | DDRCLKP | DDR | O |
| K28 | DDRCOMPX | Analog | I/O |
| L29 | DDRCRES | Analog | I/O |
| E34 | DDRCN_N | DDR | O |
| M32 | DDRD[0] | DDR | I/O |
| P35 | DDRD[1] | DDR | I/O |
| N34 | DDRD[2] | DDR | I/O |
| P34 | DDRD[3] | DDR | I/O |
| M36 | DDRD[4] | DDR | I/O |

Table 22-2. Pin Listing by Signal Name
(Sheet 2 of 33)

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|------------------|--------------------|-----------|
| N33 | DDRD[5] | DDR | I/O |
| M30 | DDRD[6] | DDR | I/O |
| P32 | DDRD[7] | DDR | I/O |
| M35 | DDRDM | DDR | O |
| M33 | DDRDM_N | DDR | O |
| M29 | DDRDRVCRES | Analog | I/O |
| N31 | DDREDQSN | DDR | O |
| N30 | DDREDQSP | DDR | O |
| U29 | DDRFREQ[2] | GPIO | I |
| T31 | DDRFREQ[3] | GPIO | I |
| F30 | DDRODT | DDR | O |
| G34 | DDRPLLREFCLKN | DDR | I |
| G33 | DDRPLLREFCLKP | DDR | O |
| F29 | DDRRAS_N | DDR | O |
| M27 | DDRRES[0] | Analog | I/O |
| K30 | DDRRES[1] | Analog | I/O |
| L28 | DDRSLEWCRES | Analog | I/O |
| F32 | DDRWE_N | DDR | O |
| P31 | DUALIOH | GPIO | I/O |
| AG33 | DUALIOH_QPIPRSEL | GPIO | I |
| AD27 | ERR_N[0] | GPIO | O |
| AF28 | ERR_N[1] | GPIO | O |
| AE28 | ERR_N[2] | GPIO | O |
| AL18 | ESIRN[0] | PCIEX | I |
| AL20 | ESIRN[1] | PCIEX | I |
| AK21 | ESIRN[2] | PCIEX | I |
| AJ19 | ESIRN[3] | PCIEX | I |
| AL19 | ESIRP[0] | PCIEX | I |
| AK20 | ESIRP[1] | PCIEX | I |
| AJ21 | ESIRP[2] | PCIEX | I |
| AK19 | ESIRP[3] | PCIEX | I |
| AR19 | ESITN[0] | PCIEX | O |
| AN20 | ESITN[1] | PCIEX | O |
| AR20 | ESITN[2] | PCIEX | O |
| AN21 | ESITN[3] | PCIEX | O |
| AP19 | ESITP[0] | PCIEX | O |
| AN19 | ESITP[1] | PCIEX | O |
| AR21 | ESITP[2] | PCIEX | O |
| AM21 | ESITP[3] | PCIEX | O |
| AK36 | EXTSYSTRIG | GPIO | I/O |



**Table 22-2. Pin Listing by Signal Name
(Sheet 3 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AL36 | FERR_N | GPIO | O |
| AE30 | INIT_N | GPIO | I |
| AB31 | INTR | GPIO | I |
| R32 | LEGACYIOH | GPIO | I/O |
| AE27 | LTRESET_N | GPIO | O |
| N27 | ME_CLK_SRC | GPIO | I |
| AD32 | NMI | GPIO | I |
| AN10 | PEOCLKN | HCSL | I |
| AN11 | PEOCLKP | HCSL | I |
| AP3 | PEOICOMPI | Analog | I/O |
| AL1 | PEOICOMPO | Analog | I/O |
| AT12 | PEOJCLKN | HCSL | I |
| AT13 | PEOJCLKP | HCSL | I |
| AN13 | PEORBIAS | Analog | I/O |
| AM1 | PEORCOMPO | Analog | I/O |
| AK33 | PE10RN[0] | PCIEX2 | I |
| AL34 | PE10RN[1] | PCIEX2 | I |
| AK32 | PE10RN[2] | PCIEX2 | I |
| AH30 | PE10RN[3] | PCIEX2 | I |
| AL33 | PE10RP[0] | PCIEX2 | I |
| AM34 | PE10RP[1] | PCIEX2 | I |
| AJ32 | PE10RP[2] | PCIEX2 | I |
| AJ30 | PE10RP[3] | PCIEX2 | I |
| AT32 | PE10TN[0] | PCIEX2 | O |
| AR31 | PE10TN[1] | PCIEX2 | O |
| AP33 | PE10TN[2] | PCIEX2 | O |
| AN32 | PE10TN[3] | PCIEX2 | O |
| AT31 | PE10TP[0] | PCIEX2 | O |
| AP31 | PE10TP[1] | PCIEX2 | O |
| AR33 | PE10TP[2] | PCIEX2 | O |
| AP32 | PE10TP[3] | PCIEX2 | O |
| AG29 | PE1CLKN | HCSL | I |
| AF29 | PE1CLKP | HCSL | I |
| AN35 | PE1ICOMPI | Analog | I/O |
| AN36 | PE1ICOMPO | Analog | I/O |
| AM30 | PE1JCLKN | HCSL | I |
| AM29 | PE1JCLKP | HCSL | I |
| AN34 | PE1RBIAS | Analog | I/O |
| AP35 | PE1RCOMPO | Analog | I/O |
| AJ16 | PE1RN[0] | PCIEX2 | I |

**Table 22-2. Pin Listing by Signal Name
(Sheet 4 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AJ18 | PE1RN[1] | PCIEX2 | I |
| AK16 | PE1RP[0] | PCIEX2 | I |
| AJ17 | PE1RP[1] | PCIEX2 | I |
| AP16 | PE1TN[0] | PCIEX2 | O |
| AR16 | PE1TN[1] | PCIEX2 | O |
| AN16 | PE1TP[0] | PCIEX2 | O |
| AR15 | PE1TP[1] | PCIEX2 | O |
| AL17 | PE2RN[0] | PCIEX2 | I |
| AM17 | PE2RN[1] | PCIEX2 | I |
| AK17 | PE2RP[0] | PCIEX2 | I |
| AM18 | PE2RP[1] | PCIEX2 | I |
| AP18 | PE2TN[0] | PCIEX2 | O |
| AR17 | PE2TN[1] | PCIEX2 | O |
| AP17 | PE2TP[0] | PCIEX2 | O |
| AT17 | PE2TP[1] | PCIEX2 | O |
| AJ6 | PE3RN[0] | PCIEX2 | I |
| AL4 | PE3RN[1] | PCIEX2 | I |
| AL5 | PE3RN[2] | PCIEX2 | I |
| AK7 | PE3RN[3] | PCIEX2 | I |
| AK6 | PE3RP[0] | PCIEX2 | I |
| AL3 | PE3RP[1] | PCIEX2 | I |
| AK5 | PE3RP[2] | PCIEX2 | I |
| AL7 | PE3RP[3] | PCIEX2 | I |
| AN4 | PE3TN[0] | PCIEX2 | O |
| AR4 | PE3TN[1] | PCIEX2 | O |
| AP6 | PE3TN[2] | PCIEX2 | O |
| AT5 | PE3TN[3] | PCIEX2 | O |
| AN5 | PE3TP[0] | PCIEX2 | O |
| AP4 | PE3TP[1] | PCIEX2 | O |
| AN6 | PE3TP[2] | PCIEX2 | O |
| AT4 | PE3TP[3] | PCIEX2 | O |
| AJ7 | PE4RN[0] | PCIEX2 | I |
| AM7 | PE4RN[1] | PCIEX2 | I |
| AL8 | PE4RN[2] | PCIEX2 | I |
| AJ9 | PE4RN[3] | PCIEX2 | I |
| AJ8 | PE4RP[0] | PCIEX2 | I |
| AM8 | PE4RP[1] | PCIEX2 | I |
| AL9 | PE4RP[2] | PCIEX2 | I |
| AK9 | PE4RP[3] | PCIEX2 | I |
| AR5 | PE4TN[0] | PCIEX2 | O |

Table 22-2. Pin Listing by Signal Name
(Sheet 5 of 33)

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AR7 | PE4TN[1] | PCIEX2 | O |
| AP7 | PE4TN[2] | PCIEX2 | O |
| AT9 | PE4TN[3] | PCIEX2 | O |
| AR6 | PE4TP[0] | PCIEX2 | O |
| AT7 | PE4TP[1] | PCIEX2 | O |
| AP8 | PE4TP[2] | PCIEX2 | O |
| AT8 | PE4TP[3] | PCIEX2 | O |
| AK10 | PE5RN[0] | PCIEX2 | I |
| AK11 | PE5RN[1] | PCIEX2 | I |
| AM13 | PE5RN[2] | PCIEX2 | I |
| AL13 | PE5RN[3] | PCIEX2 | I |
| AL10 | PE5RP[0] | PCIEX2 | I |
| AJ11 | PE5RP[1] | PCIEX2 | I |
| AM12 | PE5RP[2] | PCIEX2 | I |
| AL14 | PE5RP[3] | PCIEX2 | I |
| AR9 | PE5TN[0] | PCIEX2 | O |
| AT10 | PE5TN[1] | PCIEX2 | O |
| AP11 | PE5TN[2] | PCIEX2 | O |
| AP13 | PE5TN[3] | PCIEX2 | O |
| AP9 | PE5TP[0] | PCIEX2 | O |
| AR10 | PE5TP[1] | PCIEX2 | O |
| AR11 | PE5TP[2] | PCIEX2 | O |
| AP12 | PE5TP[3] | PCIEX2 | O |
| AK12 | PE6RN[0] | PCIEX2 | I |
| AL15 | PE6RN[1] | PCIEX2 | I |
| AJ14 | PE6RN[2] | PCIEX2 | I |
| AJ12 | PE6RN[3] | PCIEX2 | I |
| AL12 | PE6RP[0] | PCIEX2 | I |
| AK15 | PE6RP[1] | PCIEX2 | I |
| AK14 | PE6RP[2] | PCIEX2 | I |
| AJ13 | PE6RP[3] | PCIEX2 | I |
| AN15 | PE6TN[0] | PCIEX2 | O |
| AT15 | PE6TN[1] | PCIEX2 | O |
| AR14 | PE6TN[2] | PCIEX2 | O |
| AM16 | PE6TN[3] | PCIEX2 | O |
| AN14 | PE6TP[0] | PCIEX2 | O |
| AT14 | PE6TP[1] | PCIEX2 | O |
| AP14 | PE6TP[2] | PCIEX2 | O |
| AM15 | PE6TP[3] | PCIEX2 | O |
| AH24 | PE7RN[0] | PCIEX2 | I |

