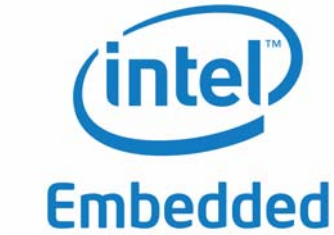


Intel® Xeon® Processor C5500/ C3500 Series

Datasheet - Volume 1

February 2010

Order Number: 323103-001

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. Intel products are not intended for use in medical, life saving, 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 information here is subject to change without notice. Do not finalize a design with this information.

The products described in this document 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.

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 processor numbers are not a measure of performance. Processor numbers differentiate features within each processor family, not across different processor families. See http://www.intel.com/products/processor_number for details.

Code names are only for use by Intel to identify products, platforms, programs, services, etc. ("products") in development by Intel that have not been made commercially available to the public, i.e., announced, launched or shipped. They are never to be used as "commercial" names for products. Also, they are not intended to function as trademarks.

BunnyPeople, Celeron, Celeron Inside, Centrino, Centrino logo, Core Inside, FlashFile, i960, InstantIP, Intel, Intel logo, Intel386, Intel486, Intel740, IntelDX2, IntelDX4, IntelSX2, Intel Core, Intel Inside, Intel Inside logo, Intel. Leap ahead., Intel. Leap ahead. logo, Intel NetBurst, Intel NetMerge, Intel NetStructure, Intel SingleDriver, Intel SpeedStep, Intel StrataFlash, Intel Viiv, Intel vPro, Intel XScale, Itanium, Itanium Inside, MCS, MMX, Oplus, OverDrive, PDCharm, Pentium, Pentium Inside, skoool, Sound Mark, The Journey Inside, VTune, Xeon, and Xeon Inside are trademarks of Intel Corporation in the U.S. and other countries.

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

Copyright © 2010, Intel Corporation. All rights reserved.

Contents

1.0 Features Summary 24
- 1.1 Introduction 24
- 1.2 Processor Feature Details 27
  - 1.2.1 Supported Technologies 27
- 1.3 SKUs 27
- 1.4 Interfaces 28
  - 1.4.1 Intel® QuickPath Interconnect (Intel® QPI) 28
  - 1.4.2 System Memory Support 28
  - 1.4.3 PCI Express 29
  - 1.4.4 Direct Media Interface (DMI) 30
  - 1.4.5 Platform Environment Control Interface (PECI) 30
  - 1.4.6 SMBus 30
- 1.5 Power Management Support 31
  - 1.5.1 Processor Core 31
  - 1.5.2 System 31
  - 1.5.3 Memory Controller 31
  - 1.5.4 PCI Express 31
  - 1.5.5 DMI 31
  - 1.5.6 Intel® QuickPath Interconnect 31
- 1.6 Thermal Management Support 31
- 1.7 Package 31
- 1.8 Terminology 32
- 1.9 Related Documents 33

2.0 Interfaces 35
- 2.1 System Memory Interface 35
  - 2.1.1 System Memory Technology Supported 35
  - 2.1.2 System Memory DIMM Configuration Support 36
  - 2.1.3 System Memory Timing Support 37
    - 2.1.3.1 System Memory Operating Modes 38
    - 2.1.3.2 Single-Channel Mode 39
    - 2.1.3.3 Independent Channel Mode 39
    - 2.1.3.4 Spare Channel Mode 40
    - 2.1.3.5 Mirrored Channel Mode 41
    - 2.1.3.6 Lockstep Mode 42
    - 2.1.3.7 Dual/Triple - Channel Modes 43
  - 2.1.4 DIMM Population Requirements 45
    - 2.1.4.1 General Population Requirements 45
    - 2.1.4.2 Populating DIMMs Within a Channel 45
    - 2.1.4.3 Channel Population Requirements for Memory RAS Modes 48
  - 2.1.5 Technology Enhancements of Intel® Fast Memory Access (Intel® FMA) 48
    - 2.1.5.1 Just-in-Time Command Scheduling 48
    - 2.1.5.2 Command Overlap 49
    - 2.1.5.3 Out-of-Order Scheduling 49
  - 2.1.6 DDR3 On-Die Termination 49
  - 2.1.7 Memory Error Signaling 49
    - 2.1.7.1 Enabling SMI/NMI for Memory Corrected Errors 50
    - 2.1.7.2 Per DIMM Error Counters 50
    - 2.1.7.3 Identifying the Cause of An Interrupt 51
  - 2.1.8 Single Device Data Correction (SDDC) Support 51
  - 2.1.9 Patrol Scrub 51
  - 2.1.10 Memory Address Decode 52
    - 2.1.10.1 First Level Decode 52

2.1.10.2 Second Level Address Translation 54
2.1.11 Address Translations 55
2.1.11.1 Translating System Address to Channel Address 55
2.1.11.2 Translating Channel Address to Rank Address 56
2.1.11.3 Low Order Address Bit Mapping 56
2.1.11.4 Supported Configurations 58
2.1.12 DDR Protocol Support 58
2.1.13 Refresh 58
2.1.13.1 DRAM Driver Impedance Calibration 58
2.1.14 Power Management 59
2.1.14.1 Interface to Uncore Power Manager 59
2.1.14.2 DRAM Power Down States 59
2.1.14.3 Dynamic DRAM Interface Power Savings Features 60
2.1.14.4 Static DRAM Interface Power Savings Features 61
2.1.14.5 DRAM Temperature Throttling 61
2.1.14.6 Closed Loop Thermal Throttling (CLTT) 64
2.1.14.7 Advanced Throttling Options 65
2.1.14.8 2X Refresh 65
2.1.14.9 Demand Observation 66
2.1.14.10 Rank Sharing 67
2.1.14.11 Registers 67
2.2 Platform Environment Control Interface (PECI) 69
2.2.1 PECI Client Capabilities 70
2.2.1.1 Thermal Management 70
2.2.1.2 Platform Manageability 71
2.2.1.3 Processor Interface Tuning and Diagnostics 71
2.2.2 Client Command Suite 71
2.2.2.1 Ping() 71
2.2.2.2 GetDIB() 72
2.2.2.3 GetTemp() 73
2.2.2.4 PCIConfigRd() 74
2.2.2.5 PCIConfigWr() 76
2.2.2.6 Mailbox 77
2.2.2.7 MbxSend() 82
2.2.2.8 MbxGet() 84
2.2.2.9 Mailbox Usage Definition 85
2.2.3 Multi-Domain Commands 86
2.2.4 Client Responses 87
2.2.4.1 Abort FCS 87
2.2.4.2 Completion Codes 87
2.2.5 Originator Responses 88
2.2.6 Temperature Data 89
2.2.6.1 Format 89
2.2.6.2 Interpretation 89
2.2.6.3 Temperature Filtering 89
2.2.6.4 Reserved Values 89
2.2.7 Client Management 90
2.2.7.1 Power-up Sequencing 90
2.2.7.2 Device Discovery 91
2.2.7.3 Client Addressing 91
2.2.7.4 C-States 91
2.2.7.5 S-States 91
2.2.7.6 Processor Reset 92
2.3 SMBus 92
2.3.1 Slave SMBus 92
2.3.2 Master SMBus 93
2.3.3 SMBus Physical Layer 93
2.3.4 SMBus Supported Transactions 93

2.3.5 Addressing 95
2.3.6 SMBus Initiated Southbound Configuration Cycles 97
2.3.7 SMBus Error Handling 97
2.3.8 SMBus Interface Reset 97
2.3.9 Configuration and Memory Read Protocol 98
2.3.9.1 SMBus Configuration and Memory Block-Size Reads 99
2.3.9.2 SMBus Configuration and Memory Word-Size Reads 100
2.3.9.3 SMBus Configuration and Memory Byte Reads 101
2.3.9.4 Configuration and Memory Write Protocol 103
2.3.9.5 SMBus Configuration and Memory Block Writes 103
2.3.9.6 SMBus Configuration and Memory Word Writes 104
2.3.9.7 SMBus Configuration and Memory Byte Writes 104
2.4 Intel® QuickPath Interconnect (Intel® QPI) 105
2.4.1 Processor's Intel® QuickPath Interconnect Platform Overview 105
2.4.2 Physical Layer Implementation 107
2.4.2.1 Processor's Intel® QuickPath Interconnect Physical Layer Attributes 107
2.4.3 Processor's Intel® QuickPath Interconnect Link Speed Configuration 107
2.4.3.1 Detect Intel® QuickPath Interconnect Speeds Supported by the Processors 107
2.4.4 Intel® QuickPath Interconnect Probing Considerations 108
2.4.5 Link Layer 108
2.4.5.1 Link Layer Attributes 108
2.4.6 Routing Layer 108
2.4.6.1 Routing Layer Attributes 108
2.4.7 Intel® QuickPath Interconnect Address Decoding 109
2.4.8 Transport Layer 109
2.4.9 Protocol Layer 109
2.4.9.1 Protocol Layer Attributes 109
2.4.9.2 Intel® QuickPath Interconnect Coherent Protocol Attributes 110
2.4.9.3 Intel® QuickPath Interconnect Non-Coherent Protocol Attributes 110
2.4.9.4 Interrupt Handling 110
2.4.9.5 Fault Handling 111
2.4.9.6 Reset/Initialization 111
2.4.9.7 Other Attributes 111
2.5 IIO Intel® QPI Coherent Interface and Address Decode 111
2.5.1 Introduction 111
2.5.2 Link Layer 112
2.5.2.1 Link Error Protection 112
2.5.2.2 Message Class 112
2.5.2.3 Link-Level Credit Return Policy 112
2.5.2.4 Ordering 112
2.5.3 Protocol Layer 113
2.5.4 Snooping Modes 113
2.5.5 IIO Source Address Decoder (SAD) 113
2.5.5.1 NodeID Generation 114
2.5.5.2 Memory Decoder 114
2.5.5.3 I/O Decoder 114
2.5.6 Special Response Status 115
2.5.7 Illegal Completion/Response/Request 115
2.5.8 Inbound Coherent 116
2.5.9 Inbound Non-Coherent 116
2.5.9.1 Peer-to-Peer Tunneling 116
2.5.10 Profile Support 116
2.5.11 Write Cache 117
2.5.11.1 Write Cache Depth 117

2.5.11.2 Coherent Write Flow 117
2.5.11.3 Eviction Policy 117
2.5.12 Outgoing Request Buffer (ORB) 118
2.5.13 Time-Out Counter 118
2.6 PCI Express Interface 119
2.6.1 PCI Express Architecture 119
2.6.1.1 Transaction Layer 120
2.6.1.2 Data Link Layer 120
2.6.1.3 Physical Layer 120
2.6.2 PCI Express Link Characteristics - Link Training, Bifurcation, Downgrading and Lane Reversal Support 120
2.6.2.1 Link Training 120
2.6.2.2 Port Bifurcation 121
2.6.2.3 Port Bifurcation via BIOS 121
2.6.2.4 Degraded Mode 122
2.6.2.5 Lane Reversal 123
2.6.3 Gen1/Gen2 Speed Selection 123
2.6.4 Link Upconfigure Capability 123
2.6.5 Error Reporting 123
2.6.5.1 Chipset-Specific Vendor-Defined 123
2.6.5.2 ASSERT_GPE / DEASSERT_GPE 124
2.6.6 Configuration Retry Completions 124
2.6.7 Inbound Transactions 125
2.6.7.1 Inbound PCI Express Messages Supported 125
2.6.8 Outbound Transactions 126
2.6.8.1 Memory, I/O and Configuration Transactions Supported 126
2.6.9 Lock Support 126
2.6.10 Outbound Messages Supported 127
2.6.10.1 Unlock 127
2.6.10.2 EOI 127
2.6.11 32/64 bit Addressing 127
2.6.12 Transaction Descriptor 128
2.6.12.1 Transaction ID 128
2.6.12.2 Attributes 129
2.6.12.3 Traffic Class 129
2.6.13 Completer ID 129
2.6.14 Miscellaneous 129
2.6.14.1 Number of Outbound Non-posted Requests 129
2.6.14.2 MSIs Generated from Root Ports and Locks 129
2.6.14.3 Completions for Locked Read Requests 130
2.6.15 PCI Express RAS 130
2.6.16 ECRC Support 130
2.6.17 Completion Timeout 130
2.6.18 Data Poisoning 130
2.6.19 Role-Based Error Reporting 130
2.6.20 Data Link Layer Specifics 131
2.6.20.1 Ack/Nak 131
2.6.20.2 Link Level Retry 131
2.6.21 Ack Time-out 131
2.6.22 Flow Control 131
2.6.22.1 Flow Control Credit Return by IIO 133
2.6.22.2 FC Update DLLP Timeout 133
2.6.23 Physical Layer Specifics 133
2.6.23.1 Polarity Inversion 133
2.6.24 Non-Transparent Bridge 133
2.7 Direct Media Interface (DMI2) 134

2.7.1 DMI Error Flow 134
2.7.2 Processor/PCH Compatibility Assumptions 134
2.7.3 DMI Link Down 134

3.0 PCI Express Non-Transparent Bridge 135
3.1 Introduction 135
3.2 NTB Features Supported on Intel® Xeon® Processor C5500/C3500 Series 135
3.2.1 Features Not Supported on the Intel® Xeon® Processor C5500/C3500 Series NTB 136
3.3 Non-Transparent Bridge vs. Transparent Bridge 136
3.4 NTB Support in Intel® Xeon® Processor C5500/C3500 Series 139
3.5 NTB Supported Configurations 139
3.5.1 Connecting Intel® Xeon® Processor C5500/C3500 Series Systems Back-to-Back with NTB Ports 139
3.5.2 Connecting NTB Port on Intel® Xeon® Processor C5500/C3500 Series to Root Port on Another Intel® Xeon® Processor C5500/C3500 Series System - Symmetric Configuration 140
3.5.3 Connecting NTB Port on Intel® Xeon® Processor C5500/C3500 Series to Root Port on Another System - Non-Symmetric Configuration 141
3.6 Architecture Overview 143
3.6.1 “A Priori” Configuration Knowledge 146
3.6.2 Power On Sequence for RP and NTB 146
3.6.3 Crosslink Configuration 146
3.6.4 B2B BAR and Translate Setup 149
3.6.5 Enumeration and Power Sequence 150
3.6.6 Address Translation 152
3.6.6.1 Direct Address Translation 152
3.6.7 Requester ID Translation 155
3.6.8 Peer-to-Peer Across NTB Bridge 157
3.7 NTB Inbound Transactions 158
3.7.1 Memory, I/O and Configuration Transactions 158
3.7.2 Inbound PCI Express Messages Supported 159
3.7.2.1 Error Reporting 159
3.8 Outbound Transactions 160
3.8.1 Memory, I/O and Configuration Transactions 160
3.8.2 Lock Support 161
3.8.3 Outbound Messages Supported 161
3.8.3.1 EOI 163
3.9 32-/64-Bit Addressing 163
3.10 Transaction Descriptor 163
3.10.1 Transaction ID 163
3.10.2 Attributes 164
3.10.3 Traffic Class 165
3.11 Completer ID 165
3.12 Initialization 165
3.12.1 Initialization Sequence with NTB Ports Connected Back-to-Back (NTB/NTB) 165
3.12.2 Initialization Sequence with NTB Port Connected to Root Port 166
3.13 Reset Requirements 167
3.14 Power Management 167
3.15 Scratch Pad and Doorbell Registers 167
3.16 MSI-X Vector Mapping 169
3.17 RAS Capability and Error Handling 169
3.18 Registers and Register Description 169
3.18.1 Additional Registers Outside of NTB Required (Per Stepping) 169
3.18.2 Known Errata (Per Stepping) 169
3.18.3 Bring Up Help 170

3.19 PCI Express Configuration Registers (NTB Primary Side) 170
3.19.1 Configuration Register Map (NTB Primary Side) 170
3.19.2 Standard PCI Configuration Space (0x0 to 0x3F) - Type 0 Common Configuration Space 175
3.19.2.1 VID: Vendor Identification Register 175
3.19.2.2 DID: Device Identification Register (Dev#3, PCIE NTB Pri Mode) .. 175
3.19.2.3 PCICMD: PCI Command Register (Dev#3, PCIE NTB Pri Mode) 176
3.19.2.4 PCISTS: PCI Status Register 178
3.19.2.5 RID: Revision Identification Register 180
3.19.2.6 CCR: Class Code Register 180
3.19.2.7 CLSR: Cacheline Size Register 181
3.19.2.8 PLAT: Primary Latency Timer 181
3.19.2.9 HDR: Header Type Register (Dev#3, PCIe NTB Pri Mode) 181
3.19.2.10 BIST: Built-In Self Test 182
3.19.2.11 PB01BASE: Primary BAR 0/1 Base Address 182
3.19.2.12 PB23BASE: Primary BAR 2/3 Base Address 183
3.19.2.13 PB45BASE: Primary BAR 4/5 Base Address 184
3.19.2.14 SUBVID: Subsystem Vendor ID (Dev#3, PCIE NTB Pri Mode) 184
3.19.2.15 SID: Subsystem Identity (Dev#3, PCIE NTB Pri Mode) 185
3.19.2.16 CAPPTR: Capability Pointer 185
3.19.2.17 INTL: Interrupt Line Register 185
3.19.2.18 INTPIN: Interrupt Pin Register 186
3.19.2.19 MINGNT: Minimum Grant Register 186
3.19.2.20 MAXLAT: Maximum Latency Register 186
3.19.3 Device-Specific PCI Configuration Space - 0x40 to 0xFF 187
3.19.3.1 MSICAPID: MSI Capability ID 187
3.19.3.2 MSINXTPTR: MSI Next Pointer 187
3.19.3.3 MSICTRL: MSI Control Register 187
3.19.3.4 MSIAR: MSI Address Register 189
3.19.3.5 MSIDR: MSI Data Register 190
3.19.3.6 MSIMSK: MSI Mask Bit Register 191
3.19.3.7 MSIPENDING: MSI Pending Bit Register 191
3.19.3.8 MSIXCAPID: MSI-X Capability ID 191
3.19.3.9 MSIXNXTPTR: MSI-X Next Pointer 192
3.19.3.10 MSIXMSGCTRL: MSI-X Message Control Register 192
3.19.3.11 TABLEOFF_BIR: MSI-X Table Offset and BAR Indicator Register (BIR) 193
3.19.3.12 PBAOFF_BIR: MSI-X Pending Array Offset and BAR Indicator 193
3.19.3.13 PXPCAPID: PCI Express Capability Identity Register 194
3.19.3.14 PXPNXTPTR: PCI Express Next Pointer Register 194
3.19.3.15 PXPCAP: PCI Express Capabilities Register 195
3.19.3.16 DEVCAP: PCI Express Device Capabilities Register 196
3.19.3.17 DEVCTRL: PCI Express Device Control Register (Dev#3, PCIE NTB Pri Mode) 198
3.19.3.18 DEVSTS: PCI Express Device Status Register 200
3.19.3.19 PBAR23SZ: Primary BAR 2/3 Size 201
3.19.3.20 PBAR45SZ: Primary BAR 4/5 Size 201
3.19.3.21 SBAR23SZ: Secondary BAR 2/3 Size 202
3.19.3.22 SBAR45SZ: Secondary BAR 4/5 Size 202
3.19.3.23 PPD: PCIE Port Definition 203
3.19.3.24 PMCAP: Power Management Capabilities Register 204
3.19.3.25 PMCSR: Power Management Control and Status Register 205
3.19.4 PCI Express Enhanced Configuration Space 206
3.19.4.1 VSECPHDR: Vendor Specific Enhanced Capability Header 206
3.19.4.2 VSHDR: Vender Specific Header 207
3.19.4.3 UNCERRSTS: Uncorrectable Error Status 207
3.19.4.4 UNCERRMSK: Uncorrectable Error Mask 208
3.19.4.5 UNCERRSEV: Uncorrectable Error Severity 209
3.19.4.6 CORERRSTS: Correctable Error Status 210