Table 22-2. Pin Listing by Signal Name
(Sheet 6 of 33)

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AJ23 | PE7RN[1] | PCIEX2 | I |
| AL23 | PE7RN[2] | PCIEX2 | I |
| AK24 | PE7RN[3] | PCIEX2 | I |
| AH23 | PE7RP[0] | PCIEX2 | I |
| AK23 | PE7RP[1] | PCIEX2 | I |
| AL24 | PE7RP[2] | PCIEX2 | I |
| AK25 | PE7RP[3] | PCIEX2 | I |
| AR28 | PE7TN[0] | PCIEX2 | O |
| AN27 | PE7TN[1] | PCIEX2 | O |
| AP25 | PE7TN[2] | PCIEX2 | O |
| AP23 | PE7TN[3] | PCIEX2 | O |
| AP28 | PE7TP[0] | PCIEX2 | O |
| AP27 | PE7TP[1] | PCIEX2 | O |
| AN25 | PE7TP[2] | PCIEX2 | O |
| AR23 | PE7TP[3] | PCIEX2 | O |
| AM26 | PE8RN[0] | PCIEX2 | I |
| AJ26 | PE8RN[1] | PCIEX2 | I |
| AL27 | PE8RN[2] | PCIEX2 | I |
| AH25 | PE8RN[3] | PCIEX2 | I |
| AL26 | PE8RP[0] | PCIEX2 | I |
| AJ27 | PE8RP[1] | PCIEX2 | I |
| AL28 | PE8RP[2] | PCIEX2 | I |
| AJ25 | PE8RP[3] | PCIEX2 | I |
| AN23 | PE8TN[0] | PCIEX2 | O |
| AR25 | PE8TN[1] | PCIEX2 | O |
| AR26 | PE8TN[2] | PCIEX2 | O |
| AT27 | PE8TN[3] | PCIEX2 | O |
| AN24 | PE8TP[0] | PCIEX2 | O |
| AR24 | PE8TP[1] | PCIEX2 | O |
| AP26 | PE8TP[2] | PCIEX2 | O |
| AT26 | PE8TP[3] | PCIEX2 | O |
| AL29 | PE9RN[0] | PCIEX2 | I |
| AK30 | PE9RN[1] | PCIEX2 | I |
| AL31 | PE9RN[2] | PCIEX2 | I |
| AK28 | PE9RN[3] | PCIEX2 | I |
| AK29 | PE9RP[0] | PCIEX2 | I |
| AK31 | PE9RP[1] | PCIEX2 | I |
| AL32 | PE9RP[2] | PCIEX2 | I |
| AJ28 | PE9RP[3] | PCIEX2 | I |
| AN29 | PE9TN[0] | PCIEX2 | O |



**Table 22-2. Pin Listing by Signal Name
(Sheet 7 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| AT29 | PE9TN[1] | PCIEX2 | O |
| AR30 | PE9TN[2] | PCIEX2 | O |
| AN30 | PE9TN[3] | PCIEX2 | O |
| AN28 | PE9TP[0] | PCIEX2 | O |
| AT28 | PE9TP[1] | PCIEX2 | O |
| AR29 | PE9TP[2] | PCIEX2 | O |
| AP30 | PE9TP[3] | PCIEX2 | O |
| AB28 | PEHPSCL | GPIO | O |
| AD30 | PEHPSDA | GPIO | I/O |
| AK35 | PESBLCSEL | GPIO | I |
| R33 | PEWIDTH[0] | GPIO | I/O |
| T30 | PEWIDTH[1] | GPIO | I/O |
| R36 | PEWIDTH[2] | GPIO | I/O |
| N28 | PEWIDTH[3] | GPIO | I/O |
| T28 | PEWIDTH[4] | GPIO | I/O |
| P28 | PEWIDTH[5] | GPIO | I/O |
| C35 | PLLWRDET | GPIO | I |
| Y9 | QPI0ICOMP | Analog | I/O |
| AA9 | QPI0RCOMP | Analog | I/O |
| AC7 | QPI0REFCLKN | HCSL | I |
| AB7 | QPI0REFCLKP | HCSL | I |
| L5 | QPI0RNCLK[0] | Intel QPI | I |
| K8 | QPI0RNDAT[0] | Intel QPI | I |
| J7 | QPI0RNDAT[1] | Intel QPI | I |
| N1 | QPI0RNDAT[10] | Intel QPI | I |
| L3 | QPI0RNDAT[11] | Intel QPI | I |
| N2 | QPI0RNDAT[12] | Intel QPI | I |
| T1 | QPI0RNDAT[13] | Intel QPI | I |
| P3 | QPI0RNDAT[14] | Intel QPI | I |
| R4 | QPI0RNDAT[15] | Intel QPI | I |
| T5 | QPI0RNDAT[16] | Intel QPI | I |
| P5 | QPI0RNDAT[17] | Intel QPI | I |
| R7 | QPI0RNDAT[18] | Intel QPI | I |
| P6 | QPI0RNDAT[19] | Intel QPI | I |
| H6 | QPI0RNDAT[2] | Intel QPI | I |
| G5 | QPI0RNDAT[3] | Intel QPI | I |
| J4 | QPI0RNDAT[4] | Intel QPI | I |
| G4 | QPI0RNDAT[5] | Intel QPI | I |
| G1 | QPI0RNDAT[6] | Intel QPI | I |
| H2 | QPI0RNDAT[7] | Intel QPI | I |

**Table 22-2. Pin Listing by Signal Name
(Sheet 8 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| J3 | QPI0RNDAT[8] | Intel QPI | I |
| K1 | QPI0RNDAT[9] | Intel QPI | I |
| K5 | QPI0RPCLK[0] | Intel QPI | I |
| L8 | QPI0RPDAT[0] | Intel QPI | I |
| K7 | QPI0RPDAT[1] | Intel QPI | I |
| M1 | QPI0RPDAT[10] | Intel QPI | I |
| M3 | QPI0RPDAT[11] | Intel QPI | I |
| P2 | QPI0RPDAT[12] | Intel QPI | I |
| R1 | QPI0RPDAT[13] | Intel QPI | I |
| R3 | QPI0RPDAT[14] | Intel QPI | I |
| T4 | QPI0RPDAT[15] | Intel QPI | I |
| U5 | QPI0RPDAT[16] | Intel QPI | I |
| N5 | QPI0RPDAT[17] | Intel QPI | I |
| T7 | QPI0RPDAT[18] | Intel QPI | I |
| R6 | QPI0RPDAT[19] | Intel QPI | I |
| J6 | QPI0RPDAT[2] | Intel QPI | I |
| H5 | QPI0RPDAT[3] | Intel QPI | I |
| K4 | QPI0RPDAT[4] | Intel QPI | I |
| F4 | QPI0RPDAT[5] | Intel QPI | I |
| F1 | QPI0RPDAT[6] | Intel QPI | I |
| G2 | QPI0RPDAT[7] | Intel QPI | I |
| H3 | QPI0RPDAT[8] | Intel QPI | I |
| J1 | QPI0RPDAT[9] | Intel QPI | I |
| N7 | QPI0RXBG[0] | Analog | I/O |
| M7 | QPI0RXBG[1] | Analog | I/O |
| AC1 | QPI0TNCLK[0] | Intel QPI | O |
| AG6 | QPI0TNDAT[0] | Intel QPI | O |
| AG5 | QPI0TNDAT[1] | Intel QPI | O |
| AA2 | QPI0TNDAT[10] | Intel QPI | O |
| Y1 | QPI0TNDAT[11] | Intel QPI | O |
| W2 | QPI0TNDAT[12] | Intel QPI | O |
| Y3 | QPI0TNDAT[13] | Intel QPI | O |
| W4 | QPI0TNDAT[14] | Intel QPI | O |
| W5 | QPI0TNDAT[15] | Intel QPI | O |
| W8 | QPI0TNDAT[16] | Intel QPI | O |
| W7 | QPI0TNDAT[17] | Intel QPI | O |
| Y6 | QPI0TNDAT[18] | Intel QPI | O |
| AA5 | QPI0TNDAT[19] | Intel QPI | O |
| AJ4 | QPI0TNDAT[2] | Intel QPI | O |
| AG3 | QPI0TNDAT[3] | Intel QPI | O |