3.19.4.7 CORERRMSK: Correctable Error Mask 210
3.19.4.8 ERRCAP: Advanced Error Capabilities and Control Register 211
3.19.4.9 HDRLOG: Header Log 211
3.19.4.10 RPERRCMD: Root Port Error Command Register 212
3.19.4.11 RPERRSTS: Root Port Error Status Register 212
3.19.4.12 ERRSID: Error Source Identification Register 214
3.19.4.13 SSMSK: Stop and Scream Mask Register 214
3.19.4.14 APICBASE: APIC Base Register 215
3.19.4.15 APICLIMIT: APIC Limit Register 215
3.19.4.16 ACSCAPHDR: Access Control Services Extended Capability Header 215
3.19.4.17 ACSCAP: Access Control Services Capability Register 215
3.19.4.18 ACSCTRL: Access Control Services Control Register 216
3.19.4.19 PERFCTRLSTS: Performance Control and Status Register 216
3.19.4.20 MISCCTRLSTS: Misc. Control and Status Register 216
3.19.4.21 PCIE_IOU0_BIF_CTRL: PCIE IOU0 Bifurcation Control Register 217
3.19.4.22 NTBDEVCAP: PCI Express Device Capabilities Register 217
3.19.4.23 LNKCAP: PCI Express Link Capabilities Register 219
3.19.4.24 LNKCON: PCI Express Link Control Register 221
3.19.4.25 LNKSTS: PCI Express Link Status Register 223
3.19.4.26 SLTCAP: PCI Express Slot Capabilities Register 225
3.19.4.27 SLTCON: PCI Express Slot Control Register 227
3.19.4.28 SLTSTS: PCI Express Slot Status Register 229
3.19.4.29 ROOTCON: PCI Express Root Control Register 231
3.19.4.30 DEVCAP2: PCI Express Device Capabilities 2 Register 233
3.19.4.31 DEVCTRL2: PCI Express Device Control 2 Register 234
3.19.4.32 LNKCON2: PCI Express Link Control Register 2 235
3.19.4.33 LNKSTS2: PCI Express Link Status 2 Register 236
3.19.4.34 CTOCTRL: Completion Time-out Control Register 236
3.19.4.35 PCIE_LER_SS_CTRLSTS: PCI Express Live Error Recovery/Stop and Scream Control and Status Register 236
3.19.4.36 XPCORERRSTS - XP Correctable Error Status Register 236
3.19.4.37 XPCORERRMSK - XP Correctable Error Mask Register 236
3.19.4.38 XPUNCERRSTS - XP Uncorrectable Error Status Register 236
3.19.4.39 XPUNCERRMSK - XP Uncorrectable Error Mask Register 236
3.19.4.40 XPUNCERRSEV - XP Uncorrectable Error Severity Register 237
3.19.4.41 XPUNCERRPTR - XP Uncorrectable Error Pointer Register 237
3.19.4.42 UNCEDMASK: Uncorrectable Error Detect Status Mask 237
3.19.4.43 COREDMASK: Correctable Error Detect Status Mask 237
3.19.4.44 RPEDMASK - Root Port Error Detect Status Mask 237
3.19.4.45 XPUNCEDMASK - XP Uncorrectable Error Detect Mask Register 237
3.19.4.46 XPCOREDMASK - XP Correctable Error Detect Mask Register 237
3.19.4.47 XPGLBERRSTS - XP Global Error Status Register 237
3.19.4.48 XPGLBERRPTR - XP Global Error Pointer Register 237
3.20 PCI Express Configuration Registers (NTB Secondary Side) 238
3.20.1 Configuration Register Map (NTB Secondary Side) 238
3.20.2 Standard PCI Configuration Space (0x0 to 0x3F) - Type 0 Common Configuration Space 240
3.20.2.1 VID: Vendor Identification Register 240
3.20.2.2 DID: Device Identification Register (Dev#N, PCIE NTB Sec Mode) 240
3.20.2.3 PCICMD: PCI Command Register (Dev#N, PCIE NTB Sec Mode) .. 241
3.20.2.4 PCISTS: PCI Status Register 243
3.20.2.5 RID: Revision Identification Register 245
3.20.2.6 CCR: Class Code Register 245
3.20.2.7 CLSR: Cacheline Size Register 246
3.20.2.8 PLAT: Primary Latency Timer 246
3.20.2.9 HDR: Header Type Register (Dev#3, PCIe NTB Sec Mode) 246
3.20.2.10 BIST: Built-In Self Test 247
3.20.2.11 SB01BASE: Secondary BAR 0/1 Base Address (PCIE NTB Mode) .. 247
3.20.2.12 SB23BASE: Secondary BAR 2/3 Base Address (PCIE NTB Mode) .. 248

3.20.2.13 SB45BASE: Secondary BAR 4/5 Base Address 249
3.20.2.14 SUBVID: Subsystem Vendor ID (Dev#3, PCIE NTB Sec Mode) 250
3.20.2.15 SID: Subsystem Identity (Dev#3, PCIE NTB Sec Mode) 250
3.20.2.16 CAPPTR: Capability Pointer 250
3.20.2.17 INTL: Interrupt Line Register 251
3.20.2.18 INTPIN: Interrupt Pin Register 251
3.20.2.19 MINGNT: Minimum Grant Register 252
3.20.2.20 MAXLAT: Maximum Latency Register 252
3.20.3 Device-Specific PCI Configuration Space - 0x40 to 0xFF 252
3.20.3.1 MSICAPID: MSI Capability ID 252
3.20.3.2 MSINXTPTR: MSI Next Pointer 252
3.20.3.3 MSICTRL: MSI Control Register 253
3.20.3.4 MSIAR: MSI Lower Address Register 254
3.20.3.5 MSIUAR: MSI Upper Address Register 254
3.20.3.6 MSIDR: MSI Data Register 255
3.20.3.7 MSIMSK: MSI Mask Bit Register 256
3.20.3.8 MSIPENDING: MSI Pending Bit Register 256
3.20.3.9 MSIXCAPID: MSI-X Capability ID 257
3.20.3.10 MSIXNXTPTR: MSI-X Next Pointer 257
3.20.3.11 MSIXMSGCTRL: MSI-X Message Control Register 257
3.20.3.12 TABLEOFF_BIR: MSI-X Table Offset and BAR Indicator Register (BIR) 258
3.20.3.13 PBAOFF_BIR: MSI-X Pending Bit Array Offset and BAR Indicator 259
3.20.3.14 PXPCAPID: PCI Express Capability Identity Register 259
3.20.3.15 PXPNXTPTR: PCI Express Next Pointer Register 260
3.20.3.16 PXPCAP: PCI Express Capabilities Register 260
3.20.3.17 DEVCAP: PCI Express Device Capabilities Register 261
3.20.3.18 DEVCTRL: PCI Express Device Control Register (PCIE NTB Secondary) 263
3.20.3.19 DEVSTS: PCI Express Device Status Register 265
3.20.3.20 LNKCAP: PCI Express Link Capabilities Register 266
3.20.3.21 LNKCON: PCI Express Link Control Register 268
3.20.3.22 LNKSTS: PCI Express Link Status Register 270
3.20.3.23 DEVCAP2: PCI Express Device Capabilities Register 2 272
3.20.3.24 DEVCTRL2: PCI Express Device Control Register 2 272
3.20.3.25 SSCNTL: Secondary Side Control 274
3.20.3.26 PMCAP: Power Management Capabilities Register 274
3.20.3.27 PMCSR: Power Management Control and Status Register 275
3.20.3.28 SEXTCAPHDR: Secondary Extended Capability Header 276
3.21 NTB MMIO Space 277
3.21.1 NTB Shadowed MMIO Space 277
3.21.1.1 PBAR2LMT: Primary BAR 2/3 Limit 279
3.21.1.2 PBAR4LMT: Primary BAR 4/5 Limit 280
3.21.1.3 PBAR2XLAT: Primary BAR 2/3 Translate 281
3.21.1.4 PBAR4XLAT: Primary BAR 4/5 Translate 281
3.21.1.5 SBAR2LMT: Secondary BAR 2/3 Limit 282
3.21.1.6 SBAR4LMT: Secondary BAR 4/5 Limit 283
3.21.1.7 SBAR2XLAT: Secondary BAR 2/3 Translate 284
3.21.1.8 SBAR4XLAT: Secondary BAR 4/5 Translate 285
3.21.1.9 SBAR0BASE: Secondary BAR 0/1 Base Address 285
3.21.1.10 SBAR2BASE: Secondary BAR 2/3 Base Address 286
3.21.1.11 SBAR4BASE: Secondary BAR 4/5 Base Address 287
3.21.1.12 NTBCNTL: NTB Control 288
3.21.1.13 SBDF: Secondary Bus, Device and Function 290
3.21.1.14 CBDF: Captured Bus, Device and Function 290
3.21.1.15 PDOORBELL: Primary Doorbell 291
3.21.1.16 PDBMSK: Primary Doorbell Mask 292
3.21.1.17 SDOORBELL: Secondary Doorbell 292
3.21.1.18 SDBMSK: Secondary Doorbell Mask 292
3.21.1.19 USMEMMISS: Upstream Memory Miss 292

3.21.1.20 SPAD[0 - 15]: Scratchpad Registers 0 - 15 293
3.21.1.21 SPADSEMA4: Scratchpad Semaphore 294
3.21.1.22 RSDBMSIXV70: Route Secondary Doorbell MSI-X Vector 7 to 0 ... 295
3.21.1.23 RSDBMSIXV158: Route Secondary Doorbell MSI-X Vector 15 to 8 296
3.21.1.24 WCCNTRL: Write Cache Control Register 297
3.21.1.25 B2BSPAD[0 - 15]: Back-to-back Scratchpad Registers 0 - 15 297
3.21.1.26 B2BDOORBELL: Back-to-Back Doorbell 298
3.21.1.27 B2BBAR0XLAT: Back-to-Back BAR 0/1 Translate 299
3.21.2 MSI-X MMIO Registers (NTB Primary side) 300
3.21.2.1 PMSIXTBL[0-3]: Primary MSI-X Table Address Register 0 - 3 301
3.21.2.2 PMSIXDATA[0-3]: Primary MSI-X Message Data Register 0 - 3 301
3.21.2.3 PMSIXVECCNTL[0-3]: Primary MSI-X Vector Control Register 0 - 3 301
3.21.2.4 PMSIXPBA: Primary MSI-X Pending Bit Array Register 302
3.21.3 MSI-X MMIO registers (NTB Secondary Side) 303
3.21.3.1 SMSIXTBL[0-3]: Secondary MSI-X Table Address Register 0 - 3 .. 304
3.21.3.2 SMSIXDATA[0-3]: Secondary MSI-X Message Data Register 0 - 3 304
3.21.3.3 SMSIXVECCNTL[0-3]: Secondary MSI-X Vector Control Register 0 - 3 305
3.21.3.4 SMSIXPBA: Secondary MSI-X Pending Bit Array Register 305

4.0 Technologies 306
4.1 Intel® Virtualization Technology (Intel® VT) 306
4.1.1 Intel® VT-x Objectives 306
4.1.2 Intel® VT-x Features 307
4.1.3 Intel® VT-d Objectives 307
4.1.4 Intel® VT-d Features 308
4.1.5 Intel® VT-d Features Not Supported 308
4.2 Intel® I/O Acceleration Technology (Intel® IOAT) 308
4.2.1 Intel® QuickData Technology 309
4.2.1.1 Port/Stream Priority 309
4.2.1.2 Write Combining 309
4.2.1.3 Marker Skipping 309
4.2.1.4 Buffer Hint 309
4.2.1.5 DCA 309
4.2.1.6 DMA 309
4.3 Simultaneous Multi Threading (SMT) 311
4.4 Intel® Turbo Boost Technology 311

5.0 IIO Ordering Model 312
5.1 Introduction 312
5.2 Inbound Ordering Rules 313
5.2.1 Inbound Ordering Requirements 313
5.2.2 Special Ordering Relaxations 314
5.2.2.1 Inbound Writes Can Pass Outbound Completions 314
5.2.2.2 PCI Express Relaxed Ordering 314
5.2.3 Inbound Ordering Rules Summary 315
5.3 Outbound Ordering Rules 315
5.3.1 Outbound Ordering Requirements 315
5.3.2 Outbound Ordering Rules Summary 316
5.4 Peer-to-Peer Ordering Rules 317
5.4.1 Hinted Peer-to-Peer 317
5.4.2 Local Peer-to-Peer 317
5.4.3 Remote Peer-to-Peer 318
5.5 Interrupt Ordering Rules 318
5.5.1 SpcEOI Ordering 318
5.5.2 SpcINTA Ordering 318

5.6 Configuration Register Ordering Rules 319
5.7 Intel® VT-d Ordering Exceptions 319

6.0 System Address Map 320
6.1 Memory Address Space 321
6.1.1 System DRAM Memory Regions 322
6.1.2 VGA/SMM and Legacy C/D/E/F Regions 322
6.1.2.1 VGA/SMM Memory Space 323
6.1.2.2 C/D/E/F Segments 323
6.1.3 Address Region Between 1 MB and TOLM 324
6.1.3.1 Relocatable TSeg 324
6.1.4 PAM Memory Area Details 324
6.1.5 ISA Hole (15 MB –16 MB) 324
6.1.6 Memory Address Range TOLM – 4 GB 325
6.1.6.1 PCI Express Memory Mapped Configuration Space (PCI MMCFG) 325
6.1.6.2 MMIOL 325
6.1.6.3 I/OxAPIC Memory Space 325
6.1.6.4 HPET/Others 326
6.1.6.5 Local XAPIC 326
6.1.6.6 Firmware 326
6.1.7 Address Regions above 4 GB 327
6.1.7.1 High System Memory 327
6.1.7.2 Memory Mapped IO High 327
6.1.8 Protected System DRAM Regions 327
6.2 IO Address Space 328
6.2.1 VGA I/O Addresses 328
6.2.2 ISA Addresses 328
6.2.3 CFC/CF8 Addresses 328
6.2.4 PCIe Device I/O Addresses 328
6.3 IIO Address Map Notes 329
6.3.1 Memory Recovery 329
6.3.2 Non-Coherent Address Space 329
6.4 IIO Address Decoding 329
6.4.1 Outbound Address Decoding 329
6.4.1.1 General Overview 329
6.4.1.2 FWH Decoding 331
6.4.1.3 I/OxAPIC Decoding 331
6.4.1.4 Other Outbound Target Decoding 331
6.4.1.5 Summary of Outbound Target Decoder Entries 331
6.4.1.6 Summary of Outbound Memory/IO/Configuration Decoding 333
6.4.2 Inbound Address Decoding 335
6.4.2.1 Overview 335
6.4.2.2 Summary of Inbound Address Decoding 337
6.4.3 Intel® VT-d Address Map Implications 338

7.0 Interrupts 339
7.1 Overview 339
7.2 Legacy PCI Interrupt Handling 339
7.2.1 Integrated I/OxAPIC 340
7.2.1.1 Integrated I/OxAPIC EOI Flow 341
7.2.2 PCI Express INTx Message Ordering 341
7.2.3 INTR_Ack/INTR_Ack_Reply Messages 342
7.3 MSI 342
7.3.1 Interrupt Remapping 344
7.3.2 MSI Forwarding: IA32 Processor-based Platform 345
7.3.2.1 Legacy Logical Mode Interrupts 345
7.3.3 External IOxAPIC Support 346

7.4 Virtual Legacy Wires (VLW) 346
7.5 Platform Interrupts 347
7.6 Interrupt Flow 347
7.6.1 Legacy Interrupt Handled By IIO Module IOxAPIC 348
7.6.2 MSI Interrupt 348

8.0 Power Management 349
8.1 Introduction 349
8.1.1 ACPI States Supported 349
8.1.2 Supported System Power States 350
8.1.3 Processor Core/Package States 351
8.1.4 Integrated Memory Controller States 351
8.1.5 PCIe Link States 351
8.1.6 DMI States 352
8.1.7 Intel® QPI States 352
8.1.8 Intel® QuickData Technology State 352
8.1.9 Interface State Combinations 352
8.1.10 Supported DMI Power States 353
8.2 Processor Core Power Management 353
8.2.1 Enhanced Intel SpeedStep® Technology 353
8.2.2 Low-Power Idle States 354
8.2.3 Requesting Low-Power Idle States 355
8.2.4 Core C-States 356
8.2.4.1 Core C0 State 356
8.2.4.2 Core C1E State 356
8.2.4.3 Core C3 State 357
8.2.4.4 Core C6 State 357
8.2.4.5 C-State Auto-Demotion 357
8.2.5 Package C-States 357
8.2.5.1 Package C0 359
8.2.5.2 Package C1E 359
8.2.5.3 Package C3 State 359
8.2.5.4 Package C6 State 360
8.3 IMC Power Management 360
8.3.1 Disabling Unused System Memory Outputs 360
8.3.2 DRAM Power Management and Initialization 360
8.3.2.1 Initialization Role of CKE 360
8.3.2.2 Conditional Self-Refresh 360
8.3.2.3 Dynamic Power Down Operation 361
8.3.2.4 DRAM I/O Power Management 361
8.3.2.5 Asynch DRAM Self Refresh (ADR) 361
8.4 Device and Slot Power Limits 365
8.4.1 DMI Power Management Rules for the IIO Module 365
8.4.2 Support for P-States 365
8.4.3 S0 -> S1 Transition 365
8.4.4 S1 -> S0 Transition 366
8.4.5 S0 -> S3/S4/S5 Transition 366
8.5 PCIe Power Management 367
8.5.1 Power Management Messages 367
8.6 DMI Power Management 367
8.7 Intel® QPI Power Management 368
8.8 Intel® QuickData Technology Power Management 368
8.8.1 Power Management w/Assistance from OS-Level Software 368

9.0 Thermal Management 369

10.0 Reset 370

10.1 Introduction 370
10.1.1 Types of Reset 370
10.1.2 Trigger, Type, and Domain Association 370
10.2 Node ID Configuration 371
10.3 CPU-Only Reset 372
10.4 Reset Timing Diagrams 373
10.4.1 Cold Reset, CPU-Only Reset Timing Sequences 373
10.4.2 Miscellaneous Requirements and Limitations 373

11.0 Reliability, Availability, Serviceability (RAS) 375
11.1 IIO RAS Overview 375
11.2 System Level RAS 376
11.2.1 Inband System Management 376
11.2.2 Outband System Management 376
11.3 IIO Error Reporting 376
11.3.1 Error Severity Classification 377
11.3.1.1 Correctable Errors (Severity 0 Error) 377
11.3.1.2 Recoverable Errors (Severity 1 Error) 377
11.3.1.3 Fatal Errors (Severity 2 Error) 377
11.3.2 Inband Error Reporting 378
11.3.2.1 Synchronous Inband Error Reporting 378
11.3.2.2 Asynchronous Error Reporting 379
11.3.3 IIO Error Registers Overview 381
11.3.3.1 Local Error Registers 382
11.3.3.2 Global Error Registers 383
11.3.3.3 First and Next Error Log Registers 388
11.3.3.4 Error Logging Summary 388
11.3.3.5 Error Registers Flow 389
11.3.3.6 Error Containment 390
11.3.3.7 Error Counters 391
11.3.3.8 Stop on Error 391
11.4 IIO Intel® QuickPath Interconnect Interface RAS 391
11.4.1 Intel® QuickPath Interconnect Error Detection, Logging, and Reporting 392
11.5 PCI Express* RAS 392
11.5.1 PCI Express* Link CRC and Retry 392
11.5.2 Link Retraining and Recovery 392
11.5.3 PCI Express Error Reporting Mechanism 392
11.5.3.1 PCI Express Error Severity Mapping in IIO 392
11.5.3.2 Unsupported Transactions and Unexpected Completions 393
11.5.3.3 Error Forwarding 393
11.5.3.4 Unconnected Ports 393
11.6 IIO Errors Handling Summary 393
11.7 Hot Add/Remove Support 408
11.7.1 Hot Add/Remove Rules 409
11.7.2 PCIe Hot Plug 409
11.7.2.1 PCI Express Hot Plug Interface 410
11.7.2.2 PCI Express Hot Plug Interrupts 411
11.7.2.3 Virtual Pin Ports (VPP) 413
11.7.2.4 Operation 414
11.7.2.5 Miscellaneous Notes 416
11.7.3 Intel® QPI Hot Plug 417