**Table 22-2. Pin Listing by Signal Name
(Sheet 9 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| AJ1 | QPIOTNDAT[4] | Intel QPI | O |
| AF4 | QPIOTNDAT[6] | Intel QPI | O |
| AF1 | QPIOTNDAT[7] | Intel QPI | O |
| AE2 | QPIOTNDAT[8] | Intel QPI | O |
| AD3 | QPIOTNDAT[9] | Intel QPI | O |
| AB1 | QPIOTPCLK[0] | Intel QPI | O |
| AF6 | QPIOTPDAT[0] | Intel QPI | O |
| AH5 | QPIOTPDAT[1] | Intel QPI | O |
| AB2 | QPIOTPDAT[10] | Intel QPI | O |
| W1 | QPIOTPDAT[11] | Intel QPI | O |
| V2 | QPIOTPDAT[12] | Intel QPI | O |
| AA3 | QPIOTPDAT[13] | Intel QPI | O |
| Y4 | QPIOTPDAT[14] | Intel QPI | O |
| V6 | QPIOTPDAT[15] | Intel QPI | O |
| V8 | QPIOTPDAT[16] | Intel QPI | O |
| Y7 | QPIOTPDAT[17] | Intel QPI | O |
| AA6 | QPIOTPDAT[18] | Intel QPI | O |
| AB5 | QPIOTPDAT[19] | Intel QPI | O |
| AH4 | QPIOTPDAT[2] | Intel QPI | O |
| AF3 | QPIOTPDAT[3] | Intel QPI | O |
| AH1 | QPIOTPDAT[4] | Intel QPI | O |
| AG2 | QPIOTPDAT[5] | Intel QPI | O |
| AE4 | QPIOTPDAT[6] | Intel QPI | O |
| AE1 | QPIOTPDAT[7] | Intel QPI | O |
| AD2 | QPIOTPDAT[8] | Intel QPI | O |
| AC3 | QPIOTPDAT[9] | Intel QPI | O |
| AA8 | QPIOTXBG[0] | Analog | I/O |
| AB8 | QPIOTXBG[1] | Analog | I/O |
| L10 | QPIOVRMVREF0 | Cmos | I |
| M11 | QPIOVRMVREF1 | Cmos | I |
| T2 | QPIOVRMVREF2 | Cmos | I |
| U3 | QPIOVRMVREF3 | Cmos | I |
| AB10 | QPIOVRMVREF4 | Cmos | I |
| J14 | QPI1ICOMP | Analog | I/O |
| J13 | QPI1RCOMP | Analog | I/O |
| G11 | QPI1REFCLKN | HCSL | I |
| G12 | QPI1REFCLKP | HCSL | I |
| E23 | QPI1RNCLK[0] | Intel QPI | I |
| H24 | QPI1RNDAT[0] | Intel QPI | I |
| G25 | QPI1RNDAT[1] | Intel QPI | I |

**Table 22-2. Pin Listing by Signal Name
(Sheet 10 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| A21 | QPI1RNDAT[10] | Intel QPI | I |
| C23 | QPI1RNDAT[11] | Intel QPI | I |
| B21 | QPI1RNDAT[12] | Intel QPI | I |
| A18 | QPI1RNDAT[13] | Intel QPI | I |
| C20 | QPI1RNDAT[14] | Intel QPI | I |
| D19 | QPI1RNDAT[15] | Intel QPI | I |
| E18 | QPI1RNDAT[16] | Intel QPI | I |
| E20 | QPI1RNDAT[17] | Intel QPI | I |
| G19 | QPI1RNDAT[18] | Intel QPI | I |
| F20 | QPI1RNDAT[19] | Intel QPI | I |
| F26 | QPI1RNDAT[2] | Intel QPI | I |
| E27 | QPI1RNDAT[3] | Intel QPI | I |
| D25 | QPI1RNDAT[4] | Intel QPI | I |
| D27 | QPI1RNDAT[5] | Intel QPI | I |
| A27 | QPI1RNDAT[6] | Intel QPI | I |
| B26 | QPI1RNDAT[7] | Intel QPI | I |
| C25 | QPI1RNDAT[8] | Intel QPI | I |
| A24 | QPI1RNDAT[9] | Intel QPI | I |
| E24 | QPI1RPCLK[0] | Intel QPI | I |
| H23 | QPI1RPDAT[0] | Intel QPI | I |
| G24 | QPI1RPDAT[1] | Intel QPI | I |
| A22 | QPI1RPDAT[10] | Intel QPI | I |
| C22 | QPI1RPDAT[11] | Intel QPI | I |
| B20 | QPI1RPDAT[12] | Intel QPI | I |
| A19 | QPI1RPDAT[13] | Intel QPI | I |
| C19 | QPI1RPDAT[14] | Intel QPI | I |
| D18 | QPI1RPDAT[15] | Intel QPI | I |
| E17 | QPI1RPDAT[16] | Intel QPI | I |
| E21 | QPI1RPDAT[17] | Intel QPI | I |
| G18 | QPI1RPDAT[18] | Intel QPI | I |
| F19 | QPI1RPDAT[19] | Intel QPI | I |
| F25 | QPI1RPDAT[2] | Intel QPI | I |
| E26 | QPI1RPDAT[3] | Intel QPI | I |
| D24 | QPI1RPDAT[4] | Intel QPI | I |
| D28 | QPI1RPDAT[5] | Intel QPI | I |
| A28 | QPI1RPDAT[6] | Intel QPI | I |
| B27 | QPI1RPDAT[7] | Intel QPI | I |
| C26 | QPI1RPDAT[8] | Intel QPI | I |
| A25 | QPI1RPDAT[9] | Intel QPI | I |
| G21 | QPI1RXBG[0] | Analog | I/O |



**Table 22-2. Pin Listing by Signal Name
(Sheet 11 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|---------------|--------------------|-----------|
| G22 | QPI1RXBG[1] | Analog | I/O |
| A11 | QPI1TNCLK[0] | Intel QPI | O |
| F7 | QPI1TNDAT[0] | Intel QPI | O |
| E7 | QPI1TNDAT[1] | Intel QPI | O |
| B13 | QPI1TNDAT[10] | Intel QPI | O |
| A14 | QPI1TNDAT[11] | Intel QPI | O |
| B15 | QPI1TNDAT[12] | Intel QPI | O |
| C14 | QPI1TNDAT[13] | Intel QPI | O |
| D15 | QPI1TNDAT[14] | Intel QPI | O |
| E15 | QPI1TNDAT[15] | Intel QPI | I |
| H15 | QPI1TNDAT[16] | Intel QPI | O |
| G15 | QPI1TNDAT[17] | Intel QPI | O |
| F14 | QPI1TNDAT[18] | Intel QPI | O |
| E13 | QPI1TNDAT[19] | Intel QPI | O |
| D5 | QPI1TNDAT[2] | Intel QPI | O |
| C7 | QPI1TNDAT[3] | Intel QPI | O |
| A5 | QPI1TNDAT[4] | Intel QPI | O |
| B6 | QPI1TNDAT[5] | Intel QPI | O |
| D8 | QPI1TNDAT[6] | Intel QPI | O |
| A8 | QPI1TNDAT[7] | Intel QPI | O |
| B9 | QPI1TNDAT[8] | Intel QPI | O |
| C10 | QPI1TNDAT[9] | Intel QPI | O |
| A12 | QPI1TPCLK[0] | Intel QPI | O |
| F8 | QPI1TPDAT[0] | Intel QPI | O |
| E6 | QPI1TPDAT[1] | Intel QPI | O |
| B12 | QPI1TPDAT[10] | Intel QPI | O |
| A15 | QPI1TPDAT[11] | Intel QPI | O |
| B16 | QPI1TPDAT[12] | Intel QPI | O |
| C13 | QPI1TPDAT[13] | Intel QPI | O |
| D14 | QPI1TPDAT[14] | Intel QPI | O |
| F16 | QPI1TPDAT[15] | Intel QPI | O |
| H16 | QPI1TPDAT[16] | Intel QPI | O |
| G14 | QPI1TPDAT[17] | Intel QPI | O |
| F13 | QPI1TPDAT[18] | Intel QPI | O |
| E12 | QPI1TPDAT[19] | Intel QPI | O |
| D6 | QPI1TPDAT[2] | Intel QPI | O |
| C8 | QPI1TPDAT[3] | Intel QPI | O |
| A6 | QPI1TPDAT[4] | Intel QPI | O |
| B7 | QPI1TPDAT[5] | Intel QPI | O |
| D9 | QPI1TPDAT[6] | Intel QPI | O |

**Table 22-2. Pin Listing by Signal Name
(Sheet 12 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|----------------|--------------------|-----------|
| A9 | QPI1TPDAT[7] | Intel QPI | O |
| B10 | QPI1TPDAT[8] | Intel QPI | O |
| C11 | QPI1TPDAT[9] | Intel QPI | O |
| H13 | QPI1TXBG[0] | Analog | I/O |
| H12 | QPI1TXBG[1] | Analog | I/O |
| J23 | QPI1VRMVREFRX0 | Cmos | I |
| H26 | QPI1VRMVREFRX1 | Cmos | I |
| B18 | QPI1VRMVREFRX2 | Cmos | I |
| C17 | QPI1VRMVREFRX3 | Cmos | I |
| K11 | QPI1VRMVREFTX | Cmos | I |
| AG35 | QPIFREQSELO | GPIO | I |
| AA30 | QPIFREQSEL1 | GPIO | I |
| AJ35 | QPISBLCSEL | GPIO | I |
| AL2 | RESETO_N | GPIO | I/O |
| G28 | RMIICLK | GPIO | I |
| J26 | RMIICLKREFOUT | GPIO | O |
| A31 | RMIICRSDV | GPIO | I |
| D30 | RMIIMDC | GPIO | O |
| D31 | RMIIMDIO | GPIO | I/O |
| B32 | RMIIRXD[0] | GPIO | I |
| E29 | RMIIRXD[1] | GPIO | I |
| E31 | RMIITXD[0] | GPIO | O |
| J27 | RMIITXD[1] | GPIO | O |
| G27 | RMIITXEN | GPIO | O |
| AT24 | RSVD | No Connect | |
| AT23 | RSVD | No Connect | |
| AP21 | RSVD | No Connect | |
| AN33 | RSVD | No Connect | |
| AN18 | RSVD | No Connect | |
| AM35 | RSVD | No Connect | |
| AM31 | RSVD | No Connect | |
| AM27 | RSVD | No Connect | |
| AM25 | RSVD | No Connect | |
| AM24 | RSVD | No Connect | |
| AM22 | RSVD | No Connect | |
| AM20 | RSVD | No Connect | |
| AM11 | RSVD | No Connect | |
| AM10 | RSVD | No Connect | |
| AM5 | RSVD | No Connect | |
| AM3 | RSVD | No Connect | |