12.0 Packaging and Signal Information 418
12.1 Signal Descriptions 418
12.1.1 Intel® QPI Signals 418
12.1.2 System Memory Interface 419
12.1.2.1 DDR Channel A Signals 419

12.1.2.2 DDR Channel B Signals 420
12.1.2.3 DDR Channel C Signals 421
12.1.2.4 System Memory Compensation Signals 421
12.1.3 PCI Express* Signals 422
12.1.4 Processor SMBus Signals 422
12.1.5 DMI / ESI Signals 423
12.1.6 Clock Signals 423
12.1.7 Reset and Miscellaneous Signals 424
12.1.8 Thermal Signals 424
12.1.9 Processor Core Power Signals 425
12.1.10 Power Sequencing Signals 426
12.1.11 No Connect and Reserved Signals 426
12.1.12 ITP Signals 427
12.2 Physical Layout and Signals 427

13.0 Electrical Specifications 483
13.1 Processor Signaling 483
13.1.1 Intel® QuickPath Interconnect 483
13.1.2 DDR3 Signal Groups 483
13.1.3 Platform Environmental Control Interface (PECI) 484
13.1.3.1 Input Device Hysteresis 484
13.1.4 PCI Express/DMI 484
13.1.5 SMBus Interface 485
13.1.6 Clock Signals 486
13.1.7 Reset and Miscellaneous 486
13.1.8 Thermal 486
13.1.9 Test Access Port (TAP) Signals 486
13.1.10 Power / Other Signals 486
13.1.10.1 Power and Ground Lands 487
13.1.10.2 Decoupling Guidelines 487
13.1.10.3 Processor VCC Voltage Identification (VID) Signals 487
13.1.10.4 Processor VTT Voltage Identification (VTT_VID) Signals 494
13.1.11 Reserved or Unused Signals 495
13.2 Signal Group Summary 495
13.3 Mixing Processors 500
13.4 Flexible Motherboard Guidelines (FMB) 500
13.5 Absolute Maximum and Minimum Ratings 500
13.6 Processor DC Specifications 501
13.6.1 VCC Overshoot Specifications 507
13.6.2 Die Voltage Validation 508
13.6.3 DDR3 Signal DC Specifications 508
13.6.4 PCI Express Signal DC Specifications 510
13.6.5 SMBus Signal DC Specifications 511
13.6.6 PECI Signal DC Specifications 512
13.6.7 System Reference Clock Signal DC Specifications 512
13.6.8 Reset and Micscellaneous DC Specifications 513
13.6.9 Thermal DC Specification 513
13.6.10 Test Access Port (TAP) DC Specification 514
13.6.11 Power Sequencing Signal DC Specification 514

14.0 Testability 515
14.1 Boundary-Scan 515
14.2 TAP Controller Operation and State Diagram 515
14.3 TAP Instructions and Opcodes 517
14.3.1 Processor Core TAP Controller 517
14.3.2 Processor Un-Core TAP Controller 517

14.3.3 Processor Integrated I/O TAP Controller 517
14.3.4 TAP Interface 518
14.4 TAP Port Timings 520
14.5 Boundary-Scan Register Definition 520

Figures

1 Intel® Xeon® Processor C5500/C3500 Series on the Picket Post Platform -- UP Configuration 25
2 Intel® Xeon® Processor C5500/C3500 Series on the Picket Post Platform -- DP Configuration 26
3 Independent Code Layout 40
4 Lockstep Code Layout 42
5 Dual-Channel Symmetric (Interleaved) and Dual-Channel Asymmetric Modes 44
6 Intel® Flex Memory Technology Operation 44
7 DIMM Population Within a Channel 46
8 DIMM Population Within a Channel for Two Slots per Channel 47
9 Error Signaling Logic 50
10 First Level Address Decode Flow 52
11 Mapping Throttlers to Ranks 62
12 Ping() 71
13 Ping() Example 71
14 GetDIB() 72
15 Device Info Field Definition 72
16 Revision Number Definition 73
17 GetTemp() 74
18 GetTemp() Example 74
19 PCI Configuration Address 75
20 PCIConfigRd() 75
21 PCIConfigWr() 77
22 Thermal Status Word 79
23 Thermal Data Configuration Register 80
24 Machine Check Read MbxSend() Data Format 80
25 ACPI T-State Throttling Control Read / Write Definition 82
26 MbxSend() Command Data Format 83
27 MbxSend() 83
28 MbxGet() 85
29 Temperature Sensor Data Format 89
30 PECI Power-up Timeline 90
31 SMBus Block-Size Configuration Register Read 99
32 SMBus Block-size Memory Register Read 99
33 SMBus Word-Size Configuration Register Read 100
34 SMBus Word-Size Memory Register Read 100
35 SMBus Byte-Size Configuration Register Read 101
36 SMBus Byte-Size Memory Register Read 102
37 SMBus Block-Size Configuration Register Write 103
38 SMBus Block-Size Memory Register Write 103
39 SMBus Word-Size Configuration Register Write 104
40 SMBus Word-Size Memory Register Write 104
41 SMBus Configuration (Byte Write, PEC enabled) 104
42 SMBus Memory (Byte Write, PEC enabled) 105
43 Intel® Xeon® Processor C5500/C3500 Series Dual Processor Configuration Block Diagram 106
44 PCI Express Layering Diagram 119
45 Packet Flow through the Layers 120
46 Enumeration in System with Transparent Bridges and Endpoint Devices 137
47 Non-Transparent Bridge Based Systems 138
48 NTB Ports Connected Back-to-Back 139
49 NTB Port on Intel® Xeon® Processor C5500/C3500 Series Connected to Root Port - Symmetric Configuration 140

50 NTB Port on Intel® Xeon® Processor C5500/C3500 Series Connected to Root Port - Non-Symmetric 141
51 NTB Port Connected to Non-Intel® Xeon® Processor C5500/C3500 Series System - Non-Symmetric 142
52 Intel® Xeon® Processor C5500/C3500 Series NTB Port - Nomenclature 144
53 Crosslink Configuration 147
54 B2B BAR and Translate Setup 149
55 Intel® Xeon® Processor C5500/C3500 Series NTB Port - BARs 152
56 Direct Address Translation 153
57 NTB to NTB Read Request, ID translation Example 155
58 NTB to RP Read Request, ID translation Example 156
59 RP to NTB Read Request, ID translation Example 157
60 B2B Doorbell 168
61 PCI Express NTB (Device 3) Type0 Configuration Space 171
62 PCI Express NTB Secondary Side Type0 Configuration Space 238
63 System Address Map 321
64 VGA/SMM and Legacy C/D/E/F Regions 322
65 Intel® Xeon® Processor C5500/C3500 Series Only: Peer-to-Peer Illustration 336
66 Interrupt Transformation Table Entry (IRTE) 345
67 ACPI Power States in G0, G1, and G2 States 350
68 Idle Power Management Breakdown of the Processor Cores (Two-Core Example) 354
69 Thread and Core C-State Entry and Exit 355
70 Package C-State Entry and Exit 359
71 DDR_ADR to Self-Refresh Entry 364
72 Intel® Xeon® Processor C5500/C3500 Series System Diagram 373
73 IIO Error Registers 382
74 IIO Core Local Error Status, Control and Severity Registers 383
75 IIO Global Error Control/Status Register 384
76 IIO System Event Register 385
77 IIO Error Logging and Reporting Example 386
78 Error Logging and Reporting Example 387
79 IIO Error Logging Flow 389
80 IIO PCI Express Hog Plug Serial Interface 410
81 MSI Generation Logic at each PCI Express Port for PCI Express Hot Plug 412
82 GPE Message Generation Logic at each PCI Express Port for PCI Express Hot Plug 413
83 Active ODT for a Differential Link Example 483
84 Input Device Hysteresis 484
85 MSID Timing Requirement 494
86 VCC Static and Transient Tolerance Loadlines1,2,3,4 506
87 VCC Overshoot Example Waveform 507
88 TAP Controller State Diagram 516
89 Processor TAP Controller Connectivity 518
90 Processor TAP Connections 519
91 Boundary-Scan Port Timing Waveforms 520

Tables

1 Available SKUs 27
2 Terminology 32
3 Processor Documents 33
4 PCH Documents 34
5 Public Specifications 34
6 System Memory Feature Summary 35
7 Intel® Xeon® Processor C5500/C3500 Series with RDIMM Only Support 37
8 UDIMM Only Support 37
9 DDR3 System Memory Timing Support 38
10 Mapping from Logical to Physical Channels 39
11 RDIMM Population Configurations Within a Channel for Three Slots per Channel 46
12 UDIMM Population Configurations Within a Channel for Three Slots per Channel 47
13 DIMM Population Configurations Within a Channel for Two Slots per Channel 47
14 UDIMM Population Configurations Within a Channel for Two Slots per Channel 48
15 Causes of SMI or NMI 51
16 Read and Write Steering 53
17 Address Mapping Registers 54
18 Critical Word First Sequence of Read Returns 57
19 Lower System Address Bit Mapping Summary 57
20 DDR Organizations Supported 58
21 DRAM Power Savings Exit Parameters 60
22 Dynamic IO Power Savings Features 60
23 DDR_THERM# Responses 64
24 Refresh for Different DRAM Types 65
25 1 or 2 Single/Dual Rank Throttling 67
26 1 or 2 Quad Rank or 3 Single/Dual Rank Throttling 67
27 Thermal Throttling Control Fields 68
28 Thermal Throttling Status Fields 69
29 Summary of Processor-Specific PECI Commands 70
30 GetTemp() Response Definition 74
31 PCIConfigRd() Response Definition 76
32 PCIConfigWr() Device/Function Support 76
33 PCIConfigWr() Response Definition 77
34 Mailbox Command Summary 78
35 Counter Definition 79
36 Machine Check Bank Definitions 81
37 ACPI T-State Duty Cycle Definition 82
38 MbxSend() Response Definition 84
39 MbxGet() Response Definition 85
40 Domain ID Definition 87
41 Multi-Domain Command Code Reference 87
42 Completion Code Pass/Fail Mask 87
43 Device Specific Completion Code (CC) Definition 88
44 Originator Response Guidelines 88
45 Error Codes and Descriptions 90
46 PECI Client Response During Power-Up (During 'Data Not Ready') 90
47 Power Impact of PECI Commands vs. C-states 91
48 PECI Client Response During S1 92
49 SMBus Command Encoding 94
50 Internal SMBus Protocol Stack 95
51 SMBus Slave Address Format 95
52 Memory Region Address Field 96
53 Status Field Encoding for SMBus Reads 97

54 Processor's Intel® QuickPath Interconnect Physical Layer Attributes 107
55 Intel® QuickPath Interconnect Link Layer Attributes 108
56 Intel® QuickPath Interconnect Routing Layer Attributes 108
57 Processor's Intel® QuickPath Interconnect Coherent Protocol Attributes 110
58 Picket Post Platform Intel® QuickPath Interconnect Non-Coherent Protocol Attributes 110
59 Intel® QuickPath Interconnect Interrupts Attributes 110
60 Intel® QuickPath Interconnect Fault Handling Attributes 111
61 Intel® QuickPath Interconnect Reset/Initialization Attributes 111
62 Intel® QuickPath Interconnect Other Attributes 111
63 Supported Intel® QPI Message Classes 112
64 Memory Address Decoder Fields 114
65 I/O Decoder Entries 115
66 Profile Control 117
67 Time-Out Level Classification for IIO 118
68 Link Width Strapping Options 122
69 Supported Degraded Modes in IIO 122
70 Incoming PCI Express Message Cycles 125
71 Outgoing PCI Express Memory, I/O and Configuration Request/Completion Cycles 126
72 Outgoing PCI Express Message Cycles 127
73 PCI Express Transaction ID Handling 128
74 PCI Express Attribute Handling 129
75 PCI Express CompleterID Handling 129
76 PCI Express Credit Mapping for Inbound Requests 132
77 PCI Express Credit Mapping for Outbound Requests 132
78 Type 0 Configuration Header for Local and Remote Interface 144
79 Class Code 145
80 Memory Aperture Size Defined by BAR 146
81 Incoming PCI Express NTB Memory, I/O and Configuration Request/Completion Cycles 158
82 Incoming PCI Express Message Cycles 159
83 Outgoing PCI Express Memory, I/O and Configuration Request/Completion Cycles 160
84 Outgoing PCI Express Message Cycles with Respect to NTB 162
85 PCI Express Transaction ID Handling 164
86 PCI Express Attribute Handling 164
87 PCI Express CompleterID Handling 165
88 IIO Bus 0 Device 3 Legacy Configuration Map (PCI Express Registers) 172
89 IIO Devices 3 Extended Configuration Map (PCI Express Registers) Page#0 173
90 IIO Devices 3 Extended Configuration Map (PCI Express Registers) Page#1 174
91 MSI Vector Handling and Processing by IIO on Primary Side 190
92 MSI Vector Handling and Processing by IIO on Secondary Side 256
93 NTB MMIO Shadow Registers 277
94 NTB MMIO Map 278
95 NTB MMIO Map 300
96 MSI-X Vector Handling and Processing by IIO on Primary Side 301
97 NTB MMIO Map 303
98 MSI-X Vector Handling and Processing by IIO on Secondary Side 304
99 Ordering Term Definitions 312
100 Inbound Data Flow Ordering Rules 315
101 Outbound Data Flow Ordering Rules 317
102 Outbound Target Decoder Entries 332
103 Decoding of Outbound Memory Requests from Intel® QPI (from CPU or Remote Peer-to-Peer) 333
104 Decoding of Outbound Configuration Requests from Intel® QPI and Decoding of Outbound Peer-to-Peer Completions from Intel® QPI 334
105 Subtractive Decoding of Outbound I/O Requests from Intel® QPI 334

106 Inbound Memory Address Decoding 337
107 Interrupt Source in IOxAPIC Table Mapping 340
108 I/OxAPIC Table Mapping to PCI Express Interrupts 340
109 MSI Address Format when Remapping Disabled 342
110 MSI Data Format when Remapping Disabled 343
111 MSI Address Format when Remapping is Enabled 343
112 MSI Data Format when Remapping is Enabled 344
113 Platform System States 350
114 Integrated Memory Controller States 351
115 PCIe Link States 351
116 DMI States 352
117 Intel® QPI States 352
118 Intel® QuickData Technology States 352
119 G, S, and C State Combinations 352
120 System and DMI Link Power States 353
121 Coordination of Thread Power States at the Core Level 355
122 P_LVLx to MWAIT Conversion 356
123 Coordination of Core Power States at the Package Level 358
124 Targeted Memory State Conditions 361
125 ADR Self-Refresh Entry Timing - AC Characteristics (CMOS 1.5 V) 365
126 Core Trigger, Type, Domain Association 371
127 IIO Intel® QPI RAS Feature Support 391
128 IIO Default Error Severity Map 394
129 IIO Error Summary 394
130 Hot Plug Interface 410
131 I/O Port Registers in On-Board SMBus devices Supported by IIO 414
132 Hot Plug Signals on a Virtual Pin Port 414
133 Write Command 415
134 Read Command 416
135 Intel® QPI Signals 418
136 DDR Channel A Signals 419
137 DDR Channel B Signals 420
138 DDR Channel C Signals 421
139 DDR Miscellaneous Signals 421
140 PCI Express Signals 422
141 Processor SMBus Signals 422
142 DMI / ESI Signals 423
143 PLL Signals 423
144 Miscellaneous Signals 424
145 Thermal Signals 424
146 Power Signals 425
147 Reset Signals 426
148 No Connect Signals 426
149 ITP Signals 427
150 Physical Layout, Left Side 428
151 Physical Layout, Center 431
152 Physical Layout, Right 434
153 Alphabetical Listing by X and Y Coordinate 437
154 Alphabetical Signal Listing 449
155 Processor Power Supply Voltages1 487
156 Voltage Identification Definition 488
157 Power-On Configuration (POC[7:0]) Decode 493
158 VTT Voltage Identification Definition 495
159 Signal Groups 495
160 Signals With On-Die Termination (ODT) 499

161 Processor Absolute Minimum and Maximum Ratings 501
162 Voltage and Current Specifications 502
163 VCC Static and Transient Tolerance 505
164 VCC Overshoot Specifications 507
165 ICC Max and ICC TDP by SKU 508
166 DDR3 Signal Group DC Specifications 508
167 PCI Express/DMI Interface -- 2.5 and 5.0 GT/s Transmitter DC Specifications 510
168 PCI Express Interface -- 2.5 and 5.0 GT/s Recevier DC Specifications 511
169 SMBus Clock DC Electrical Limits 511
170 PECI DC Electrical Limits 512
171 System Reference Clock DC Specifications 512
172 Reset and Miscellaneous Signal Group DC Specifications 513
173 Thermal Signal Group DC Specification 513
174 Test Access Port (TAP) Signal Group DC Specification 514
175 Power Sequencing Signal Group DC Specifications 514
176 Processor Core TAP Controller Supported Boundary-Scan Instruction Opcodes 517
177 Processor Un-Core TAP Controller Supported Boundary-Scan Instruction Opcodes 517
178 Processor Integrated I/O TAP Controller Supported Boundary-Scan Instruction Opcodes .. 518
179 Processor Boundary-Scan TAP Pin Interface 519
180 Boundary-Scan Signal Timings 520

Revision History

Date	Revision	Description
February 2010	001	First release

1.0 Features Summary

1.1 Introduction

This Datasheet describes DC and AC electrical specifications, signal integrity, differential signaling specifications, pinout and signal definitions, interface functional descriptions, and additional feature information pertinent to the implementation and operation of the Intel® Xeon® processor C5500/C3500 series on its respective platform.

The Intel® Xeon® processor C5500/C3500 series is the next generation of multi-core embedded/server family of processors built on 45-nanometer process technology. Included in this family of processors is an integrated memory controller (IMC) and integrated I/O (IIO). The IIO provides PCI Express*, DMI, SMBus, Intel® QuickData Technology (DMA architecture), Intel® VTd-2 for server security, etc. All of this is integratated on a single silicon die.

Based on low-power/high-performance Intel® Core™ processor, the Intel® Xeon® processor C5500/C3500 series allows for a two-chip uni-processor (UP) platform as opposed to the traditional three-chip platforms (processor, MCH, and ICH). The two-chip platform consists of a processor and the Platform Controller Hub (PCH). This two-chip platform enables higher performance, lower cost, easier validation and an improved x-y footprint.

In addition, a dual-processor (DP) configuration is supported for more performance-demanding applications. This configuration adds an additional Intel® Xeon® processor C5500/C3500 series. The processor and the chipset (PCH) comprise the Picket Post UP and DP platforms illustrated respectively in Figure 1 on page 25 and Figure 2 on page 26.

Throughout this document, the Intel® Xeon® processor C5500/C3500 series might be referred to as the “processor”.