**Table 22-2. Pin Listing by Signal Name
(Sheet 13 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AM2 | RSVD | No Connect | |
| AK3 | RSVD | No Connect | |
| AH31 | RSVD | No Connect | |
| AH29 | RSVD | No Connect | |
| AH28 | RSVD | No Connect | |
| AH26 | RSVD | No Connect | |
| AH21 | RSVD | No Connect | |
| AH20 | RSVD | No Connect | |
| AH19 | RSVD | No Connect | |
| AH18 | RSVD | No Connect | |
| AH16 | RSVD | No Connect | |
| AH15 | RSVD | No Connect | |
| AH14 | RSVD | No Connect | |
| AH13 | RSVD | No Connect | |
| AH11 | RSVD | No Connect | |
| AH10 | RSVD | No Connect | |
| AH9 | RSVD | No Connect | |
| AH8 | RSVD | No Connect | |
| AG30 | RSVD | No Connect | |
| AG27 | RSVD | No Connect | |
| AG26 | RSVD | No Connect | |
| AG9 | RSVD | No Connect | |
| AG8 | RSVD | No Connect | |
| AF26 | RSVD | No Connect | |
| AF10 | RSVD | No Connect | |
| AF9 | RSVD | No Connect | |
| AF7 | RSVD | No Connect | |
| AE10 | RSVD | No Connect | |
| AE8 | RSVD | No Connect | |
| AE7 | RSVD | No Connect | |
| AD10 | RSVD | No Connect | |
| AD9 | RSVD | No Connect | |
| AD8 | RSVD | No Connect | |
| AC25 | RSVD | No Connect | |
| AC9 | RSVD | No Connect | |
| AC6 | RSVD | No Connect | |
| AC4 | RSVD | No Connect | |
| AB4 | RSVD | No Connect | |
| AA32 | RSVD | No Connect | |
| AA27 | RSVD | No Connect | |

**Table 22-2. Pin Listing by Signal Name
(Sheet 14 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| Y32 | RSVD | No Connect | |
| Y10 | RSVD | No Connect | |
| W30 | RSVD | No Connect | |
| W28 | RSVD | No Connect | |
| W10 | RSVD | No Connect | |
| V27 | RSVD | No Connect | |
| V9 | RSVD | No Connect | |
| V5 | RSVD | No Connect | |
| U10 | RSVD | No Connect | |
| U9 | RSVD | No Connect | |
| U8 | RSVD | No Connect | |
| U6 | RSVD | No Connect | |
| T10 | RSVD | No Connect | |
| T8 | RSVD | No Connect | |
| R9 | RSVD | No Connect | |
| P10 | RSVD | No Connect | |
| P9 | RSVD | No Connect | |
| P8 | RSVD | No Connect | |
| N11 | RSVD | No Connect | |
| N8 | RSVD | No Connect | |
| M9 | RSVD | No Connect | |
| M4 | RSVD | No Connect | |
| L35 | RSVD | No Connect | |
| L34 | RSVD | No Connect | |
| L32 | RSVD | No Connect | |
| L31 | RSVD | No Connect | |
| L26 | RSVD | No Connect | |
| L9 | RSVD | No Connect | |
| L6 | RSVD | No Connect | |
| L2 | RSVD | No Connect | |
| K27 | RSVD | No Connect | |
| K25 | RSVD | No Connect | |
| K24 | RSVD | No Connect | |
| K13 | RSVD | No Connect | |
| K12 | RSVD | No Connect | |
| K2 | RSVD | No Connect | |
| J24 | RSVD | No Connect | |
| J21 | RSVD | No Connect | |
| J19 | RSVD | No Connect | |
| J16 | RSVD | No Connect | |



**Table 22-2. Pin Listing by Signal Name
(Sheet 15 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|----------------|--------------------|-----------|
| J11 | RSVD | No Connect | |
| J10 | RSVD | No Connect | |
| H35 | RSVD | No Connect | |
| H34 | RSVD | No Connect | |
| H21 | RSVD | No Connect | |
| H20 | RSVD | No Connect | |
| H18 | RSVD | No Connect | |
| H17 | RSVD | No Connect | |
| H10 | RSVD | No Connect | |
| H9 | RSVD | No Connect | |
| G9 | RSVD | No Connect | |
| G8 | RSVD | No Connect | |
| F23 | RSVD | No Connect | |
| F17 | RSVD | No Connect | |
| F11 | RSVD | No Connect | |
| E16 | RSVD | No Connect | |
| D22 | RSVD | No Connect | |
| D12 | RSVD | No Connect | |
| D11 | RSVD | No Connect | |
| C5 | RSVD | No Connect | |
| C4 | RSVD | No Connect | |
| B24 | RSVD | No Connect | |
| B23 | RSVD | No Connect | |
| B4 | RSVD | No Connect | |
| AB27 | SMBSCSCL | GPIO | I/O |
| AD29 | SMBSDA | GPIO | I/O |
| AE31 | SMBUSID | GPIO | I |
| AG32 | SMI_N | GPIO | I |
| T33 | TCK | GPIO | I |
| V26 | TDI | GPIO | I |
| V30 | TDO | GPIO | O |
| A2 | TEST[0] | No Connect | I/O |
| A36 | TEST[1] | No Connect | I/O |
| B1 | TEST[2] | No Connect | I/O |
| AT1 | TEST[3] | No Connect | I/O |
| AT36 | TEST[4] | No Connect | I/O |
| P29 | FWAGNT_ESIMODE | GPIO | I |
| U28 | TESTHI[2] | GPIO | I |
| R29 | TESTHI[3] | GPIO | I |
| AR12 | TESTLO1 | Analog | I/O |

**Table 22-2. Pin Listing by Signal Name
(Sheet 16 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|--------------|--------------------|-----------|
| AF32 | TESTLO10 | GPIO | I |
| AE34 | TESTLO11 | GPIO | I/O |
| AC32 | TESTLO12 | GPIO | I |
| AB30 | TESTLO13 | GPIO | I |
| AA29 | QPICKFAIL | GPIO | I |
| Y28 | TESTLO15 | GPIO | I |
| W27 | TESTLO16 | GPIO | I |
| V32 | TESTLO17 | GPIO | I |
| T27 | TESTLO18 | GPIO | I |
| R35 | TESTLO19 | GPIO | I/O |
| AN9 | TESTLO2 | Analog | I/O |
| AD33 | TESTLO21 | GPIO | I/O |
| C33 | TESTLO22 | GPIO | I/O |
| AC29 | TESTLO23 | GPIO | I/O |
| AA26 | TESTLO24 | GPIO | I/O |
| AE33 | TESTLO25 | GPIO | I/O |
| D36 | TESTLO26 | GPIO | I/O |
| AN8 | TESTLO3 | Analog | I/O |
| AM6 | TESTLO4 | Analog | I/O |
| AJ34 | TESTLO5 | GPIO | I |
| AH34 | NODEID[2] | GPIO | I |
| AH33 | TESTHI[4] | GPIO | I |
| AF35 | TESTLO8 | GPIO | I |
| AF34 | TESTLO9 | GPIO | I |
| AD26 | THERMALERT_N | GPIO | O |
| AC26 | THERMTRIP_N | GPIO | I/O |
| U31 | TMS | GPIO | I |
| R30 | TRST_N | GPIO | I |
| AJ3 | TSIREF | Analog | I |
| AC22 | VCC | Analog | PWR |
| AC20 | VCC | Analog | PWR |
| AB23 | VCC | Analog | PWR |
| AB21 | VCC | Analog | PWR |
| AB19 | VCC | Analog | PWR |
| AB17 | VCC | Analog | PWR |
| AB15 | VCC | Analog | PWR |
| AA24 | VCC | Analog | PWR |
| AA22 | VCC | Analog | PWR |
| AA20 | VCC | Analog | PWR |
| AA18 | VCC | Analog | PWR |



**Table 22-2. Pin Listing by Signal Name
(Sheet 17 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AA16 | VCC | Analog | PWR |
| Y26 | VCC | Analog | PWR |
| Y25 | VCC | Analog | PWR |
| Y23 | VCC | Analog | PWR |
| Y21 | VCC | Analog | PWR |
| Y19 | VCC | Analog | PWR |
| Y17 | VCC | Analog | PWR |
| W24 | VCC | Analog | PWR |
| W22 | VCC | Analog | PWR |
| W20 | VCC | Analog | PWR |
| W18 | VCC | Analog | PWR |
| W16 | VCC | Analog | PWR |
| V24 | VCC | Analog | PWR |
| V23 | VCC | Analog | PWR |
| V21 | VCC | Analog | PWR |
| V19 | VCC | Analog | PWR |
| V17 | VCC | Analog | PWR |
| U22 | VCC | Analog | PWR |
| U20 | VCC | Analog | PWR |
| U18 | VCC | Analog | PWR |
| U16 | VCC | Analog | PWR |
| T19 | VCC | Analog | PWR |
| T17 | VCC | Analog | PWR |
| R20 | VCC | Analog | PWR |
| R18 | VCC | Analog | PWR |
| R16 | VCC | Analog | PWR |
| P19 | VCC | Analog | PWR |
| P17 | VCC | Analog | PWR |
| P15 | VCC | Analog | PWR |
| N36 | VCCADDRPLL | Analog | PWR |
| AF19 | VCCAPE | Analog | PWR |
| AF17 | VCCAPE | Analog | PWR |
| AF15 | VCCAPE | Analog | PWR |
| AF13 | VCCAPE | Analog | PWR |
| AF12 | VCCAPE | Analog | PWR |
| AE19 | VCCAPE | Analog | PWR |
| AE18 | VCCAPE | Analog | PWR |
| AE17 | VCCAPE | Analog | PWR |
| AE16 | VCCAPE | Analog | PWR |
| AE15 | VCCAPE | Analog | PWR |

**Table 22-2. Pin Listing by Signal Name
(Sheet 18 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AE14 | VCCAPE | Analog | PWR |
| AE13 | VCCAPE | Analog | PWR |
| AE12 | VCCAPE | Analog | PWR |
| AD19 | VCCAPE | Analog | PWR |
| AD18 | VCCAPE | Analog | PWR |
| AD17 | VCCAPE | Analog | PWR |
| AD16 | VCCAPE | Analog | PWR |
| AD15 | VCCAPE | Analog | PWR |
| AD14 | VCCAPE | Analog | PWR |
| AD13 | VCCAPE | Analog | PWR |
| AD12 | VCCAPE | Analog | PWR |
| AC19 | VCCAPE | Analog | PWR |
| AC17 | VCCAPE | Analog | PWR |
| AC15 | VCCAPE | Analog | PWR |
| AC13 | VCCAPE | Analog | PWR |
| AC12 | VCCAPE | Analog | PWR |
| AF23 | VCCAPE1 | Analog | PWR |
| AF22 | VCCAPE1 | Analog | PWR |
| AF21 | VCCAPE1 | Analog | PWR |
| AF20 | VCCAPE1 | Analog | PWR |
| AE25 | VCCAPE1 | Analog | PWR |
| AE24 | VCCAPE1 | Analog | PWR |
| AE23 | VCCAPE1 | Analog | PWR |
| AE22 | VCCAPE1 | Analog | PWR |
| AE21 | VCCAPE1 | Analog | PWR |
| AE20 | VCCAPE1 | Analog | PWR |
| AD25 | VCCAPE1 | Analog | PWR |
| AD24 | VCCAPE1 | Analog | PWR |
| AD23 | VCCAPE1 | Analog | PWR |
| AD22 | VCCAPE1 | Analog | PWR |
| AD21 | VCCAPE1 | Analog | PWR |
| AD20 | VCCAPE1 | Analog | PWR |
| AT19 | VCCAPE1BG | Analog | PWR |
| AT20 | VCCAPE1PLL | Analog | PWR |
| AN2 | VCCAPEBG | Analog | PWR |
| AN1 | VCCAPEPLL | Analog | PWR |
| AA15 | VCCAQPIO | Analog | PWR |
| AA14 | VCCAQPIO | Analog | PWR |
| AA13 | VCCAQPIO | Analog | PWR |
| AA12 | VCCAQPIO | Analog | PWR |



**Table 22-2. Pin Listing by Signal Name
(Sheet 19 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|--------------|--------------------|-----------|
| Y15 | VCCAQPIO | Analog | PWR |
| Y14 | VCCAQPIO | Analog | PWR |
| W15 | VCCAQPIO | Analog | PWR |
| W14 | VCCAQPIO | Analog | PWR |
| W13 | VCCAQPIO | Analog | PWR |
| W12 | VCCAQPIO | Analog | PWR |
| W11 | VCCAQPIO | Analog | PWR |
| V15 | VCCAQPIO | Analog | PWR |
| V14 | VCCAQPIO | Analog | PWR |
| U15 | VCCAQPIO | Analog | PWR |
| U14 | VCCAQPIO | Analog | PWR |
| U13 | VCCAQPIO | Analog | PWR |
| U12 | VCCAQPIO | Analog | PWR |
| U11 | VCCAQPIO | Analog | PWR |
| T15 | VCCAQPIO | Analog | PWR |
| T14 | VCCAQPIO | Analog | PWR |
| R14 | VCCAQPIO | Analog | PWR |
| R13 | VCCAQPIO | Analog | PWR |
| R12 | VCCAQPIO | Analog | PWR |
| P13 | VCCAQPIO | Analog | PWR |
| P12 | VCCAQPIO | Analog | PWR |
| N12 | VCCAQPIO | Analog | PWR |
| AD5 | VCCAQPIOPLL | Analog | I/O |
| D1 | VCCAQPIORXBG | Analog | I/O |
| U2 | VCCAQPIOTXBG | Analog | I/O |
| N19 | VCCAQPI1 | Analog | PWR |
| N18 | VCCAQPI1 | Analog | PWR |
| N17 | VCCAQPI1 | Analog | PWR |
| N16 | VCCAQPI1 | Analog | PWR |
| N15 | VCCAQPI1 | Analog | PWR |
| N14 | VCCAQPI1 | Analog | PWR |
| N13 | VCCAQPI1 | Analog | PWR |
| M20 | VCCAQPI1 | Analog | PWR |
| M19 | VCCAQPI1 | Analog | PWR |
| M18 | VCCAQPI1 | Analog | PWR |
| M17 | VCCAQPI1 | Analog | PWR |
| M16 | VCCAQPI1 | Analog | PWR |
| M15 | VCCAQPI1 | Analog | PWR |
| M14 | VCCAQPI1 | Analog | PWR |
| M13 | VCCAQPI1 | Analog | PWR |

**Table 22-2. Pin Listing by Signal Name
(Sheet 20 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|--------------|--------------------|-----------|
| M12 | VCCAQPI1 | Analog | PWR |
| L20 | VCCAQPI1 | Analog | PWR |
| L19 | VCCAQPI1 | Analog | PWR |
| L17 | VCCAQPI1 | Analog | PWR |
| L15 | VCCAQPI1 | Analog | PWR |
| L13 | VCCAQPI1 | Analog | PWR |
| K19 | VCCAQPI1 | Analog | PWR |
| K17 | VCCAQPI1 | Analog | PWR |
| K15 | VCCAQPI1 | Analog | PWR |
| E10 | VCCAQPI1PLL | Analog | PWR |
| A30 | VCCAQPI1RXBG | Analog | I/O |
| B17 | VCCAQPI1TXBG | Analog | I/O |
| R27 | VCCCLPWRP | Analog | I/O |
| P26 | VCCDDR18 | Analog | PWR |
| P25 | VCCDDR18 | Analog | PWR |
| P23 | VCCDDR18 | Analog | PWR |
| N25 | VCCDDR18 | Analog | PWR |
| N24 | VCCDDR18 | Analog | PWR |
| M26 | VCCDDR18 | Analog | PWR |
| M24 | VCCDDR18 | Analog | PWR |
| M23 | VCCDDR18 | Analog | PWR |
| L25 | VCCDDR18 | Analog | PWR |
| AT22 | VCCDPE1PLL | Analog | PWR |
| AP2 | VCCDPEPLL | Analog | PWR |
| V29 | VCCEPW | Analog | I/O |
| T24 | VCCEPW | Analog | PWR |
| T23 | VCCEPW | Analog | PWR |
| T21 | VCCEPW | Analog | PWR |
| R26 | VCCEPW | Analog | PWR |
| R22 | VCCEPW | Analog | PWR |
| P21 | VCCEPW | Analog | PWR |
| N22 | VCCEPW | Analog | PWR |
| L22 | VCCEPW | Analog | PWR |
| K23 | VCCEPW | Analog | PWR |
| K22 | VCCEPW | Analog | PWR |
| AB25 | VCCMISC33 | Analog | PWR |
| AB24 | VCCMISC33 | Analog | PWR |
| M21 | VCCMISC33EPW | Analog | PWR |
| K21 | VCCMISC33EPW | Analog | PWR |
| K20 | VCCMISC33EPW | Analog | PWR |



**Table 22-2. Pin Listing by Signal Name
(Sheet 21 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-----------------|--------------------|-----------|
| AR34 | VCCPE1VRM | Analog | PWR |
| AT18 | VCCPEVRM | Analog | PWR |
| D2 | VCCQPI0VRMRX0 | Analog | I/O |
| E3 | VCCQPI0VRMRX1 | Analog | I/O |
| F3 | VCCQPI0VRMRX2 | Analog | I/O |
| E2 | VCCQPI0VRMRX3 | Analog | I/O |
| N4 | VCCQPI0VRMRXOP0 | Analog | I/O |
| M6 | VCCQPI0VRMRXOP1 | Analog | I/O |
| AD6 | VCCQPI0VRMRXOP2 | Analog | I/O |
| V3 | VCCQPI0VRMRXOP3 | Analog | I/O |
| AE5 | VCCQPI0VRMTX | Analog | I/O |
| AC10 | VCCQPI0VRMTXOP0 | Analog | I/O |
| B30 | VCCQPI1VRMRX0 | Analog | I/O |
| C29 | VCCQPI1VRMRX1 | Analog | I/O |
| C28 | VCCQPI1VRMRX2 | Analog | I/O |
| B29 | VCCQPI1VRMRX3 | Analog | I/O |
| D21 | VCCQPI1VRMRXOP0 | Analog | I/O |
| F22 | VCCQPI1VRMRXOP1 | Analog | I/O |
| F10 | VCCQPI1VRMRXOP2 | Analog | I/O |
| C16 | VCCQPI1VRMRXOP3 | Analog | I/O |
| E9 | VCCQPI1VRMTX | Analog | I/O |
| K10 | VCCQPI1VRMTXOP0 | Analog | I/O |
| AK2 | VCCTS | Analog | PWR |
| U26 | VCCXDP18 | Analog | PWR |
| T25 | VCCXDP18 | Analog | PWR |
| T36 | VREFCL | Cmos | I/O |
| Y29 | VRMEN | GPIO | I |
| AT35 | VSS | Analog | PWR |
| AT34 | VSS | Analog | PWR |
| AT33 | VSS | Analog | PWR |
| AT30 | VSS | Analog | PWR |
| AT25 | VSS | Analog | PWR |
| AT21 | VSS | Analog | PWR |
| AT16 | VSS | Analog | PWR |
| AT11 | VSS | Analog | PWR |
| AT6 | VSS | Analog | PWR |
| AT3 | VSS | Analog | PWR |
| AT2 | VSS | Analog | PWR |
| AR36 | VSS | Analog | PWR |
| AR35 | VSS | Analog | PWR |