Figure 1. Intel® Xeon® Processor C5500/C3500 Series on the Picket Post Platform -- UP Configuration

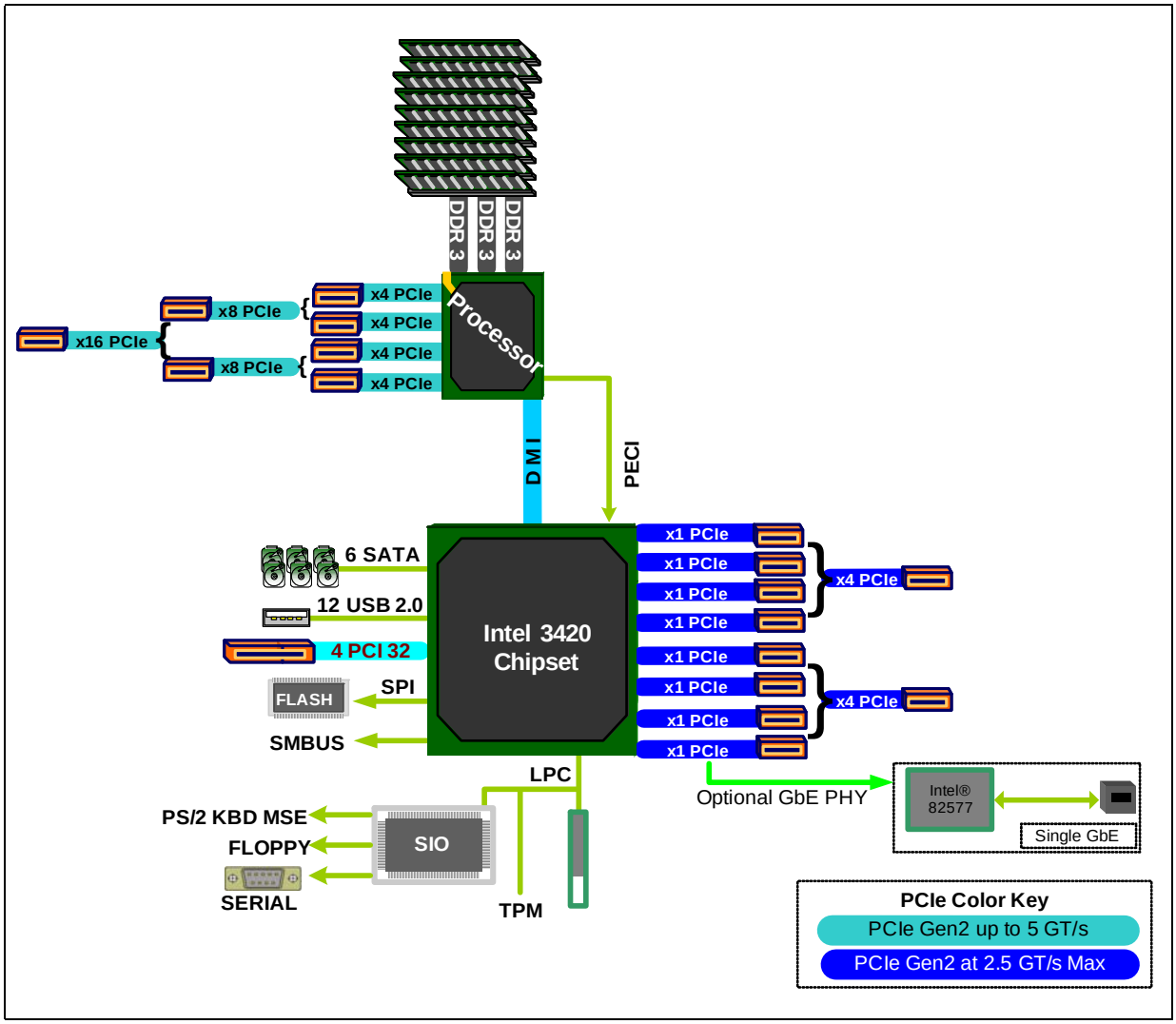
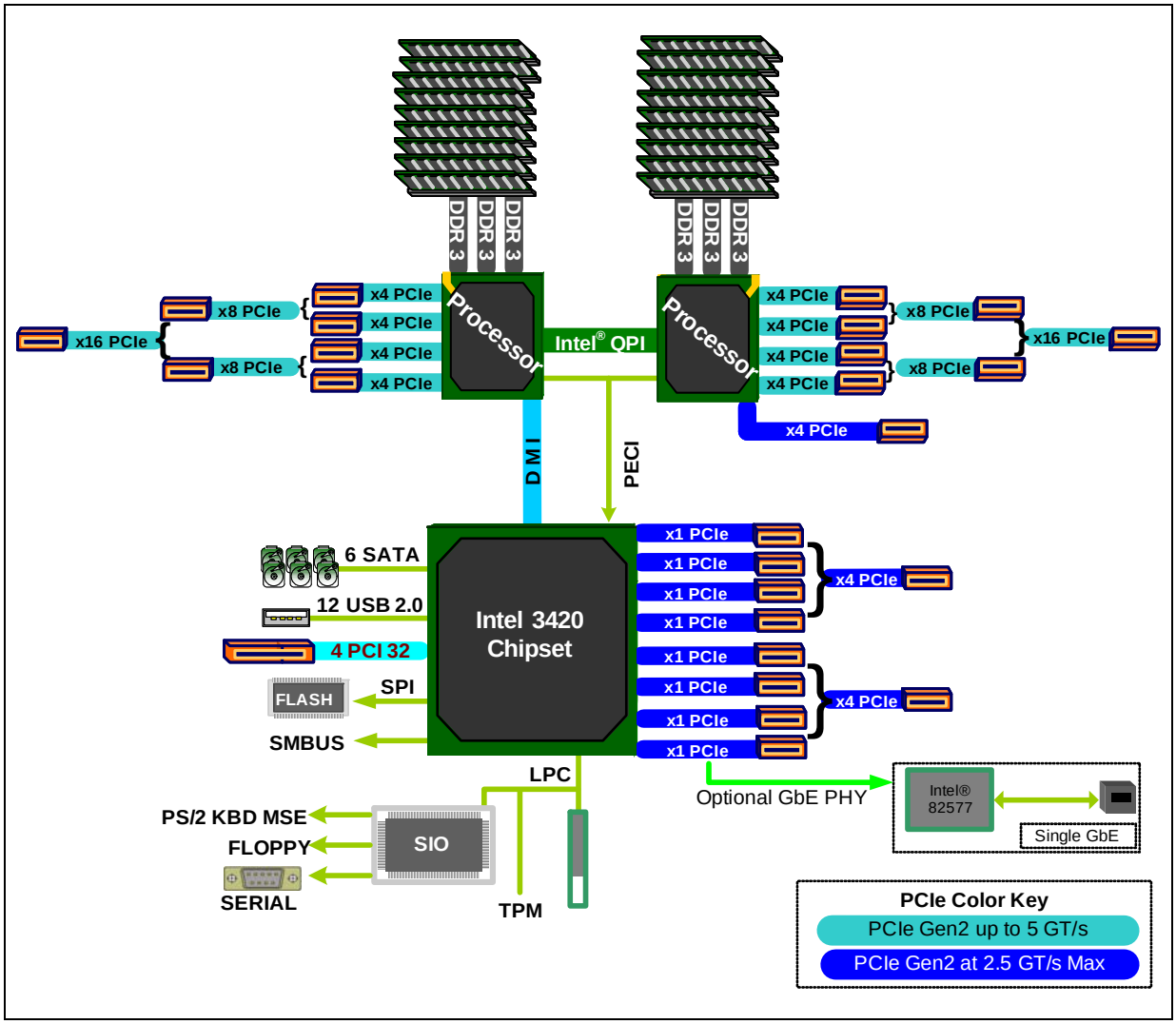


Figure 2. Intel® Xeon® Processor C5500/C3500 Series on the Picket Post Platform -- DP Configuration



1.2 Processor Feature Details

- SKUs supporting one, two, and four cores
- Separate 32 kB instruction and 32 kB data L1 cache
  - — L1 data and instruction cache are implemented as two redundent caches each of which is parity protected
- A 256-KB shared instruction/data L2 cache with ECC for each core
- Up to 8 MB shared instruction/data L3 cache with ECC, shared among all cores
- SKUs at different power and performance levels supporting UP (uni-processor) and DP (dual-processor)

1.2.1 Supported Technologies

- Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d2)
- Intel® QuickData Technology
- Intel® Streaming SIMD Extensions 4.1 (Intel® SSE4.1)
- Intel® Streaming SIMD Extensions 4.2 (Intel® SSE4.2)
- Simultaneous Multi-Threading (SMT)
- Intel® 64 Architecture
- Execute Disable Bit

1.3 SKUs

The Intel® Xeon® processor C5500/C3500 series comes in multiple SKUs thereby allowing it to support a broad range of performance levels, capabilities, and features. The SKUs and their attributes are summarized in the following table.

Table 1. Available SKUs

Processor Number[1]	DP Capable	TDP (W)	Base Clock Speed (GHz)	Turbo Freq	Intel® Hyper-Threading Tech	LLC Cache (MB)	Cores/ Threads	Thermal Profile (High TCase)	Intel® QuickPath Link Speed	DDR3 Memory	Memory Channels
ECC5549	Yes	85	2.53	Up to 2.93 GHz	Yes	8	4/8	Standard	5.86 GT/s	1333/ 1066/ 800	3
ECC5509	Yes	85	2.00	No	No	8	4	Standard	4.8 GT/s	1066/ 800	3
ECC3539	No	65	2.13	No	No	8	4	Standard	NA	1066/ 800	3
LC5528	Yes	60	2.13	Up to 2.53 GHz	Yes	8	4/8	70° C (nominal) 85° C (short)	4.8 GT/s	1066/ 800	3
EC5539	Yes	65	2.27	No	No	4	2	Standard	5.86 GT/s	1333/ 1066/ 800	3
LC5518	Yes	48	1.73	Up to 2.13 GHz	Yes	8	4/8	77.5° C (nominal) 92.5° C (short)	4.8 GT/s	1066/ 800	3

Table 1. Available SKUs

Processor Number[1]	DP Capable	TDP (W)	Base Clock Speed (GHz)	Turbo Freq	Intel® Hyper-Threading Tech	LLC Cache (MB)	Cores/ Threads	Thermal Profile (High TCase)	Intel® QuickPath Link Speed	DDR3 Memory	Memory Channels
LC3528	No	35	1.73	Up to 2.13 GHz	Yes	4	2/4	79.6° C (nominal) 94.6° C (short)	NA	1066/ 800	2
LC3518	No	23	1.73	No	No	2	1	79.5° C (nominal) 94.5° C (short)	NA	800	2
P1053	No	30	1.33	No	Yes	2	1/2	Standard	NA	800	2

Note:

1. All processors are Intel® Xeon® processors except for processor number P1053. Processor number P1053 is the Intel® Celeron® processor. The P1053 does not support memory RAS features, i.e., scrubbing (demand and patrol), mirroring, lockset, and SDDC. The sections of the Datasheet describing these features DO NOT apply to he P1053.

1.4 Interfaces

1.4.1 Intel® QuickPath Interconnect (Intel® QPI)

The Intel® QuickPath Interconnect (Intel® QPI) interconnets two processors for SKUs that support DP configurations. Intel® QPI is a cache-coherent links-based interconnect, which is an Intel proprietary specification for links-based processor and chipset components.

Intel® QPI on the Intel® Xeon® processor C5500/C3500 series supports the following features:

- 64-byte cache lines
- SKU dependent Link transfer rates: 4.8 and 5.87 GT/s
- L0, L0S, and L1 power states
- 40-bit Physical Addressing
- Intel® QPI Route-Through to allow DP systems to seamlessly access each other's resources.

1.4.2 System Memory Support

- SKUs supporting two or three channels of DDR3 memory:
  - — Registered, ECC, up to three DIMMs per channel (three DIMMS only supported at 800 MT/s), X4 or X8 with 1-Gb, 2-Gb, or 4-Gb DRAM technology.
  - — Unbuffered, ECC or not, up to two DIMMs per channel, X8 or X16 with 512 Mb or 1-Gb or 2-Gb or 4-Gb DRAM technology.
  - — Max memory supported 192 GB (with 2-Gb DDR3 devices).
- Data burst length of 4 for lockstep mode and 8 for other memory operation modes.
- Memory DDR3 data transfer rates of 800, 1066, and 1333 MT/s
- 64-bit wide channels
- DDR3 I/O Voltage of 1.5 V
- Memory operating modes:

  - — Single Channel Mode
  - — Independent Channel Mode
  - — Spare Channel Mode
  - — Mirrored Mode
  - — Lockstep Mode
  - — Dual-channel: Modes; Symmetric (Interleaved); Asymmetric
  - — Intel® Flex Memory Technology
- Command launch modes of 1n/2n
- Various RAS modes
- On-Die Termination (ODT)
- Intel® Fast Memory Access (Intel® FMA):
  - — Just-in-Time Command Scheduling
  - — Command Overlap
  - — Out-of-Order Scheduling
- Asynchronous DRAM Refresh (ADR)

1.4.3 PCI Express

- One 16-lane PCI Express port that is fully compliant with the *PCI Express Base Specification,* Revision 2.0. The 16 lanes can be bifurcated into two x8 ports, one x8 port and two x4 ports, or four x4 ports.
- Support is provided for one port (X4 or X8) Non-Transparent Bridge (NTB). When the NTB mode is enabled, the remainder of the x16 lanes can only be configured as ordinary PCIe root ports.
- Negotiating down to narrower widths is supported: A x16 port may negotiate down to x8, x4, x2, or x1. A x8 port may negotiate down to x4, x2, or x1. A x4 port may negotiate down to x2, or x1. Restrictions as to how lane reversal is supported exist when negotiating down to narrower widths. See Table 1.4.3, “PCI Express” on page 29 for details.
- Support for Degraded Mode Operation.
- Support for both PCIe Gen1 & Gen2 frequencies.
- Automatic discovery, negotiation, and training of link out of reset.
- Support peer-to-peer memory reads and memory writes between PCIe links on the processor, or processors, in DP systems.

Note: Peer-to-peer traffic is not supported between PCIe links on the processor and PCIe links on the PCH.

- 64-bit downstream host address format, however since the processor’s addressiblity is limited to 40 bits (1 TB), bits 63:40 will always be set to zeros.
- 64-bit upstream host address format, however since the processor’s addressibility is limited to 40 bits (1 TB) it responds to upstream read transactions with an Unsupported Request response for addresses above 1 TB. Upstream write transactions to host addresses beyond 1 TB will be dropped.
- PCI Express reference clock is 100-MHz differential clock buffered out of system clock generator.
- Power Management Event (PME) functions.
- Static lane numbering reversal:

  - Does not support dynamic lane reversal.
- Supports Half Swing “low-power/low-voltage” mode.
- Message Signaled Interrupt (MSI and MSI-X) messages.
- Polarity inversion.

1.4.4 Direct Media Interface (DMI)

- Compliant to Direct Media Interface Second Generation (DMI2).
- Four lanes in each direction.
- 2.5 GT/s point-to-point DMI2 interface to PCH is supported.
- Uses the 100-MHz PCI Express reference clock (supplied through PCH).
- 64-bit downstream host address format. However, since the processor’s addressiblity is limited to 40 bits (1 TB), bits 63:40 will always be set to zeros.
- 64-bit upstream host address format. However, since the processor’s addressibility is limited to 40 bits (1 TB) it responds to upstream read transactions with an Unsupported Request response for addresses above 1 TB. Upstream write transactions to host addresses beyond 1 TB will be dropped.
- APIC and MSI interrupt messaging support:
  - Message Signaled Interrupt (MSI and MSI-X) messages.
- Virtual Legacy Wire (VLW) Messasage Support allows commuicating status of A20M#, INTR, SM#, INIT#, and NMI as messages, thereby eliminating the need for these sideband signals.
- Downstream SMI, SCI and SERR error indication.
- Legacy support for ISA regime protocol (PHOLD/PHOLDA) required for parallel port DMA, floppy drive, and LPC bus masters.
- Support for both AC (capacitors between the processor and PCH) & DC (no capacitors between the processor and PCH) coupling.
- Polarity inversion.
- PCH end-to-end lane reversal across the link.
- Supports Half Swing “low-power/low-voltage” and Full Swing “high-power/high-voltage” modes.
- In DP configurations, the unused DMI port can be configured as a Gen1, x4 or x1, non-bifurcatable, PCI Express port.

1.4.5 Platform Environment Control Interface (PECI)

The PECI is a one-wire interface that provides a communication channel between processor and a PECI master, usually the PCH.

1.4.6 SMBus

The Intel® Xeon® processor C5500/C3500 series supports a 2-pin SMBus slave for accessing the on-die system management registers. There is also a 2-pin SMBus master to support PCI Express hot plug.

1.5 Power Management Support

1.5.1 Processor Core

- Full support of ACPI C-states as implemented by the following processor core & package C-states:
  - — Core: C0, C1E, C3, C6
  - — Package: C0, C3, C6
- Enhanced Intel SpeedStep® Technology

1.5.2 System

- S0, S1, S3, S4, S5

1.5.3 Memory Controller

- Conditional self-refresh (Intel® Rapid Memory Power Management (Intel® RMPM)
- Dynamic power-down
- Asynchronous DRAM Refresh

1.5.4 PCI Express

- L0, L0s, L1, L3

1.5.5 DMI

- L0, L0s, L1, L3

1.5.6 Intel® QuickPath Interconnect

- L0, L0s, and L1

1.6 Thermal Management Support

PECI (Platform Environment Control Interface) is a serial processor interface used primarily for thermal power and error management. The PECI data may be read by the PCH or by a BMC or other external logic.

The Intel® Xeon® processor C5500/C3500 series contains six digital thermal sensors – one for each core, one for uncore, and one for the IIO portion of the die. The time average temperature, of the thermal sensor indicating the highest temperature, is reported via the PECI bus. This reflects the maximum die temperature. These five digital thermal sensors are used to initiate Adaptive Intel® Thermal Monitor.

1.7 Package

The Intel® Xeon® processor C5500/C3500 series socket type is noted as Socket B. The package is a 42.5X45.0mm Flip Chip Land Grid Array (LGA/FCLGA1366), with a 40-mil land pitch.