**Table 22-2. Pin Listing by Signal Name
(Sheet 22 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AR32 | VSS | Analog | PWR |
| AR27 | VSS | Analog | PWR |
| AR22 | VSS | Analog | PWR |
| AR18 | VSS | Analog | PWR |
| AR13 | VSS | Analog | PWR |
| AR8 | VSS | Analog | PWR |
| AR3 | VSS | Analog | PWR |
| AR2 | VSS | Analog | PWR |
| AR1 | VSS | Analog | PWR |
| AP36 | VSS | Analog | PWR |
| AP34 | VSS | Analog | PWR |
| AP29 | VSS | Analog | PWR |
| AP24 | VSS | Analog | PWR |
| AP22 | VSS | Analog | PWR |
| AP20 | VSS | Analog | PWR |
| AP15 | VSS | Analog | PWR |
| AP10 | VSS | Analog | PWR |
| AP5 | VSS | Analog | PWR |
| AP1 | VSS | Analog | PWR |
| AN31 | VSS | Analog | PWR |
| AN26 | VSS | Analog | PWR |
| AN22 | VSS | Analog | PWR |
| AN17 | VSS | Analog | PWR |
| AN12 | VSS | Analog | PWR |
| AN7 | VSS | Analog | PWR |
| AN3 | VSS | Analog | PWR |
| AM36 | VSS | Analog | PWR |
| AM33 | VSS | Analog | PWR |
| AM32 | VSS | Analog | PWR |
| AM28 | VSS | Analog | PWR |
| AM23 | VSS | Analog | PWR |
| AM19 | VSS | Analog | PWR |
| AM14 | VSS | Analog | PWR |
| AM9 | VSS | Analog | PWR |
| AM4 | VSS | Analog | PWR |
| AL35 | VSS | Analog | PWR |
| AL30 | VSS | Analog | PWR |
| AL25 | VSS | Analog | PWR |
| AL22 | VSS | Analog | PWR |
| AL21 | VSS | Analog | PWR |



**Table 22-2. Pin Listing by Signal Name
(Sheet 23 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AL16 | VSS | Analog | PWR |
| AL11 | VSS | Analog | PWR |
| AL6 | VSS | Analog | PWR |
| AK34 | VSS | Analog | PWR |
| AK27 | VSS | Analog | PWR |
| AK26 | VSS | Analog | PWR |
| AK22 | VSS | Analog | PWR |
| AK18 | VSS | Analog | PWR |
| AK13 | VSS | Analog | PWR |
| AK8 | VSS | Analog | PWR |
| AK4 | VSS | Analog | PWR |
| AK1 | VSS | Analog | PWR |
| AJ36 | VSS | Analog | PWR |
| AJ33 | VSS | Analog | PWR |
| AJ31 | VSS | Analog | PWR |
| AJ29 | VSS | Analog | PWR |
| AJ24 | VSS | Analog | PWR |
| AJ22 | VSS | Analog | PWR |
| AJ20 | VSS | Analog | PWR |
| AJ15 | VSS | Analog | PWR |
| AJ10 | VSS | Analog | PWR |
| AJ5 | VSS | Analog | PWR |
| AJ2 | VSS | Analog | PWR |
| AH35 | VSS | Analog | PWR |
| AH32 | VSS | Analog | PWR |
| AH27 | VSS | Analog | PWR |
| AH22 | VSS | Analog | PWR |
| AH17 | VSS | Analog | PWR |
| AH12 | VSS | Analog | PWR |
| AH7 | VSS | Analog | PWR |
| AH6 | VSS | Analog | PWR |
| AH3 | VSS | Analog | PWR |
| AG34 | VSS | Analog | PWR |
| AG31 | VSS | Analog | PWR |
| AG28 | VSS | Analog | PWR |
| AG25 | VSS | Analog | PWR |
| AG24 | VSS | Analog | PWR |
| AG23 | VSS | Analog | PWR |
| AG22 | VSS | Analog | PWR |
| AG21 | VSS | Analog | PWR |

**Table 22-2. Pin Listing by Signal Name
(Sheet 24 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AG20 | VSS | Analog | PWR |
| AG19 | VSS | Analog | PWR |
| AG18 | VSS | Analog | PWR |
| AG17 | VSS | Analog | PWR |
| AG16 | VSS | Analog | PWR |
| AG15 | VSS | Analog | PWR |
| AG14 | VSS | Analog | PWR |
| AG13 | VSS | Analog | PWR |
| AG12 | VSS | Analog | PWR |
| AG11 | VSS | Analog | PWR |
| AG10 | VSS | Analog | PWR |
| AG7 | VSS | Analog | PWR |
| AG4 | VSS | Analog | PWR |
| AG1 | VSS | Analog | PWR |
| AF36 | VSS | Analog | PWR |
| AF33 | VSS | Analog | PWR |
| AF30 | VSS | Analog | PWR |
| AF27 | VSS | Analog | PWR |
| AF25 | VSS | Analog | PWR |
| AF24 | VSS | Analog | PWR |
| AF18 | VSS | Analog | PWR |
| AF16 | VSS | Analog | PWR |
| AF14 | VSS | Analog | PWR |
| AF11 | VSS | Analog | PWR |
| AF8 | VSS | Analog | PWR |
| AF5 | VSS | Analog | PWR |
| AF2 | VSS | Analog | PWR |
| AE35 | VSS | Analog | PWR |
| AE32 | VSS | Analog | PWR |
| AE29 | VSS | Analog | PWR |
| AE26 | VSS | Analog | PWR |
| AE11 | VSS | Analog | PWR |
| AE9 | VSS | Analog | PWR |
| AE6 | VSS | Analog | PWR |
| AE3 | VSS | Analog | PWR |
| AD34 | VSS | Analog | PWR |
| AD31 | VSS | Analog | PWR |
| AD28 | VSS | Analog | PWR |
| AD11 | VSS | Analog | PWR |
| AD7 | VSS | Analog | PWR |



**Table 22-2. Pin Listing by Signal Name
(Sheet 25 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AD4 | VSS | Analog | PWR |
| AD1 | VSS | Analog | PWR |
| AC36 | VSS | Analog | PWR |
| AC33 | VSS | Analog | PWR |
| AC30 | VSS | Analog | PWR |
| AC27 | VSS | Analog | PWR |
| AC24 | VSS | Analog | PWR |
| AC23 | VSS | Analog | PWR |
| AC21 | VSS | Analog | PWR |
| AC18 | VSS | Analog | PWR |
| AC16 | VSS | Analog | PWR |
| AC14 | VSS | Analog | PWR |
| AC11 | VSS | Analog | PWR |
| AC8 | VSS | Analog | PWR |
| AC5 | VSS | Analog | PWR |
| AC2 | VSS | Analog | PWR |
| AB35 | VSS | Analog | PWR |
| AB32 | VSS | Analog | PWR |
| AB29 | VSS | Analog | PWR |
| AB26 | VSS | Analog | PWR |
| AB22 | VSS | Analog | PWR |
| AB20 | VSS | Analog | PWR |
| AB18 | VSS | Analog | PWR |
| AB16 | VSS | Analog | PWR |
| AB14 | VSS | Analog | PWR |
| AB13 | VSS | Analog | PWR |
| AB12 | VSS | Analog | PWR |
| AB11 | VSS | Analog | PWR |
| AB9 | VSS | Analog | PWR |
| AB6 | VSS | Analog | PWR |
| AB3 | VSS | Analog | PWR |
| AA34 | VSS | Analog | PWR |
| AA31 | VSS | Analog | PWR |
| AA28 | VSS | Analog | PWR |
| AA25 | VSS | Analog | PWR |
| AA23 | VSS | Analog | PWR |
| AA21 | VSS | Analog | PWR |
| AA19 | VSS | Analog | PWR |
| AA17 | VSS | Analog | PWR |
| AA11 | VSS | Analog | PWR |