1.8 Terminology

Table 2. Terminology (Sheet 1 of 2)

Term	Description
BLT	Block Level Transfer
CRC	Cyclic Redundency Code
DCA	Direct Cache Access
DDR3	Third generation Double Data Rate SDRAM memory technology
DMA	Direct Memory Access
DMI	Direct Media Interface
DP	Dual processor
DTS	Digital Thermal Sensor
ECC	Error Correction Code
Enhanced Intel SpeedStep® Technology	Technology that provides power management capabilities
Execute Disable Bit	The Execute Disable bit allows memory to be marked as executable or non-executable, when combined with a supporting operating system. If code attempts to run in non-executable memory the processor raises an error to the operating system. This feature can prevent some classes of viruses or worms that exploit buffer overrun vulnerabilities and can thus help improve the overall security of the system. See the *Intel® 64 and IA-32 Architectures Software Developer's Manuals* for more detailed information.
EU	Execution Unit
FCLGA	Flip Chip Land Grid Array
Flit	
(G)MCH	Legacy component - Graphics Memory Controller Hub
ICH	The legacy I/O Controller Hub component that contains the main PCI interface, LPC interface, USB2, Serial ATA, and other I/O functions. It communicates with the legacy (G)MCH over a proprietary interconnect called DMI.
IIO	Integrated Input/Output (IOH module integrated into the processor)
IMC	Integrated Memory Controller
Intel® 64 Technology	64-bit memory extensions to the IA-32 architecture
Intel® Core™ i7	Intel's 45nm processor design, follow-on to the 45nm Penryn design
Intel® TXT	Intel® Trusted Execution Technology
Intel® VT-d2	Intel® Virtualization Technology (Intel® VT) for Directed I/O. Intel® VT-d is a hardware assist, under system software (Virtual Machine Manager or OS) control, for enabling I/O device virtualization. VT-d also brings robust security by providing protection from errant DMAs by using DMA remapping, a key feature of VT-d.
Intel® Virtualization Technology	Processor virtualization which when used in conjunction with Virtual Machine Monitor software enables multiple, robust independent software environments inside a single platform.
INTx	An interrupt request signal where X stands for interrupts A,B,C or D.
IOV	I/O Virtualization
LLC	Last Level Cache. The shared cache amongst all processor execution cores.
MCP	Multi-Chip Package
MLC	Mid-Level Cache
P2P	Peer-To-Peer, usually used to refer to Peer-To-Peer traffic flows

Table 2. Terminology (Sheet 2 of 2)

Term	Description
PCH	Platform Controller Hub. The new, 2009 chipset with centralized platform capabilities including the main I/O interfaces along with display connectivity, audio features, power management, manageability, security and storage features. The PCH may also be referred to by the name Intel® 3420 chipset.
PECI	Platform Environment Control Interface
Processor	The 64-bit, single-core or multi-core component (package)
Processor Core	The term “processor core” refers to a processing element containing an execution unit with its own instruction cache, data cache, and MLC. A die may contain one or more cores, all sharing one common LLC.
Rank	A unit of DRAM composed of and adequate number of memory chips in parallel so as to provide 64 bits of data or 72 bits data + ECC. These devices are usually mounted on a single side of a DIMM.
Resilvering	The process of re-synchronizing a memory channel that experienced an uncorrectable ECC error in a system utilizing mirroring of memory channels.
SAD	Source Address Decoder
SCI	System Control Interrupt. Used in ACPI protocol.
SMT	Simultaneous Multi-Threading.
SS engine	Sparing/Scrub engine
Storage Conditions	A non-operational state. The processor may be installed in a platform, in a tray, or loose. Processors may be sealed in packaging or exposed to free air. Under these conditions, processor landings should not be connected to any supply voltages, have any I/Os biased or receive any clocks. Upon exposure to “free air” (i.e., unsealed packaging or a device removed from packaging material) the processor must be handled in accordance with moisture sensitivity labeling (MSL) as indicated on the packaging material.
TAC	Thermal Averaging Constant
TDP	Thermal Design Power
TOM	Top of Memory
TTM	Time-To-Market
UP	Uni-processor
x1	Refers to a Link or Port with one Physical Lane
x4	Refers to a Link or Port with four Physical Lanes
x8	Refers to a Link or Port with eight Physical Lanes
x16	Refers to a Link or Port with sixteen Physical Lanes

1.9 Related Documents

See the following documents for additional information. Unless otherwise noted, obtain the documents from http://www.intel.com.

Table 3. Processor Documents

Document	Document Number
Intel® Xeon® Processor C5500/C3500 Series and LGA1366 Socket Thermal Mechanical Design Guide	323107
Voltage Regulator Module (VRM) and Enterprise Voltgage Regulator Down (EVRD) 11.1 Design Guidelines, Revision 1.5	397898

Table 4. PCH Documents

Document	Document Number
Ibex Peak Platform Controller Hub (PCH) - External Design Specification (EDS)	401376
Ibex Peak Platform Controller Hub (PCH) - Thermal Mechanical Specifications & Guidelines	407051

Notes:
1. Contact your Intel representative for the latest revision of this item.

Table 5. Public Specifications

Document	Document Number/ Location
Advanced Configuration and Power Interface Specification 3.0	http://www.acpi.info/
PCI Local Bus Specification 3.0	http://www.pcisig.com/specifications
PCI Express Base Specification 2.0	http://www.pcisig.com
DDR3 SDRAM Specification	http://www.jedec.org
Intel® 64 and IA-32 Architectures Software Developer's Manuals	See http://www.intel.com/products/processor/manuals/index.htm
Volume 1: Basic Architecture	253665
Volume 2A: Instruction Set Reference, A-M	253666
Volume 2B: Instruction Set Reference, N-Z	253667
Volume 3A: System Programming Guide	253668
Volume 3B: System Programming Guide	253669

§ §

2.0 Interfaces

This chapter describes the interfaces supported by the processor.

2.1 System Memory Interface

The complete list of supported memory configurations is preliminary, and is subject to change before product launch.

2.1.1 System Memory Technology Supported

The Intel® Xeon® processor C5500/C3500 series contains an integrated memory controller (IMC). The memory interface supports up to three DDR3 channels. Each channel consists of 64 bit data and 8 ECC bits. Up to three DIMMs can be connected to each DDR3 channel for a total of nine DIMMs per socket. The IMC supports DDR3 800 MT/s, DDR3 1066 MT/s and DDR3 1333 MT/s memory technologies. Three DIMMs can only be supported at 800 MT/s.

The processor supports up to three DIMMs per channel for single-rank and/or dual-rank DIMMs, and two DIMMs per channel for quad-rank DIMMs. See Table 6 through Table 8 for the supported configurations. A single system can be designed to support both single-rank and dual-rank configurations. To support both, three dual-rank DIMM configurations and two quad-rank DIMM configurations, and several control signals must be shared amongst DIMM connectors. The guidelines for control signal topologies are provided in the Picket Post Platform Design Guide.

Both registered ECC DDR3 DIMMs and Unbuffered DDR3 DIMMs are supported. (Unbuffered and registered DIMMs cannot be mixed.) Table 6 lists key IMC features, and Table 7 through Table 8 summarize the Intel® Xeon® processor C5500/C3500 series key differences for Unbuffered/Registered DIMM support.

Table 6. System Memory Feature Summary (Sheet 1 of 2)

Feature	Unbuffered DDR3	Independent Registered DDR3	Lockstepped Registered DDR3
Physical Channels per CPU Socket	3		
# Channels in use per CPU socket	1, 2, 3		2
DIMM Technology	DDR3 Unbuffered	DDR3 Registered	
ECC Support	ECC and non-ECC DIMMs		
Banks per Rank	Eight Independent		
DRAM Speeds	800, 1067, 1333		
DRAM Sizes	1 Gb, 2 Gb, 4 Gb		
DIMMs per Channel	1, 2	1, 2, 3	
Command/Address Rate	1N(1xCK), 2N(1/2xCK);		
Max Ranks per Channel	8		
Ranks per DIMM	1, 2	1, 2, 4	

Table 6. System Memory Feature Summary (Sheet 2 of 2)

Feature	Unbuffered DDR3	Independent Registered DDR3	Lockstepped Registered DDR3
Data Lines per DRAM	x8, x16	x4, x8	
Data Mask	No		
Lockstep Channel Support	No		Yes, Channels A and B Not supported with mirroring or sparing
Error Correction Code Capability	Correction for any error within a x4 DRAM and all connected data/strobe lines		Correction for any error within a x8 DRAM and all connected data/strobe lines
Each Cacheline Comes From	One Channel		Lockstepped Pair
Address Fault Detection	None	Address parity	Address parity + ECC
Latency	Baseline, critical word first optimizations	+3 UCLK, + .5 DCLK	+3 UCLK, + .5 DCLK
Page Policy	Open with adaptive idle timer or Closed Page		
Intel® QPI Priority	Yes	No	No
Graphics	No		
DIMM Sparing[2,3]	No	Yes, entire channel spared, within a socket only	No
Hot Add of DIMMs	No		
Hot Replace DIMMs	No		
Channel Mirroring[4]	No	Within a socket Channel A and B only	No
Demand Scrub[2]	If ECC is enabled	Yes	
Patrol Scrub[2]	If ECC is enabled	Yes	
Active and Precharge Power Down	Yes, no support for turning off DRAM DLLs in pre-charge power down		
Auto Refresh	Yes		
Throttling	Virtual Temp sensor with per command energies for bandwidth throttling and Open Loop throttling. Closed Loop throttling via DDR_THERM# pin.		
Dynamic 2X Refresh	Via MC_CLOSED_LOOP register. See the MC_Closed_Loop Register in Section 4.15.7 in Volume 2 of the Datasheet.		
Memory Init	Yes		
Memory Test	When ECC DIMMs are present	Yes	
Poisoning	Yes		
Asynchronous Self Refresh	No		

Notes:
2. x16 DRAM is not supported on combo routing.
3. Channel C can be used as a spare for channels on the same socket.
4. Between Channel A and B of the same socket. No resilvering to recover mirrored state after failure.

2.1.2 System Memory DIMM Configuration Support

- Table 7 summarizes the supported DIMM configurations for platforms that are designed with RDIMM only support.
- Table 8 summarizes the supported DIMM configurations for platforms that are designed with UDIMM only support.

Table 7. Intel® Xeon® Processor C5500/C3500 Series with RDIMM Only Support

DIMM Slots per Channel	DIMMs Populated per Channel	DIMM Type	POR Speeds	Ranks per DIMM (any combination)	Population Rules
2	1	Registered DDR3 ECC	800, 1066, 1333	SR, DR	Any combination of x4 and x8 RDIMMs, with 1 Gb, 2 Gb, or 4 Gb DRAM density. Populate DIMMs starting with slot 0, furthest from the CPU.
2	1	Registered DDR3 ECC	800, 1066	QR	
2	2	Registered DDR3 ECC	800, 1066	SR, DR	
2	2	Registered DDR3 ECC	800	SR, DR, QR	
3	1	Registered DDR3 ECC	800, 1066, 1333	SR, DR	Any combination of x4 and x8 RDIMMs, with 1 Gb, 2 Gb, or 4 Gb DRAM density. Populate DIMMs starting with slot 0, furthest from the CPU.
3	1	Registered DDR3 ECC	800, 1066	QR	
3	2	Registered DDR3 ECC	800, 1066	SR, DR	
3	2	Registered DDR3 ECC	800	SR, DR, QR	
3	3	Registered DDR3 ECC	800	SR, DR	

Table 8. UDIMM Only Support

DIMM Slots per Channel	DIMMs Populated per Channel	DIMM Type	POR Speeds	Ranks per DIMM (any combination)	Population Rules
2	1	Unbuffered DDR3 (w/ or w/o ECC)	800, 1066, 1333	SR, DR	Any combination of x8 and x16 UDIMMs, with 1 Gb, 2 Gb, or 4 Gb DRAM density. Populate DIMMs starting with slot 0, furthest from the CPU.
2	2	Unbuffered DDR3 (w/ or w/o ECC)	800, 1066	SR, DR	
3	1	Unbuffered DDR3 (w/ or w/o ECC)	800, 1066, 1333	SR, DR	
3	2	Unbuffered DDR3 (w/ or w/o ECC)	800, 1066	SR, DR	

2.1.3 System Memory Timing Support

The IMC supports the following DDR3 Speed Bin, CAS Write Latency (CWL), and command signal mode timings on the main memory interface:

- tCL = CAS Latency
- tRCD = Activate Command to READ or WRITE Command delay
- tRP = PRECHARGE Command Period
- CWL = CAS Write Latency
- Command Signal modes = 1n indicates a new command may be issued every clock and 2n indicates a new command may be issued every two clocks. Command launch mode programming depends on the transfer rate and memory configuration.

Table 9. DDR3 System Memory Timing Support

Transfer Rate (MT/s)	tCL (tCK)	tRCD (tCK)	tRP (tCK)	CWL (tCK)	CMD Mode	Notes
800	5	5	5	5	1n and 2n	1
800	6	6	6	5	1n and 2n	1
1066	7	7	7	6	1n and 2n	1
	8	8	8			
1333	8	8	8	7	2n	1
1333	9	9	9	7	2n	1

Notes:
1. System Memory timing support is based on availability and is subject to change.

2.1.3.1 System Memory Operating Modes

The IMC contains three DDR3 channel controllers. Up to three channels can be operated independently or two channels (only channels A and B) can be paired for lockstep or mirroring. The DRAM controllers share a common address decode and DMA engines for RAS features. Configuration registers may be per channel or common. Each DRAM controller has a scheduler, write and read data paths, ECC logic and auxiliary structures.

Resilvering is not supported. A single block of logic is used to support Scrubbing and Sparing, therefore these functions cannot be carried out simultaneously. To spare a 16 GB channel may take up to 40 seconds. The memory must be initialized to a valid ECC before either patrol scrubbing or demand scrubbing can be enabled. The patrol scrub rate is programmable. If the patrol scrub rate was programmed to one line every 82 ms, 64 GB would require one day to fully scrub once.

All IMC errors are categorized as either Corrected, Uncorrected Non-Fatal error (e.g. Patrol scrub read), or Fatal. The IMC can be programmed to treat Uncorrected Non-Fatal errors as Fatal. Corrected errors, including uncorrected errors on a mirrored channel that are “corrected” by switching to a working partner, will not assert any signal. These errors must be monitored by SW.

Any IMC uncorrected errors will be fatal. An asynchronous Machine Check exception is signaled and the error is logged. The IMC can be configured to send a poison indication with any uncorrectable error, this can be used to achieve system level error containment.

Read and Write addresses are steered according to address decode to one of the three channels or a pair of channels (in the mirroring or lockstep cases). The channels decompose the reads and writes into precharge, activate, and column commands and issue these commands on the DDR interface as command and address lines. Write data is enqueued in the IMC write data buffers where partial writes are merged to form full line writes. Read returns from the three channels are corrected if necessary, then multiplexed back to the IMC read data buffer.

The memory channels are treated as logical channels. That is, write requests credits to the IMC are maintained on a logical channel basis. The memory controller may translate the channel select sent to one or two physical channels. The register that controls the mapping of logical channels to physical channels is described in the register section (see Volume 2 of the Datasheet). In addition, the conditions under which software or hardware can modify this mapping is also described. Table 10 summarizes how the logical to physical channel mappings are made.

Table 10. Mapping from Logical to Physical Channels

Channel Mode	Mirroring	Logical to Physical
Independent Channels	Disabled	1:1 relationship, but may not be the same number.
	Enabled	A pair of physical channels are combined to form a single logical redundant channel. Requests to logical channel A are handled by physical channels A and B.
Lockstep	Disabled	A pair of physical channels are accessed in parallel to form a single logical channel. Lockstep of arbitrary physical channels is not supported. Physical channel A provides half of the data for each request to Logical Channel A. Physical channel B provides the other half.

2.1.3.2 Single-Channel Mode

In this mode, all memory cycles are directed to a single-channel. Single-channel mode is used when only a single channel is populated with memory.

2.1.3.3 Independent Channel Mode

In this mode one, two, or all three channels operate independently. Each channel stores one complete cache line per transfer. When ECC is used with x4 DRAM devices, a failure of an entire x4 DRAM device can be corrected, x8 DRAMs can also be used but not all bit failures can be corrected, and x8 device failures are not correctable. The correction capabilities in independent mode are:

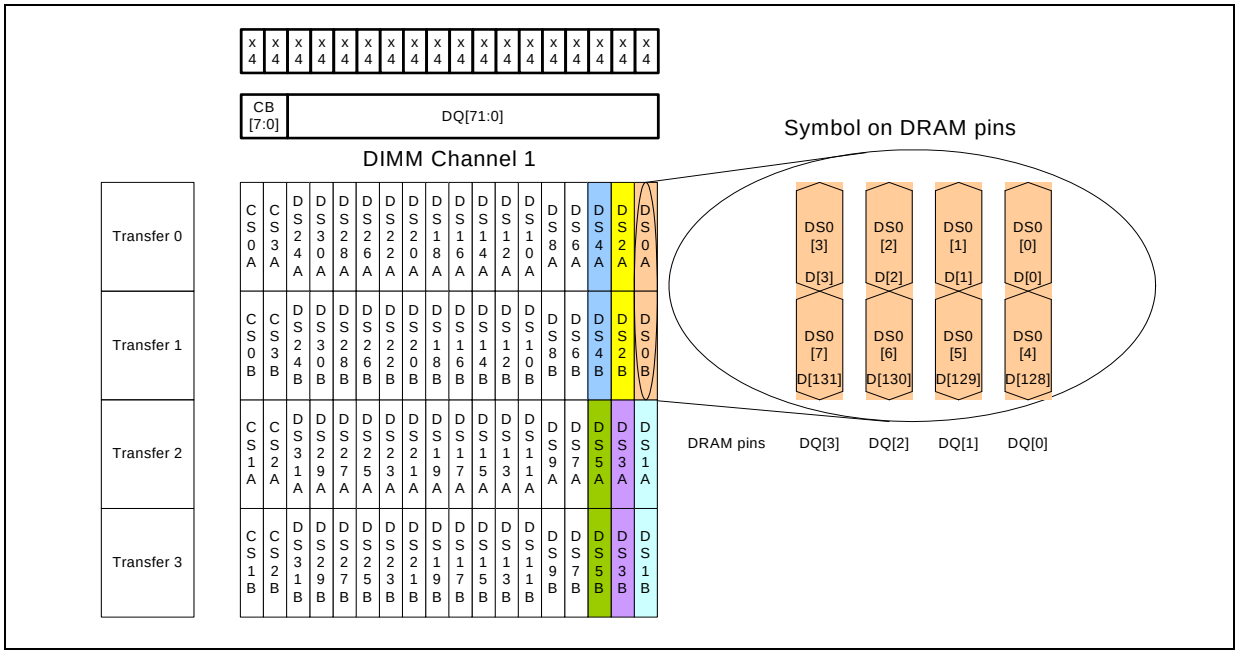
- Correction of any x4 DRAM device failure.
- Detection of 99.986% of all single bit failures that occur in addition to a x4 DRAM failure.
- Detection of all 2-bit uncorrectable errors.

This mode supports the most flexibility with respect to DIMM populations, and bandwidth performance.

Figure 3 shows how the symbols are mapped to DRAM bits on the DIMM for a transfer in which the critical 16 B is in the lower half of the codeword (A[4]=0). If the upper portion of the codeword were transferred first, bits[7:4] of each symbol would be transferred first on the DRAM interface.

The lower nibble of the symbol (DS0A) consists of DS0[3:0] and the upper nibble (DS0B) consists of DS0[7:4]. On the DRAM interface, DS0 is expanded to show that it occupies 4 DRAM lines for two transfers. DS0[3:0] appear in the first transfer. DS0[7:4] appear in the second transfer. DS0 and DS1 are the adjacent symbols that protect all four transfers in the codeword on the four lines from the first DRAM on DIMM0.

Figure 3. Independent Code Layout



2.1.3.4 Spare Channel Mode

In this mode, channels A and B operate as independent channels, with channel C functioning as a spare should either channels A or B fail. When ECC is used, error correction/detection on a single channel is the same as provided by Independent Channel Mode. The IMC initiates a sparing copy from a failed channel to the spare channel, or the SW can initiate a sparing copy if a specific channel is experiencing a high rate of correctable errors.

The Integrated Memory Controller will maintain correctable ECC error counters for each DIMM in the system that can either trigger an SMI event or be periodically polled by software to determine whether a high error rate is happening. Software can then configure the Integrated Memory Controller to copy contents from one channel to another.

While performing a sparing copy, the Integrated Memory Controller operates as follows:

- When software initiates a Sparing operation, the Integrated Memory Controller copies data from one channel to the other. The SS (Sparing/Scrub) engine performs operations in the DRAM Address space indicated by DA(x), where x is a system address.
- Software controls entry into this mode by disabling scrubbing and writing the SSR control register with source and destination channel IDs.
- If the operation succeeds without uncorrectable error, the Integrated Memory Controller will set the SSR Copy Complete (CMPLT) bit in the MC_SSRSTATUS register.
- System memory writes are duplicated to each channel while data is copied from the channel specified by the SRC_CHAN parameter to the channel specified by the DEST_CHAN parameter in the MC_SSRCONTROL register.

2.1.3.5 Mirrored Channel Mode

The following modes of operation are required to implement mirroring.

2.1.3.5.1 Mirroring Redundant Mode

Software puts the Integrated Memory Controller into this mode whenever the memory image in both channels is identical, or where they differ, the contents are not valid. In addition, both WDBs must be empty before enabling this mode.

The final step in mirroring is to inform the channels that they are now redundant, so that they stop signaling uncorrectable errors when they fail. After the mirroring setup is complete, the state of the mirrored channels can be changed from active to redundant. It is not critical to minimize the time between changing each channel to redundant state. No inconsistency results if one channel is in redundant state and the other is not when a failure occurs. If the non-redundant channel fails, it is fatal. If the redundant channel fails, it will transfer operation to the non-redundant channel. In this mode, the Integrated Memory Controller duplicates writes to both channels.

Reads are sent to one channel or the other, as described in the channel mapper.

Uncorrectable errors in this mode are logged and signaled as correctable, but change the channel state to Disabled, and the working partner to Redundancy Loss.

The BIOS follows this sequence to set up mirroring mode:

- channel active
- init done
- mem init
- mirror enable
- channel map
- smi enable
- mem config hide

2.1.3.5.2 Disabled Channel Operation

After an uncorrectable error, the logical channel disables itself. However, to support continued operation, the logical channel must complete handshakes for any requests it receives. No more channel errors will be logged. The channel behaves as if the result is correct.

The failed channel resets its columns in the channel mapper so that all subsequent requests are routed to the working partner. The coupling of channels for credit return must be removed. Write credits will be returned as soon as the working partner provides them.

2.1.3.5.3 Redundancy Loss Mode

The Integrated Memory Controller changes the state of the working channel to redundancy loss when its partner fails.

The failed channel clears its bits in the channel mapper, so that all accesses will be directed to the working channel. The working channel will enter redundancy loss state. The failed channel will enter disabled state.

If any uncorrectable errors on the working channel are detected in the same clock or later than the uncorrectable error that caused the loss of redundancy, then they must be signaled as uncorrectable. This requires that error signaling be delayed long enough

to receive inputs from a mirrored partner. If both channels fail simultaneously, an uncorrectable error must be signaled. Mirror mode only recovers from a single error (Resilvering is not supported).

2.1.3.6 Lockstep Mode

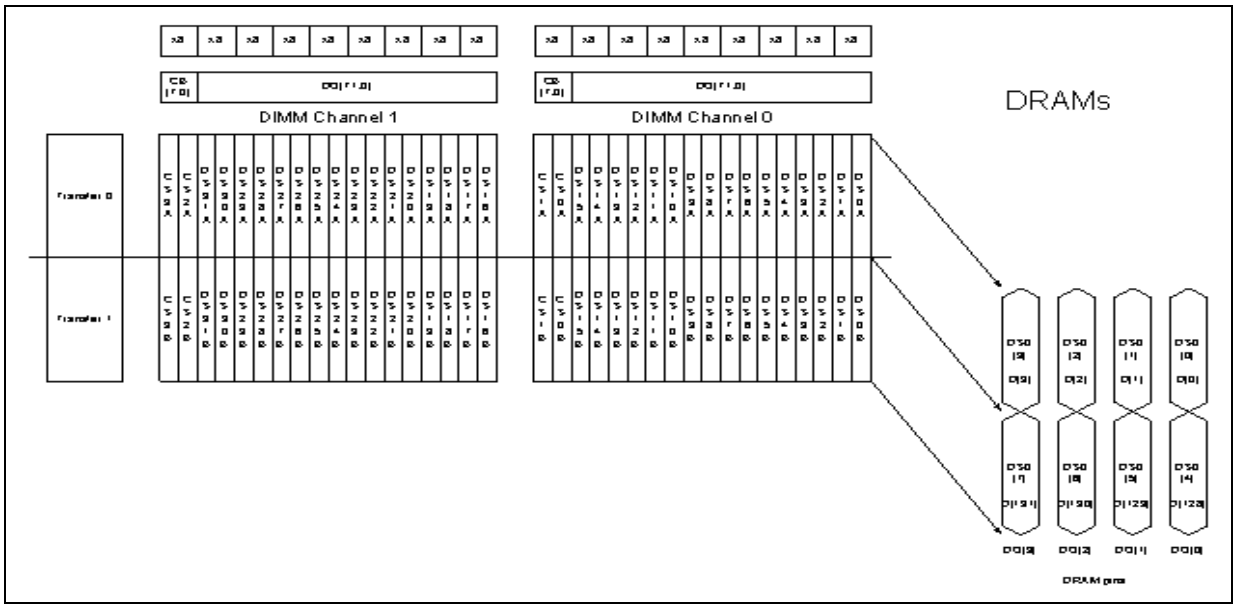
Lockstep Mode refers to splitting cache lines across channels. In this mode, the same address is used on both channels, and an error on either channel is detected. The ECC code used by the memory controller can correct 4 bits out of 72 bits. Since a single DIMM is 72 bits wide (64 bits of data and 8 bits of ECC), in order to correct an entire x8 DRAM device, the 72 bit transfer is split across two channels. The IMC always (ECC enabled or not) accumulates 32 bytes of data before forwarding to memory, therefore, there is no latency penalty for enabling ECC. The correction capabilities in lockstep mode are:

- Correction of any x4 or x8 DRAM device failure.
- Detection of 99.986% of all single bit failures that occur in addition to a x8 DRAM failure. The Integrated Memory Controller will detect a series of failures on a specific DRAM and use this information in addition to the information provided by the code to achieve 100% detection of these cases.
- Detection of all permutations of two x4 DRAM failures.

Figure 4 shows where each bit of the ECC code appears in a pair of lockstepped channels. The symbols are arranged so that the data from every x8 DRAM is mapped to two adjacent symbols, so any failure of the DRAM can be corrected.

Figure 4 traces the bits of Data Symbol 0 (DS0) from DRAM. The lower nibble of the symbol (DS0A) consists of DS0[3:0] and the upper nibble (DS0B) consists of DS0[7:4]. On the DRAM interface, DS0 is expanded to show that it occupies four DRAM lines for two transfers. DS0[3:0] appears in the first transfer. DS0[7:4] appear in the second transfer. DS0 and DS1 are the adjacent symbols that protect the eight lines from the first DRAM on DIMM0.

Figure 4. Lockstep Code Layout



2.1.3.6.1 Limitations

Lockstepped channels must be populated identically. That is, each DIMM in one channel must have an identical corresponding DIMM in the alternate channel; identical in number ranks, banks, rows, and columns. DIMMs may be of different speed grades, but the memory controller will be configured to operate all DIMMs according to the slowest parameters present. Only channels A and B support lockstep, the third channel is unused in lockstep mode.

In lockstep mode, the memory controller will align read data to the slowest lane on both channels. Read data received at different times will be buffered until both channels complete their return. If either channel needs to throttle, both are throttled. A common configuration control bit is used to enable refresh on both channels.

2.1.3.7 Dual/Triple - Channel Modes

The IMC supports three types of dual/triple-channel, memory addressing modes; Dual/Triple - Channel Symmetric (Interleaved), Dual/Triple-Channel Asymmetric, and Intel® Flex Memory mode.

2.1.3.7.1 Triple/Dual-Channel Symmetric Mode

Also known as interleaved mode, and provides maximum performance on real world applications. Addresses are ping-ponged between the channels after each cache line (64-byte boundary). If there are two requests, and the second request is to an address on the opposite channel from the first, that request can be sent before data from the first request has returned. If two consecutive cache lines are requested, both may be retrieved simultaneously, since they are ensured to be on opposite channels. Use Dual-Channel Symmetric mode when both Channel A and Channel B DIMM connectors are populated in any order, with the total amount of memory in each channel being the same. Use Triple-Channel Symmetric mode when both Channel A, Channel B, and Channel C DIMM connectors are populated in any order, with the total amount of memory in each channel being the same.

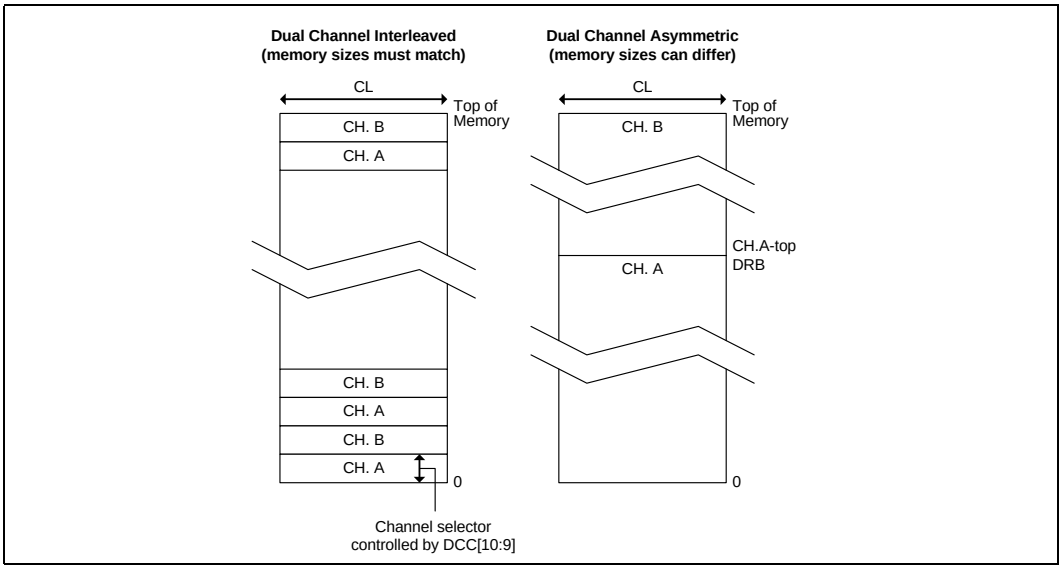
Note: The DRAM device technology and width may vary from one channel to the other.

2.1.3.7.2 Triple/Dual-Channel Asymmetric Mode

This mode trades performance for system design flexibility. Unlike the previous mode, addresses start in Channel A and stay there until the end of the highest rank in Channel A, and then addresses continue from the bottom of Channel B to the top, etc. Real world applications are unlikely to make requests that alternate between addresses that sit on opposite channels with this memory organization, so in most cases, bandwidth is limited to a single channel.

This mode is used when Intel® Flex Memory Technology is disabled and both Channel A, Channel B, and Channel C DIMM connectors are populated in any order with the total amount of memory in each channel being different.

Figure 5. Dual-Channel Symmetric (Interleaved) and Dual-Channel Asymmetric Modes

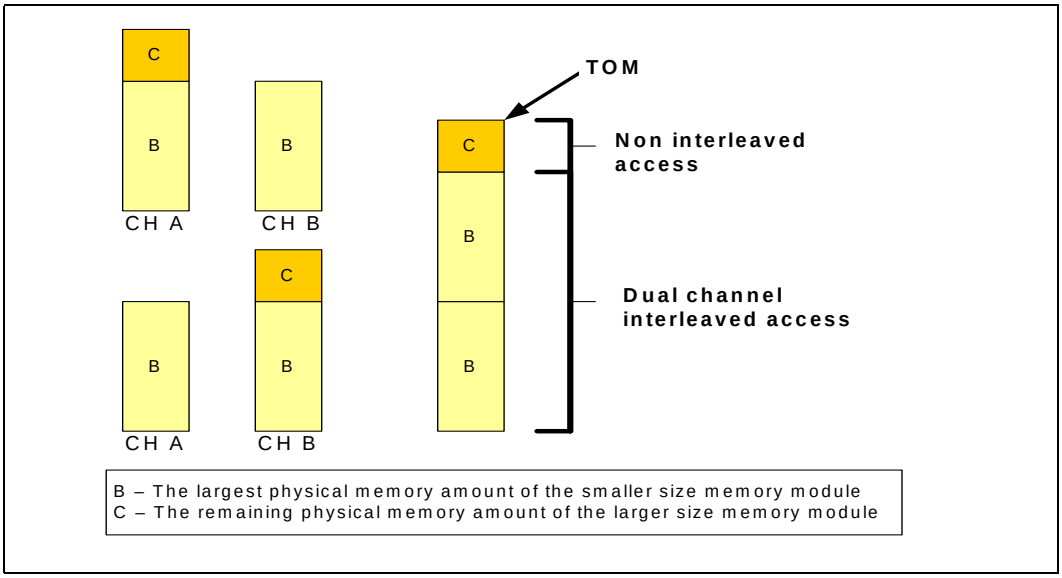


2.1.3.7.3 Intel® Flex Memory Technology Mode

This mode combines the advantages of the Dual/Triple-Channel Symmetric (Interleaved) and Dual/Triple-Channel Asymmetric Modes. Memory is divided into a symmetric and a asymmetric zone. The symmetric zone starts at the lowest address in each channel and is contiguous until the asymmetric zone begins or until the top address of the channel with the smaller capacity is reached. In this mode, the system runs at one zone of dual/triple-channel mode and one zone of single-channel mode, simultaneously, across the whole memory array.

This mode is used when Intel® Flex Memory Technology is enabled and both Channel A, Channel B, and Channel C DIMM connectors are populated in any order with the total amount of memory in each channel being different.

Figure 6. Intel® Flex Memory Technology Operation



2.1.4 DIMM Population Requirements

In all modes, the frequency of system memory is the lowest frequency of all memory modules placed in the system, as determined through the SPD registers on the memory modules.

2.1.4.1 General Population Requirements

All DIMMs must be DDR3 DIMMs. Registered DIMMs must be ECC only; Unbuffered DIMMs can be ECC or non-ECC. Mixing Registered and Unbuffered DIMMs is not allowed. It is allowed to mix ECC and non-ECC Unbuffered DIMMs. The presence of a single non-ECC Unbuffered DIMM will result disabling of ECC functionality.

DIMMs with different timing parameters can be installed on different slots within the same channel, but only timings that support the slowest DIMM will be applied to all. As a consequence, faster DIMMs will be operated at timings supported by the slowest DIMM populated. The same interface frequency (DDR3-800, DDR3-1066, or DDR3-1333) will be applied to all DIMMs on all channels.

For DP configurations, there is no relationship or requirements between DIMMs installed in different sockets. That is, the IMC from one socket may be populated differently than the IMC of the alternate socket except that the DIMMs must be of the same type, i.e. either UDIMM or RDIMM.

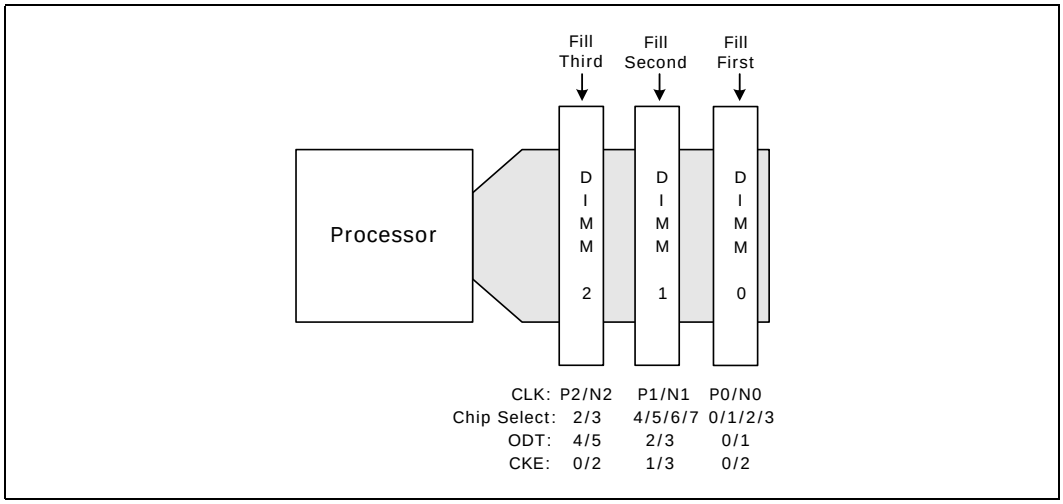
2.1.4.2 Populating DIMMs Within a Channel

2.1.4.2.1 DIMM Population for Three Slots per Channel

For three DIMM slots per channel configurations, the processor requires DIMMs within a channel to be populated starting with the DIMM slot furthest from the processor in a “fill-furthest” approach (see Figure 7).

When populating a Quad-rank DIMM with a Single- or Dual-rank DIMM in the same channel, the Quad-rank DIMM must be populated farthest from the processor. Quad-rank DIMMs and UDIMMs are not allowed in three slots populated configurations. Intel recommends checking for correct DIMM placement during BIOS initialization. Additionally, Intel strongly recommends that all designs follow the DIMM ordering, command clock, and control signal routing documented in Figure 7. This addressing must be maintained to be compliant with the reference BIOS code supplied by Intel. All allowed DIMM population configurations for three slots per channel are shown in Table 11 and Table 12.

Figure 7. DIMM Population Within a Channel



Note: ODT[5:4] is muxed with CS[7:6]#.

Table 11. RDIMM Population Configurations Within a Channel for Three Slots per Channel

Configuration Number	POR Speed	1N or 2N	DIMM2	DIMM1	DIMM0
1	DDR3-1333, 1066, & 800	1N	Empty	Empty	Single-rank
2	DDR3-1333, 1066, & 800	1N	Empty	Empty	Dual-rank
3	DDR3-1066[1] & 800	1N	Empty	Empty	Quad-rank
4	DDR3-1066[1] & 800	1N	Empty	Single-rank	Single-rank
5	DDR3-1066[1] & 800	1N	Empty	Single-rank	Dual-rank
6	DDR3-1066[1] & 800	1N	Empty	Dual-rank	Single-rank
7	DDR3-1066[1] & 800	1N	Empty	Dual-rank	Dual-rank
8	DDR3-800	1N	Empty	Single-rank	Quad-rank
9	DDR3-800	1N	Empty	Dual-rank	Quad-rank
10	DDR3-800	1N	Empty	Quad-rank	Quad-rank
11	DDR3-800	1N	Single-rank	Single-rank	Single-rank
12	DDR3-800	1N	Single-rank	Single-rank	Dual-rank
13	DDR3-800	1N	Single-rank	Dual-rank	Single-rank
14	DDR3-800	1N	Dual-rank	Single-rank	Single-rank
15	DDR3-800	1N	Single-rank	Dual-rank	Dual-rank
16	DDR3-800	1N	Dual-rank	Single-rank	Dual-rank
17	DDR3-800	1N	Dual-rank	Dual-rank	Single-rank
18	DDR3-800	1N	Dual-rank	Dual-rank	Dual-rank

Note:
1. If DD3-1333 speed DIMM is populated, BIOS will configure it at DDR3-1066 speed.

Table 12. UDIMM Population Configurations Within a Channel for Three Slots per Channel

Configuration Number	POR Speed	1N or 2N	DIMM2	DIMM1	DIMM0
1	DDR3-1333, 1066, & 800	1N	Empty	Empty	Single-rank
2	DDR3-1333, 1066, & 800	1N	Empty	Empty	Dual-rank
3	DDR3-1066 & 800	2N	Empty	Single-rank	Single-rank
4	DDR3-1066 & 800	2N	Empty	Single-rank	Dual-rank
5	DDR3-1066 & 800	2N	Empty	Dual-rank	Single-rank
6	DDR3-1066 & 800	2N	Empty	Dual-rank	Dual-rank

2.1.4.2.2 DIMM Population for Two Slots per Channel

For two DIMM slots per channel configurations, the processor requires DIMMs within a channel to be populated starting with the DIMM slots furthest from the processor in a "fill-furthest" approach (see Figure 8). In addition, when populating a Quad-rank DIMM with a Single- or Dual-rank DIMM in the same channel, the Quad-rank DIMM must be populated farthest from the processor. Intel recommends checking for correct placement during BIOS initialization. Additionally, Intel strongly recommends that all designs follow the DIMM ordering, command clock, and control signal routing documented in Figure 8. This addressing must be maintained to be compliant with the reference BIOS code supplied by Intel. All allowed DIMM population configurations for two slots per channel are shown in Table 13 and Table 14.