**Table 22-2. Pin Listing by Signal Name
(Sheet 26 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AA10 | VSS | Analog | PWR |
| AA7 | VSS | Analog | PWR |
| AA4 | VSS | Analog | PWR |
| AA1 | VSS | Analog | PWR |
| Y36 | VSS | Analog | PWR |
| Y33 | VSS | Analog | PWR |
| Y30 | VSS | Analog | PWR |
| Y27 | VSS | Analog | PWR |
| Y24 | VSS | Analog | PWR |
| Y22 | VSS | Analog | PWR |
| Y20 | VSS | Analog | PWR |
| Y18 | VSS | Analog | PWR |
| Y16 | VSS | Analog | PWR |
| Y13 | VSS | Analog | PWR |
| Y12 | VSS | Analog | PWR |
| Y11 | VSS | Analog | PWR |
| Y8 | VSS | Analog | PWR |
| Y5 | VSS | Analog | PWR |
| Y2 | VSS | Analog | PWR |
| W35 | VSS | Analog | PWR |
| W32 | VSS | Analog | PWR |
| W29 | VSS | Analog | PWR |
| W26 | VSS | Analog | PWR |
| W25 | VSS | Analog | PWR |
| W23 | VSS | Analog | PWR |
| W21 | VSS | Analog | PWR |
| W19 | VSS | Analog | PWR |
| W17 | VSS | Analog | PWR |
| W9 | VSS | Analog | PWR |
| W6 | VSS | Analog | PWR |
| W3 | VSS | Analog | PWR |
| V34 | VSS | Analog | PWR |
| V31 | VSS | Analog | PWR |
| V28 | VSS | Analog | PWR |
| V25 | VSS | Analog | PWR |
| V22 | VSS | Analog | PWR |
| V20 | VSS | Analog | PWR |
| V18 | VSS | Analog | PWR |
| V16 | VSS | Analog | PWR |
| V13 | VSS | Analog | PWR |



**Table 22-2. Pin Listing by Signal Name
(Sheet 27 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| V12 | VSS | Analog | PWR |
| V11 | VSS | Analog | PWR |
| V10 | VSS | Analog | PWR |
| V7 | VSS | Analog | PWR |
| V4 | VSS | Analog | PWR |
| V1 | VSS | Analog | PWR |
| U36 | VSS | Analog | PWR |
| U33 | VSS | Analog | PWR |
| U30 | VSS | Analog | PWR |
| U27 | VSS | Analog | PWR |
| U24 | VSS | Analog | PWR |
| U23 | VSS | Analog | PWR |
| U21 | VSS | Analog | PWR |
| U19 | VSS | Analog | PWR |
| U17 | VSS | Analog | PWR |
| U7 | VSS | Analog | PWR |
| U4 | VSS | Analog | PWR |
| U1 | VSS | Analog | PWR |
| T35 | VSS | Analog | PWR |
| T32 | VSS | Analog | PWR |
| T29 | VSS | Analog | PWR |
| T26 | VSS | Analog | PWR |
| T22 | VSS | Analog | PWR |
| T20 | VSS | Analog | PWR |
| T18 | VSS | Analog | PWR |
| T16 | VSS | Analog | PWR |
| T13 | VSS | Analog | PWR |
| T12 | VSS | Analog | PWR |
| T11 | VSS | Analog | PWR |
| T9 | VSS | Analog | PWR |
| T6 | VSS | Analog | PWR |
| T3 | VSS | Analog | PWR |
| R34 | VSS | Analog | PWR |
| R31 | VSS | Analog | PWR |
| R28 | VSS | Analog | PWR |
| R25 | VSS | Analog | PWR |
| R23 | VSS | Analog | PWR |
| R21 | VSS | Analog | PWR |
| R19 | VSS | Analog | PWR |
| R17 | VSS | Analog | PWR |

**Table 22-2. Pin Listing by Signal Name
(Sheet 28 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| R15 | VSS | Analog | PWR |
| R11 | VSS | Analog | PWR |
| R10 | VSS | Analog | PWR |
| R8 | VSS | Analog | PWR |
| R5 | VSS | Analog | PWR |
| R2 | VSS | Analog | PWR |
| P36 | VSS | Analog | PWR |
| P33 | VSS | Analog | PWR |
| P30 | VSS | Analog | PWR |
| P27 | VSS | Analog | PWR |
| P24 | VSS | Analog | PWR |
| P22 | VSS | Analog | PWR |
| P20 | VSS | Analog | PWR |
| P18 | VSS | Analog | PWR |
| P16 | VSS | Analog | PWR |
| P14 | VSS | Analog | PWR |
| P11 | VSS | Analog | PWR |
| P7 | VSS | Analog | PWR |
| P4 | VSS | Analog | PWR |
| P1 | VSS | Analog | PWR |
| N35 | VSS | Analog | PWR |
| N32 | VSS | Analog | PWR |
| N29 | VSS | Analog | PWR |
| N26 | VSS | Analog | PWR |
| N23 | VSS | Analog | PWR |
| N21 | VSS | Analog | PWR |
| N20 | VSS | Analog | PWR |
| N10 | VSS | Analog | PWR |
| N9 | VSS | Analog | PWR |
| N6 | VSS | Analog | PWR |
| N3 | VSS | Analog | PWR |
| M34 | VSS | Analog | PWR |
| M31 | VSS | Analog | PWR |
| M28 | VSS | Analog | PWR |
| M25 | VSS | Analog | PWR |
| M22 | VSS | Analog | PWR |
| M10 | VSS | Analog | PWR |
| M8 | VSS | Analog | PWR |
| M5 | VSS | Analog | PWR |
| M2 | VSS | Analog | PWR |



**Table 22-2. Pin Listing by Signal Name
(Sheet 29 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| L36 | VSS | Analog | PWR |
| L33 | VSS | Analog | PWR |
| L30 | VSS | Analog | PWR |
| L27 | VSS | Analog | PWR |
| L24 | VSS | Analog | PWR |
| L23 | VSS | Analog | PWR |
| L21 | VSS | Analog | PWR |
| L18 | VSS | Analog | PWR |
| L16 | VSS | Analog | PWR |
| L14 | VSS | Analog | PWR |
| L12 | VSS | Analog | PWR |
| L11 | VSS | Analog | PWR |
| L7 | VSS | Analog | PWR |
| L4 | VSS | Analog | PWR |
| L1 | VSS | Analog | PWR |
| K35 | VSS | Analog | PWR |
| K32 | VSS | Analog | PWR |
| K29 | VSS | Analog | PWR |
| K26 | VSS | Analog | PWR |
| K18 | VSS | Analog | PWR |
| K16 | VSS | Analog | PWR |
| K14 | VSS | Analog | PWR |
| K9 | VSS | Analog | PWR |
| K6 | VSS | Analog | PWR |
| K3 | VSS | Analog | PWR |
| J34 | VSS | Analog | PWR |
| J31 | VSS | Analog | PWR |
| J28 | VSS | Analog | PWR |
| J25 | VSS | Analog | PWR |
| J22 | VSS | Analog | PWR |
| J20 | VSS | Analog | PWR |
| J18 | VSS | Analog | PWR |
| J17 | VSS | Analog | PWR |
| J15 | VSS | Analog | PWR |
| J12 | VSS | Analog | PWR |
| J9 | VSS | Analog | PWR |
| J8 | VSS | Analog | PWR |
| J5 | VSS | Analog | PWR |
| J2 | VSS | Analog | PWR |
| H36 | VSS | Analog | PWR |

**Table 22-2. Pin Listing by Signal Name
(Sheet 30 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| H33 | VSS | Analog | PWR |
| H30 | VSS | Analog | PWR |
| H27 | VSS | Analog | PWR |
| H25 | VSS | Analog | PWR |
| H22 | VSS | Analog | PWR |
| H19 | VSS | Analog | PWR |
| H14 | VSS | Analog | PWR |
| H11 | VSS | Analog | PWR |
| H8 | VSS | Analog | PWR |
| H7 | VSS | Analog | PWR |
| H4 | VSS | Analog | PWR |
| H1 | VSS | Analog | PWR |
| G35 | VSS | Analog | PWR |
| G32 | VSS | Analog | PWR |
| G29 | VSS | Analog | PWR |
| G26 | VSS | Analog | PWR |
| G23 | VSS | Analog | PWR |
| G20 | VSS | Analog | PWR |
| G17 | VSS | Analog | PWR |
| G16 | VSS | Analog | PWR |
| G13 | VSS | Analog | PWR |
| G10 | VSS | Analog | PWR |
| G7 | VSS | Analog | PWR |
| G6 | VSS | Analog | PWR |
| G3 | VSS | Analog | PWR |
| F34 | VSS | Analog | PWR |
| F31 | VSS | Analog | PWR |
| F28 | VSS | Analog | PWR |
| F27 | VSS | Analog | PWR |
| F24 | VSS | Analog | PWR |
| F21 | VSS | Analog | PWR |
| F18 | VSS | Analog | PWR |
| F15 | VSS | Analog | PWR |
| F12 | VSS | Analog | PWR |
| F9 | VSS | Analog | PWR |
| F6 | VSS | Analog | PWR |
| F5 | VSS | Analog | PWR |
| F2 | VSS | Analog | PWR |
| E36 | VSS | Analog | PWR |
| E33 | VSS | Analog | PWR |