Figure 8. DIMM Population Within a Channel for Two Slots per Channel

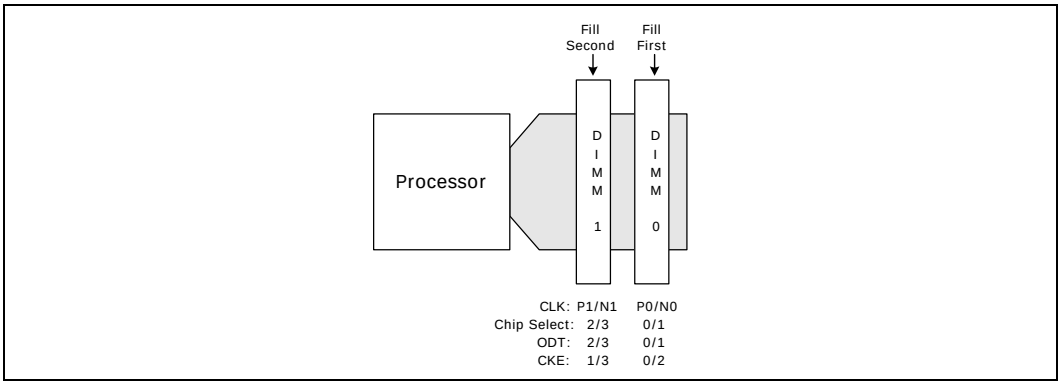


Table 13. DIMM Population Configurations Within a Channel for Two Slots per Channel (Sheet 1 of 2)

Configuration #	POR Speed	1N or 2N	DIMM1	DIMM0
1	DDR3-1333, 1066, & 800	1N	Empty	Single-rank
2	DDR3-1333, 1066, & 800	1N	Empty	Dual-rank
3	DDR3-1066 & 800	1N	Empty	Quad-rank
4	DDR3-1066 & 800	1N	Single-rank	Single-rank
5	DDR3-1066 & 800	1N	Single-rank	Dual-rank
6	DDR3-1066 & 800	1N	Dual-rank	Single-rank
7	DDR3-1066 & 800	1N	Dual-rank	Dual-rank

Table 13. DIMM Population Configurations Within a Channel for Two Slots per Channel (Sheet 2 of 2)

Configuration #	POR Speed	1N or 2N	DIMM1	DIMM0
8	DDR3-800	1N	Single-rank	Quad-rank
9	DDR3-800	1N	Dual-rank	Quad-rank
10	DDR3-800	1N	Quad-rank	Quad-rank

Table 14. UDIMM Population Configurations Within a Channel for Two Slots per Channel

Configuration #	POR Speed	1N or 2N	DIMM1	DIMM0
1	DDR3-1333, 1066, & 800	1N	Empty	Single-rank
2	DDR3-1333, 1066, & 800	1N	Empty	Dual-rank
3	DDR3-1066 & 800	2N	Single-rank	Single-rank
4	DDR3-1066 & 800	2N	Single-rank	Dual-rank
5	DDR3-1066 & 800	2N	Dual-rank	Single-rank
6	DDR3-1066 & 8003	2N	Dual-rank	Dual-rank

2.1.4.3 Channel Population Requirements for Memory RAS Modes

The Intel® Xeon® processor C5500/C3500 series supports four different memory RAS modes: Independent Channel Mode, Spare Channel Mode, Mirrored Channel Mode, and Lockstep Channel Mode. The rules on channel population and channel matching vary by the RAS mode used. Regardless of RAS mode, requirements for populating within a channel given in Section 2.1.4.2 must be met at all times. Support of RAS modes requiring matching DIMM population between channels (Sparing, Mirroring, Lockstep) require that ECC DIMMs be populated. Independent Mode only supports non-ECC DIMMs in addition to ECC DIMMs.

For RAS modes that require matching populations, the same slot positions across channels must hold the same DIMM type with regards to size and organization. DIMM timings do not have to match but timings will be set to support all DIMMs populated (i.e., DIMMs with slower timings will force faster DIMMs to the slower common timing modes). Intel recommends checking for correct DIMM matching, if applicable to the RAS mode, during BIOS initialization.

2.1.4.3.1 Independent Channel Mode

Channels can be populated in any order in Independent Channel Mode. All three channels may be populated in any order and have no matching requirements. All channels must run at the same interface frequency, but individual channels may run at different DIMM timings (RAS latency, CAS latency, etc.).

2.1.5 Technology Enhancements of Intel® Fast Memory Access (Intel® FMA)

The following sections describe the Just-in-Time Scheduling, Command Overlap, and Out-of-Order Scheduling Intel® FMA technology enhancements.

2.1.5.1 Just-in-Time Command Scheduling

The memory controller has an advanced command scheduler where all pending requests are examined simultaneously to determine the most efficient request to be issued next. The most efficient request is picked from all pending requests and issued

to system memory Just-in-Time to make optimal use of Command Overlapping. Thus, instead of having all memory access requests go individually through an arbitration mechanism forcing requests to be executed one at a time, they can be started without interfering with the current request allowing for concurrent issuing of requests. This allows for optimized bandwidth and reduced latency while maintaining appropriate command spacing to meet system memory protocol.

2.1.5.2 Command Overlap

Command Overlap allows the insertion of the DRAM commands between the Activate, Precharge, and Read/Write commands normally used, as long as the inserted commands do not affect the currently executing command. Multiple commands can be issued in an overlapping manner, increasing the efficiency of system memory protocol.

2.1.5.3 Out-of-Order Scheduling

While leveraging the Just-in-Time Scheduling and Command Overlap enhancements, the IMC continuously monitors pending requests to system memory for the best use of bandwidth and reduction of latency. If there are multiple requests to the same open page, these requests would be launched in a back to back manner to make optimum use of the open memory page. This ability to reorder requests on the fly allows the IMC to further reduce latency and increase bandwidth efficiency.

2.1.6 DDR3 On-Die Termination

On-Die Termination (ODT) allows a DRAM device to turn on/off internal termination resistance for each DQ, DQS/DQS#, and DM signal via the ODT control pin.

ODT provides improved signal integrity of the memory channel by allowing the DRAM controller to independently turn on or off the termination resistance for any or all DRAM devices themselves instead of on the motherboard.

The IMC drives out the required ODT signals, based on the memory configuration and which rank is being written to or read from, to the DRAM devices on a targeted DIMM module rank to enable or disable their termination resistance.

2.1.7 Memory Error Signaling

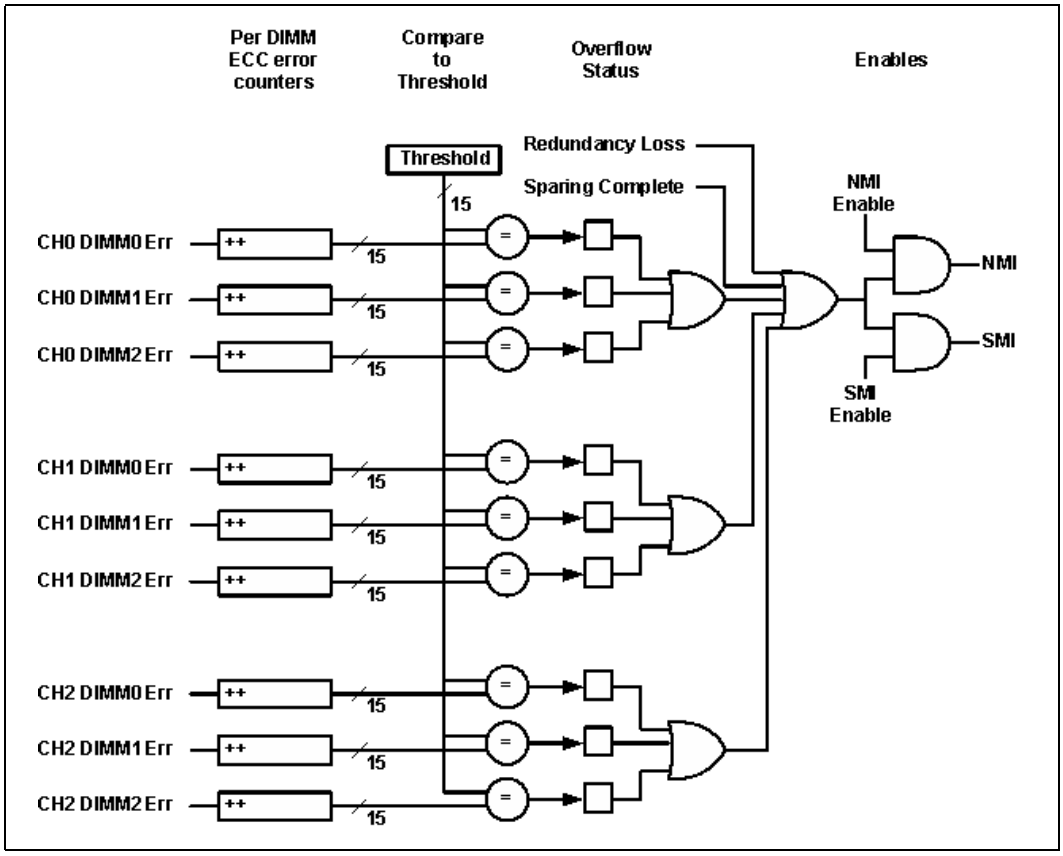
Uncorrected memory errors are reported via Machine Check Architecture. An uncorrected memory error is logged in Machine Check Bank8 registers, causes a Machine Check Exception (MCE) signaled to all processor packages, and asserts CATERR#, which can be optionally used by a platform to trigger an SMI event.

Corrected memory errors are reported via two independent mechanisms: CMCI signaling based on Machine Check Architecture and Machine Check Bank8 registers, and SMI/NMI signaling based on CSR registers located in the Integrated Memory Controller.

CMCI and Machine Check Architecture based memory error signaling is intended to be handled by the OS. This subsection covers the SMI/NMI signaling of corrected memory errors based on CSR registers.

Figure 9 depicts this logic.

Figure 9. Error Signaling Logic



2.1.7.1 Enabling SMI/NMI for Memory Corrected Errors

The MC_SMI_SPARE_CNTRL register has enables for SMI and NMI interrupts. Only one should be set. Whichever type of interrupt is enabled will be triggered if:

- a DIMM error counter exceeds the threshold,
- redundancy is lost on a mirrored configuration, or
- a sparing operation completes.

This register is set by hardware once operation is complete. Bit is cleared by hardware when a new operation is enabled. An SMI is generated when this bit is set due to a sparing copy completion event.

Such an interrupt, once enabled by software, will be signaled only to the local processor package where these events were detected. Therefore, the SMI/NMI interrupt handler must be aware of the fact that the other processor package, if present, did not receive the signalling of such SMI/NMI event.

2.1.7.2 Per DIMM Error Counters

There is one correctable ECC error counter for each DIMM that can be connected to the Integrated Memory Controller. There are six MC_COR_ECC_CNT_X registers, each of which holds a 15-bit counter and overflow bits for two DIMMs. The

MC_SMI_SPARE_CNTRL[1] register holds an SMI_ERROR_THRESHOLD[1] to which the counters are compared. If any counter exceeds the threshold, the enabled interrupt will be generated, and status bits are set to indicate which counter met threshold.

2.1.7.3 Identifying the Cause of An Interrupt

Table 15 defines how to determine what caused the interrupt.

Table 15. Causes of SMI or NMI

Condition	Cause	Recommended platform software response.
MC_SMI_SPARE_DIMM_ERROR_STATUS. DIMM_ERROR_OVERFLOW_STATUS != 0	This register has one bit for each DIMM error counter that meets threshold. This can happen at the same time as any of the other SMI events (Sparing complete, redundancy lost in Mirror Mode). It is recommended that software address one, so that the other cause remains when the second event is taken.	Examine the associated MC_COR_ECC_CNT_X register. Determine the time since the counter has been cleared. If a spare channel exists, and the threshold has been exceeded faster than would be expected given the background rate of correctable errors, Sparing should be initiated. The counter should be cleared to reset the overflow bit.
MC_RAS_STATUS.REDUNDANCY_LOSS = 1	One channel of a mirrored pair had an uncorrectable error and redundancy has been lost.	Raise an indication that a reboot should be scheduled, possibly replace the failed DIMM specified in the MC_SMI_SPARE_DIMM_ERROR_STATUS register. (Not present on Astep)
MC_SSRSTATUS. CMPLT = 1	A sparing copy operation set up by software has completed.	Advance to the next step in the sparing flow.

2.1.8 Single Device Data Correction (SDDC) Support

The Integrated Memory Controller employs a Single Device Data Correction (SDDC) algorithm that will recover from a x4/x8 component failure. In addition the Integrated Memory Controller supports demand and patrol scrubbing.

A scrub corrects a correctable error in memory. A four-byte ECC is attached to each 32-byte “payload”. An error is detected when the ECC calculated from the payload mismatches the ECC read from memory. The error is corrected by modifying either the ECC or the payload or both and writing both the ECC and payload back to memory. Only one demand or patrol scrub can be in process at a time.

2.1.9 Patrol Scrub

Patrol scrubs are intended to ensure that data with a correctable error does not remain in DRAM long enough to stand a significant chance of further corruption to an uncorrectable error due to particle error. The Integrated Memory Controller will issue a Patrol Scrub at a rate sufficient to write every line once a day. For a maximum capacity of 64 GB, this would be one scrub every 82 ms. The Sparing/Scrub (SS) engine sends scrubs to one channel at a time. The Patrol Scrub rate is configurable. The scrub engine will scrub all active channels which includes the spare channel. The spare channel will be scrubbed and errors will be signaled and logged if errors are enabled.

2.1.10 Memory Address Decode

Memory address decode is the process of taking a system address and converting it to rank, bank, row and column address on a memory channel. Memory address decode is performed in two levels. The first level selects the socket (in DP systems) and memory channel, and generates a channel address. The second level decodes the channel address into the rank, bank, row and column addresses.

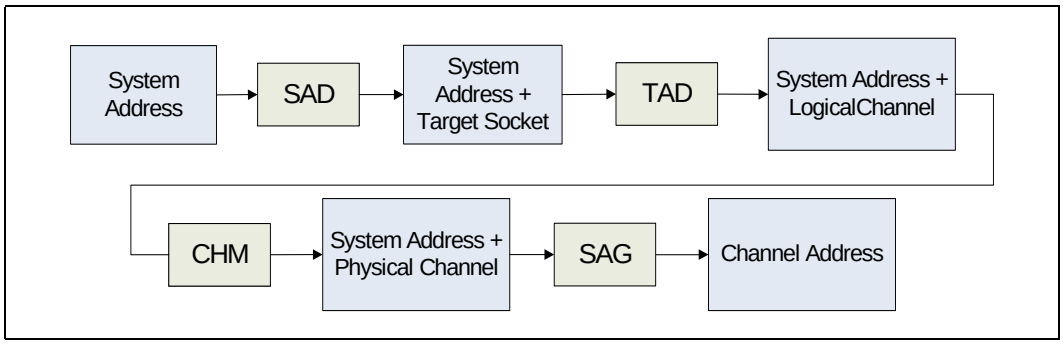
2.1.10.1 First Level Decode

Figure 10 below shows the address decode flow. First the system address is sent to the Source Address Decoder (SAD) to determine the target socket and Intel® QuickPath Interconnect node ID. The SAD also determines if a transaction will target memory or MMIO. The remainder of this section assumes the address targets system memory. See the System Address Decoder Registers in the Uncore (device 0, function 1 in uncore) and the QPIMADCTRL, QPIMADDATA and QPIPINT registers in the IIO (device 16, function 1 in the IIO) for details on the programming of the uncore and IIO SAD.

After the requested is routed to the appropriate socket, the Target Address Decoder (TAD) determines the logical memory channel that will service the request. The TAD control registers are located in device 3, function 1 in the uncore logic.

The Channel Mapping logic (CHM) is used to determine the physical channel which will service the request. The operating mode of the memory controller (Independent, Mirroring, Lockstep) will determine how logical channels are mapped to physical channels. See the MC_CHANNEL_MAPPER register in device 3, function 0 of the uncore.

Figure 10. First Level Address Decode Flow



Finally, the System Address Gap logic (SAG) is used to “squeeze out the gaps” and convert the system address to a contiguous set of address for a channel. For example on a system with a 2 channel interleave, it is possible that a given memory channel would service every other cacheline with odd cachelines going to one channel and even cachelines to the other. The SAG translates this every other cacheline system address a single channel receives to a contiguous set of channel addresses. The channel address will range from 0 to the number of bytes on that channel minus -1. There is a SAG per memory channel, see the MC_SAG_CH[2:0]_[7:0] registers in the uncore for programming details. The channel address that is the output of the SAG is the final stage of the first level address decode.

2.1.10.1.1 Address Ranges

Level 1 decode supports eight memory ranges. Each range defines a contiguous block of addresses which target either memory or MMIO (not both). Within a range, there is only one socket and channel interleave that describes how the addresses are spread between the memory channels. Different ranges may use different interleaving schemes.

2.1.10.1.2 Channel Interleaving

Cache lines (linearly increasing addresses) in an address range can interleave across 1, 2, 3, 4, or 6 memory channels.

2.1.10.1.3 Logical to Physical Channel Mapping

The MC_CHANNEL_MAPPER register and the lockstep bit define the mapping of logical channels decoded by the first level decode and physical channels in the Integrated Memory Controller. The MC_CHANNEL_MAPPER register is set to direct reads or writes from one logical channel to any two physical channels as required for sparing or mirroring. After the MC_CHANNEL_MAPPER bits take effect, the lockstep bit directs any read or write that is destined to physical Channel A, to physical Channel B as well.

These bits can be read and written by software. The Integrated Memory Controller will only modify these bits when a mirrored channel fails. In that case, the bit corresponding to the failed channel will be cleared.

There is one bit for each physical channel and separate fields for reads and writes. The least significant bit in each field is for physical channel A. The most significant bit is for physical channel C. Setting two physical channel write bits indicate that a write should be sent to both channels. If two bits are set in the write field, the write is sent to both channels. For mirroring, 2 bits are set in the read field. Reads are directed according to the hash function: SystemAddress[24] ^ [12] ^ [6].

The following table defines how the lockstep bit and CHM fields are set to steer reads and writes. In the table, h represents the hash function which evaluates to A or B. The Logical channel columns show the value of the read (e.g. R=000) and write (e.g. W=000) fields.

Table 16. Read and Write Steering

Configuration	Lockstep	Logical Channel		
		0	1	2
Independent Channel LCH0 to PCH0, LCH1 to PCH1, LCH2 to PCH2	0	W=001 R=001	W=010 R=010	W=100 R=100
Sparing from PCH0 to PCH2. LCH0 writes to PCH0 and PCH2, LCH0 reads to PCH0, LCH1 is mapped to PCH1	0	W=101 R=001	W=010 R=010	W=000 R=000
Mirror PCH0 and PCH1 LCH0 writes to PCH[0] and PCH[2] LCH0 reads to PCH[2h]	0	W=011 R=011	W=000 R=000	W=000 R=000
Lockstepped LCH0 to PCH[0] and PCH[1]	1	W=001 R=001	W=000 R=000	W=000 R=000

Mirroring consists of duplicating writes to two channels and alternating reads to a pair of channels. Thus, if any given logical channel has more than one bit for both reads and writes, they are capable of redundant operation. Mirroring is fully enabled when software changes the channel state of both channels to redundant, which allows uncorrectable channel errors to be signaled as correctable. Mirroring is only supported between channels A-B.

2.1.10.2 Second Level Address Translation

Second level address translation converts the channel address resulting from the first level decode into rank, bank row and column addresses.

The Integrated Memory Controller uses DIMM interleaving to balance loads. The channel address can be divided into 8 ranges and each range supports an interleave across 1,2 or 4 ranks. The MC_RIR_LIMIT_CH[2:0]_[7:0] and MC_RIR_WAYS_CH[2:0]_[31:0] registers define the ranges and rank interleaves.

2.1.10.2.1 Registers

Each channel has the following address mapping registers with the exception of the lockstep bit, which applies to all.