**Table 22-2. Pin Listing by Signal Name
(Sheet 31 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| E30 | VSS | Analog | PWR |
| E28 | VSS | Analog | PWR |
| E25 | VSS | Analog | PWR |
| E22 | VSS | Analog | PWR |
| E19 | VSS | Analog | PWR |
| E14 | VSS | Analog | PWR |
| E11 | VSS | Analog | PWR |
| E8 | VSS | Analog | PWR |
| E5 | VSS | Analog | PWR |
| E4 | VSS | Analog | PWR |
| E1 | VSS | Analog | PWR |
| D35 | VSS | Analog | PWR |
| D32 | VSS | Analog | PWR |
| D29 | VSS | Analog | PWR |
| D26 | VSS | Analog | PWR |
| D23 | VSS | Analog | PWR |
| D20 | VSS | Analog | PWR |
| D17 | VSS | Analog | PWR |
| D16 | VSS | Analog | PWR |
| D13 | VSS | Analog | PWR |
| D10 | VSS | Analog | PWR |
| D7 | VSS | Analog | PWR |
| D4 | VSS | Analog | PWR |
| D3 | VSS | Analog | PWR |
| C36 | VSS | Analog | PWR |
| C34 | VSS | Analog | PWR |
| C31 | VSS | Analog | PWR |
| C30 | VSS | Analog | PWR |
| C27 | VSS | Analog | PWR |
| C24 | VSS | Analog | PWR |
| C21 | VSS | Analog | PWR |
| C18 | VSS | Analog | PWR |
| C15 | VSS | Analog | PWR |
| C12 | VSS | Analog | PWR |
| C9 | VSS | Analog | PWR |
| C6 | VSS | Analog | PWR |
| C3 | VSS | Analog | PWR |
| C2 | VSS | Analog | PWR |
| C1 | VSS | Analog | PWR |
| B36 | VSS | Analog | PWR |

**Table 22-2. Pin Listing by Signal Name
(Sheet 32 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| B35 | VSS | Analog | PWR |
| B34 | VSS | Analog | PWR |
| B33 | VSS | Analog | PWR |
| B31 | VSS | Analog | PWR |
| B28 | VSS | Analog | PWR |
| B25 | VSS | Analog | PWR |
| B22 | VSS | Analog | PWR |
| B19 | VSS | Analog | PWR |
| B14 | VSS | Analog | PWR |
| B11 | VSS | Analog | PWR |
| B8 | VSS | Analog | PWR |
| B5 | VSS | Analog | PWR |
| B3 | VSS | Analog | PWR |
| B2 | VSS | Analog | PWR |
| A35 | VSS | Analog | PWR |
| A34 | VSS | Analog | PWR |
| A33 | VSS | Analog | PWR |
| A32 | VSS | Analog | PWR |
| A29 | VSS | Analog | PWR |
| A26 | VSS | Analog | PWR |
| A23 | VSS | Analog | PWR |
| A20 | VSS | Analog | PWR |
| A17 | VSS | Analog | PWR |
| A16 | VSS | Analog | PWR |
| A13 | VSS | Analog | PWR |
| A10 | VSS | Analog | PWR |
| A7 | VSS | Analog | PWR |
| A4 | VSS | Analog | PWR |
| A3 | VSS | Analog | PWR |
| R24 | VTTDDR | Analog | PWR |
| U25 | VTTXDP | Analog | PWR |
| AG36 | XDPCLK1XN | DDR | O |
| AH36 | XDPCLK1XP | DDR | O |
| W33 | XDPDQ[0] | DDR | I/O |
| W36 | XDPDQ[1] | DDR | I/O |
| AA36 | XDPDQ[10] | DDR | I/O |
| AA33 | XDPDQ[11] | DDR | I/O |
| AE36 | XDPDQ[12] | DDR | I/O |
| AC34 | XDPDQ[13] | DDR | I/O |
| AB36 | XDPDQ[14] | DDR | I/O |

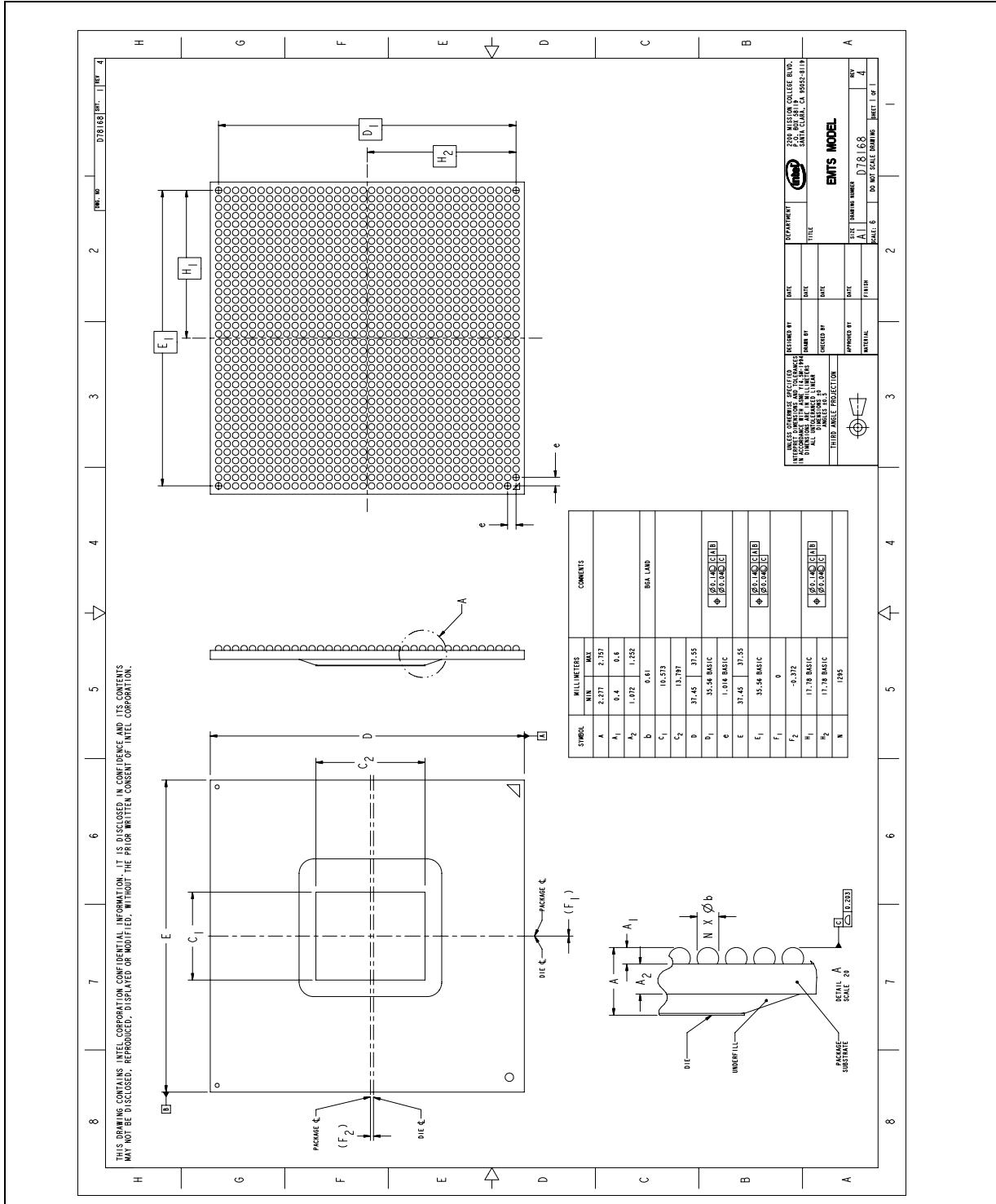


**Table 22-2. Pin Listing by Signal Name
(Sheet 33 of 33)**

| Pin Name | Signal Name | Signal Buffer Type | Direction |
|----------|-------------|--------------------|-----------|
| AB34 | XDPDQ[15] | DDR | I/O |
| V33 | XDPDQ[2] | DDR | I/O |
| V36 | XDPDQ[3] | DDR | I/O |
| Y34 | XDPDQ[4] | DDR | I/O |
| V35 | XDPDQ[5] | DDR | I/O |
| W34 | XDPDQ[6] | DDR | I/O |
| U35 | XDPDQ[7] | DDR | I/O |
| AD36 | XDPDQ[8] | DDR | O |
| AB33 | XDPDQ[9] | DDR | I/O |
| Y35 | XDPDQSN[0] | DDR | I/O |
| AC35 | XDPDQSN[1] | DDR | I/O |
| AA35 | XDPDQSP[0] | DDR | I/O |
| AD35 | XDPDQSP[1] | DDR | I/O |
| Y31 | XDPRDYACK_N | DDR | O |
| W31 | XDPRDYREQ_N | DDR | I |

22.3 Package Information

Figure 22-5. Package Diagram



Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

[Intel:](#)

[AC7500BXB S LH3K](#) [BD7500MB S LH2F](#)

Данный компонент на территории Российской Федерации

Вы можете приобрести в компании MosChip.

Для оперативного оформления запроса Вам необходимо перейти по данной ссылке:

<http://moschip.ru/get-element>

Вы можете разместить у нас заказ для любого Вашего проекта, будь то серийное производство или разработка единичного прибора.

В нашем ассортименте представлены ведущие мировые производители активных и пассивных электронных компонентов.

Нашей специализацией является поставка электронной компонентной базы двойного назначения, продукции таких производителей как XILINX, Intel (ex.ALTERA), Vicor, Microchip, Texas Instruments, Analog Devices, Mini-Circuits, Amphenol, Glenair.

Сотрудничество с глобальными дистрибьюторами электронных компонентов, предоставляет возможность заказывать и получать с международных складов практически любой перечень компонентов в оптимальные для Вас сроки.

На всех этапах разработки и производства наши партнеры могут получить квалифицированную поддержку опытных инженеров.

Система менеджмента качества компании отвечает требованиям в соответствии с ГОСТ Р ИСО 9001, ГОСТ РВ 0015-002 и ЭС РД 009

Офис по работе с юридическими лицами:

105318, г.Москва, ул.Щербаковская д.3, офис 1107, 1118, ДЦ «Щербаковский»

Телефон: +7 495 668-12-70 (многоканальный)

Факс: +7 495 668-12-70 (доб.304)

E-mail: info@moschip.ru

Skype отдела продаж:

moschip.ru

moschip.ru_4

moschip.ru_6

moschip.ru_9