Table 17. Address Mapping Registers

Register / Bit	Description
MC_CHANNEL_MAPPER Register	Channel Mapping Register. Defines the mapping of logical channels decoded by the first level decode to the physical channels in the Integrated Memory Controller.
Lockstep bit	Global config bit for all channels. Affects duplication of reads and writes and address bit mapping.
MC_DOD_CH[2:0]_[1:0]	DIMM Organization Descriptors. Each physical channel has two DOD registers.
MC_RIR_Limit_CH[2:0]_[7:0]	DIMM Interleave Registers. Defines the range of system addresses that are directed to each virtual ranks on each physical channel. There are 8 range registers for each channel.
MC_RIR_Ways_CH[2:0]_[31:0]	There are four-way registers for each Rank Interleave Range, one for each rank that may appear in that range. Each register defines the offset from channel address to rank address and the combination of address bits used to select the rank.
RankID DIMM Rank Map	Defines which virtual ranks appear on the same DIMM.
Rank Mapping Register	Defines the correspondence of virtual ranks to physical CS.

2.1.11 Address Translations

2.1.11.1 Translating System Address to Channel Address

This operation could be considered the final step of Level 1 decode. It removes the "gaps" introduced by Level 1 decode to produce a contiguous channel address. This maintains the independence of Level 1 and Level 2 decode. Independence simplifies the memory mapping problem that must be solved by BIOS. Gap removal is implemented in the Integrated Memory Controller because Intel® QPI does not allow this function to be performed before remote memory requests are sent to other sockets. The address that appears on a Intel® QPI request must be the system address, not the channel address.

The maximum number of DIMMs on a channel is three DIMMs. However, maximum memory capacity is achieved with 2 QR DIMMs, with 12 16 GB DIMMs for a total of 192 GB for the platform with 2 Gb DRAM densities. Higher capacity can be achieved with 4 Gb DRAMs but 4 Gb DRAMs are not expected to be available for Picket Post launch.

Unless the channel appears above other channels in the first level decode, the first address to access the channel will not be 0. As the MMIO gap can be considered a degenerate 0-way interleave, memory mapped above 4 GB must subtract that gap. If the channel is interleaved with other channels, the addresses it receives may not be contiguous.

For example, the set of system addresses that access a 256 MB range of channel addresses on a given channel may be the even addresses between 1.5 GB and 2 GB. For any given level 1 interleave, there will be a series of coarse gaps introduced by lower interleaves and fine gaps introduced by the interleave itself. To keep level 1 decode independent of Level 2 decode, the 1.5GB offset must be subtracted and A[6] must not be mapped to Channel address bits. In general, it is not sufficient to simply omit the system address bits not mapped to Channel Address bits.

Level 1 decode performs socket/channel interleave at various granularities. To compensate, the Level 2 decode must be able to remove any three of the address bits in Table 18. More significant bits are shifted down. The register that defines the shift for each interleave has one bit for each address bit to be removed.

The logic that removes the way selection bits from the channel address for a given channel is independent of the location of the other channels. That is, the address bits to be removed to close gaps do not depend on whether the other channels of the interleave appear on the same node or different nodes.

The subtraction and shift operation required is different for each level 1 interleave, so the level 1 interleave number is passed to the level 2 decode. The second level decode has configuration registers that hold the offset and bits to be shifted for each level 1 interleave.

After the offset and shift are completed, the set of system addresses that address a channel are converted to a set of contiguous "Channel Addresses" from 0 to the number of bytes on the channel.

2.1.11.2 Translating Channel Address to Rank Address

This section describes how gaps are removed from the channel address to form a contiguous address space for each rank. Gaps from one to three cache lines in size result from interleaving across ranks on a channel. Gaps larger than 512 MB are a result from the interleaves below.

The Integrated Memory Controller uses DIMM interleaving to balance loads. Interleaving assigns low order address bits to variable DRAM bits. Since DIMMs on a channel may be of different sizes, there is not a one to one address mapping. The larger DIMMs must be split into blocks which are the same size as smaller DIMMs. Thus the Rank Address space is divided into ranges, each of which may be interleaved differently.

The channel address may be interleaved across the DIMMs on a channel up to 4 ways. The channel address is divided into 4 interleave ranges, each of which may be interleaved differently to support interleave across different sized DIMMs.

The smallest DIMM supported will be 512 MB, which defines the granularity of the interleave. Each channel maintains 4 range decoders. It specifies which ranks are interleaved and the offset to be added to the Rank Address address to compensate for any DRAM addresses used in lower interleaves.

2.1.11.3 Low Order Address Bit Mapping

The mapping of the least significant address bits are affected by:

- whether the request is a read or a write,
- whether channels are lockstepped or independent, and
- whether ECC is enabled.

In general, the mapping is assigned to:

- Return the critical chunk of read data first.
- Simplify the mapping of ECC code bits to DRAM data transfers. When ECC is enabled, Column bits are zeroed to ensure that the sequence of transfers from the DRAM to the ECC check is the same for every request.
- Simplify the transfer of write data to the DRAM.

In all cases RankAddress[5:3] is the same as SystemAddress[5:3] and defines the order in which the Integrated Memory Controller returns data.

Writes are not latency critical and are always written to DRAM starting with chunk 0. For independent channels, Column[2:0]=0. For lockstep, Column[1:0]=0 and Column[2] is mapped to a high order System Address bit.

For reads with independent channels and ECC disabled, the critical 8B chunk can be transferred to the Global Queue before the others. Therefore, Column[2:0] are mapped to SystemAddress[5:3].

For reads with independent channels and ECC enabled, the 32 B codeword must be accumulated before forwarding any data. While it reduces latency to get the critical 32 B codeword first, the sequencing of 8 B chunks within the codeword is not important. Column[1:0] are forced to zero so that every DRAM read returns the 4 8B chunks of each codeword in the same order. However, read returns are always transferred in critical word order, so the critical word ordering is performed after the ECC check.

Table 18. Critical Word First Sequence of Read Returns

Transfer	Most Significant 8B to GQ	Least Significant 8B to GQ
	SysAdrs[3]=1	SysAdrs[3]=0
First pair	SysAdrs[5], SysAdrs[4]	SysAdrs[5], SysAdrs[4]
Second pair	SysAdrs[5], !SysAdrs[4]	SysAdrs[5], !SysAdrs[4]
Third pair	!SysAdrs[5], SysAdrs[4]	!SysAdrs[5], SysAdrs[4]
Fourth pair	!SysAdrs[5], !SysAdrs[4]	!SysAdrs[5], !SysAdrs[4]

The mapping of System Address bits to read return transfers are the same for lockstep. That is, SystemAddress[3] selects the upper/lower portion of the transfer and SystemAddress[5:4] determine the critical word sequence of transfers.

However, since two channels provide the data in lockstep, the System Address bits are mapped to different DRAM column bits. SystemAddress[4] determines the channel on which the data is stored, but not necessarily the channel that returns the data. Both lockstepped channels have duplicate copies of the entire cache line. The even channels return the least significant 8B chunks and the odd channels return the most significant 8 B chunks. Thus half of the data returned by one memory channel is stored in the other channel's buffers. Data with SystemAddress[3]=1 is driven by odd physical channels, while data with SystemAddress[3]=0 is driven by even physical channels. The Integrated Memory Controller is constructed such that SystemAddress[3] determines which channel drives the data.

SystemAddress[5] is mapped to Column[1] so that the DRAMs return the critical 32 B codeword first. Column[0] is forced to zero so that every DRAM read returns the 28 B chunks of each half-codeword in the same order. The burst of four sequences Column[1:0]. Column[2] selects a different cache line and is mapped to higher order address bits.

The table below summarizes the lower order bit mapping. It applies to both Open Page and Closed Page address mappings.

Table 19. Lower System Address Bit Mapping Summary

Lockstepped?	ECC	Critical 8B first?	Physical Channel	Col[2]	Col[1]	Col[0]
Independent Burst Length of 8	Yes	No	N/A	Reads: SysAdrs[5]	0	0
				Writes: 0	0	0
	No	Yes	N/A	Reads: SysAdrs[5]	SysAdrs[4]	SysAdrs[3]
				Writes: 0	0	0
Lockstep Burst Length of 4	Yes	No.	System Address [4]	Assigned to a higher order SysAdrs bit.	Reads: SysAdrs[5]	0
					Writes: 0	0

2.1.11.4 Supported Configurations

The following table defines the DDR3 organizations that the Integrated Memory Controller supports.

Table 20. DDR Organizations Supported

2.11.2 1Gb

Configuration	256Mb x 4	128Mb x 8	64Mb x 16
# of Banks	8	8	8
Bank Address	BA0 - BA2	BA0 - BA2	BA0 - BA2
Auto precharge	A10/AP	A10/AP	A10/AP
BC switch on the fly	A12/BC#	A12/BC#	A12/BC#
Row Address	A0 - A13	A0 - A13	A0 - A12
Column Address	A0 - A9,A11	A0 - A9	A0 - A9
Page size [1]	1 KB	1 KB	2 KB

2.11.3 2Gb

Configuration	512Mb x 4	256Mb x 8	128Mb x 16
# of Banks	8	8	8
Bank Address	BA0 - BA2	BA0 - BA2	BA0 - BA2
Auto precharge	A10/AP	A10/AP	A10/AP
BC switch on the fly	A12/BC#	A12/BC#	A12/BC#
Row Address	A0 - A14	A0 - A14	A0 - A13
Column Address	A0 - A9,A11	A0 - A9	A0 - A9
Page size [1]	1 KB	1 KB	2 KB

2.1.12 DDR Protocol Support

The Integrated Memory Controller will use burst length of 4 for lockstep and 8 for independent channels. The Integrated Memory Controller will not vary burst length during operation.

2.1.13 Refresh

The Integrated Memory Controller will issue refreshes when no commands are pending to a rank. It will refresh all banks within a rank at the same time. It will not use per-bank refresh.

The refresh engine satisfies the following requirements:

Once DRAM initialization is complete, each DRAM gets at least N-8 and no more than N refreshes in any interval of length N * 7.8 us.

- Until the time when timing constraints and the previous requirements force refresh issue, no refreshes are issued to banks for which there are uncompleted reads.
- Support configurable tRFC up to 350 ns.

2.1.13.1 DRAM Driver Impedance Calibration

The ZQ mechanism is used to maintain the impedance of DRAM drivers constant despite temperature and very low frequency voltage variations.

The delay between ZQ commands and subsequent operations and the rate of ZQCS commands are defined in the ZQ Timings register. No other commands will be issued between bank closure and ZQCS.

Initial calibration will be performed by initiating a ZQCL command using the DDR3CMD register. The ZQCL command initiates a calibration sequence in the DRAM that updates driver and termination values.

Specific DRAM vendors will specify a rate to initiate ZQCS calibration commands. BIOS performs the initial calibration using ZQ.

In general, the Integrated Memory Controller will precharge all banks before issuing ZQCS command.

The Integrated Memory Controller will issue ZQCL on exit from self refresh as required by the DDR3 spec.

ZQCL for initialization can be issued prior to normal operation by writing the DDRCMD register.

2.1.14 Power Management

2.1.14.1 Interface to Uncore Power Manager

Each mode in which the Integrated Memory Controller will reduce performance for power savings will be at the command of the Uncore power manager. The Uncore power manager will be aware of collective CPU power states, platform power states. It will request entry into a particular mode and the Integrated Memory Controller will acknowledge entry. In some cases, entry into a power mode merely enables the possibility (e.g. DRAM Precharge Power Down Enabled) of entering a low power state; in other cases, such as Self Refresh, it indicates full entry into the low power state.

2.1.14.2 DRAM Power Down States

2.1.14.2.1 Power-Down

The Integrated Memory Controller will have a configurable activity timeout for each rank. Whenever no activities are present to a given rank for the configured interval, the Integrated Memory Controller will transition the rank to power-down mode.

The maximum duration for either active or precharge power-down is limited by the refresh requirements of the device tRFC(max). The minimum duration for power-down entry and exit is limited by the tCKE(min) parameter.

The Integrated Memory Controller will transition the DRAM to Power-down by de-asserting CKE and driving a NOP command. The Integrated Memory Controller will tristate all DDR interface pins except CKE (de-asserted) and ODT while in power down. The Integrated Memory Controller will transition the DRAM out of power-down state by synchronously asserting CKE and driving a NOP command.

2.1.14.2.2 Active Power Down

The DDR frequency and supply voltage cannot be changed while the DRAM is in active power-down.

CKE will be de-asserted CKE idle clocks after most recent command to a Rank. It takes two clocks to exit. The DRAM can only remain in Active Power Down for 9*tREFI (~700 µsec).

2.1.14.2.3 Precharge Power Down

If power-down occurs when all banks are idle, this mode is referred to as precharge power-down. If power-down occurs when there is a row active in any bank, this mode is referred to as active power-down.

A DRAM in power-down deactivates its input and output buffers, excluding CK, CK#, ODT, and CKE.

The Integrated Memory Controller will not actively seek precharge power down. If requests stop for the power down delay, the channel will de-assert CKE. If page closing is enabled, CKE will be asserted as needed to issue precharges to close pages when their idle timers expire.

If the closed page operation is selected, pages will be closed when CKE is asserted. If open-page operation is selected, pages are closed according to an adaptive policy. If there were many page hits recently, it is less likely that all pages will happen to be closed when the rank CKE timeout expires.

Table 21. DRAM Power Savings Exit Parameters

Parameter	Symbol	Exit Times (DCLKs = tCK)
Command to active power down	tRDPDEN	tRL + 5 This is the delay for reads, Applies it to all commands. RDA would be used in the auto-precharge case, which is slightly longer.
Active Power Down	tDPEX	
Active or Precharge Power down to any command	tXP	3 for 800, 4 for 1067 and 1333
ODT on (power down mode)	tAONPD	~2.5ns
ODT off (power down mode)	tAOFPD	~6.5ns
Self Refresh timing Reference?	tSXR	250
Self Refresh to commands	tXSDLL	512 MC will apply tXSRD delay before issuing any commands.
Min CKE hi or lo time	tCKE	Three clocks

2.1.14.3 Dynamic DRAM Interface Power Savings Features

The following table defines the IO power saving features. The 1 and 2 DCLK on/off times do not affect command issue and are non-critical to performance. The on/off time is quoted so that power savings can be accurately evaluated.

Table 22. Dynamic IO Power Savings Features (Sheet 1 of 2)

Power Savings Feature	On Condition	Time to Turn Off	Off Condition	Time to Turn On
Power down mixers and amps in the Address and Command Phase Interpolators.	DRAMs in self refresh	10 DCLK	S3 exit request	< 1 usec
Tristate Address and Command drivers	Deselect command is driven	1 DCLK	Any other command is driven	1 DCLK
Tristate Data drivers	No data to drive. Derived from write CAS.	0 DCLK	Driving data	0 DCLK

Table 22. Dynamic IO Power Savings Features (Sheet 2 of 2)

Power Savings Feature	On Condition	Time to Turn Off	Off Condition	Time to Turn On
Disable mixer and amp in phase interpolators for data drivers	No data to drive. Derived from write CAS.	1 DCLK	Driving data	1 DCLK
Power down Data Receivers, Strobe Phase Interpolators, Strobe amplifiers, Receive enable logic	No data to receive	1 DCLK	Receiving data	1 DCLK
Disable ODT	No data to receive	2 DCLK	Receiving data	2 DCLK
Clock disable	PCU warns clocks will be removed after DRAMs in self refresh.	10 DCLK	PCU indicates clocks are stable.	1 usec

2.1.14.4 Static DRAM Interface Power Savings Features

Disable bits in the padscan registers are available to disable categories of pins.

2.1.14.5 DRAM Temperature Throttling

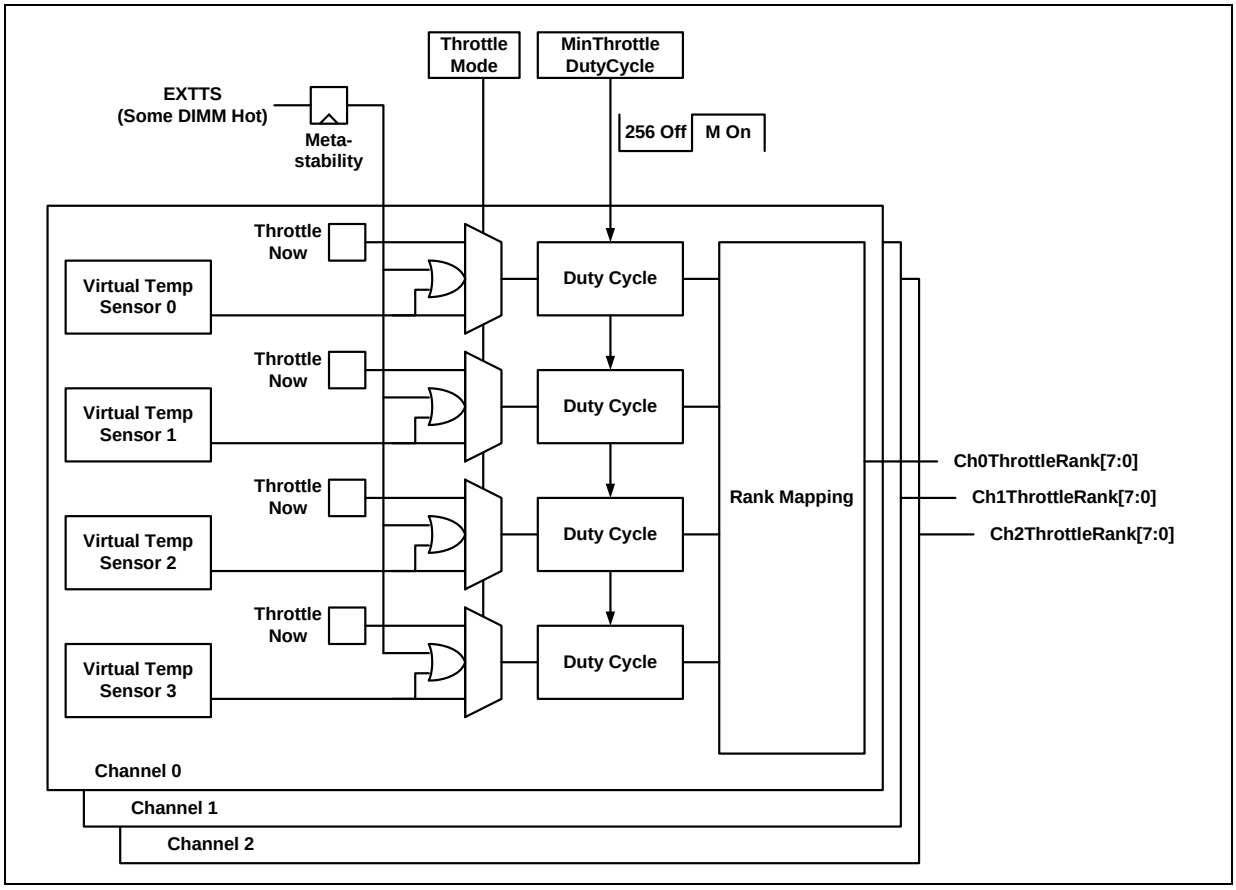
The Integrated Memory Controller currently supports open loop and closed loop throttling. The open loop throttling is compatible with the virtual temperature sensor approach implemented in desktop and mobile chipsets.

2.1.14.5.1 Throttler Logic Overview

There are 12 throttlers, four for each channel. The throttlers can be used in three modes, defined by what triggers throttling: a Virtual Temp Sensor (VTS), a ThrottleNow configuration bit, or a DDR_THERM# signal. Virtual Temperature Sensor is used where no DRAM temperature information is available. DDR_THERM# signal is used for basic closed loop throttling without any software assist. ThrottleNow mode allows software running on the CPU or thermal management agents to achieve maximum performance in varying operating conditions.

Each throttler has a VTS, ThrottleNow bit, and duty cycle generator. The DDR_THERM# signal is applied to all throttlers. The throttlers are mapped to ranks as described later.

Figure 11. Mapping Throttlers to Ranks



2.1.14.5.2 Virtual Temperature Sensor

The weights of the commands and the cooling coefficient can be dynamically modified by fan control firmware to update the virtual temperature according to current airspeed and ambient temperature. Care must be taken to avoid invalidating the virtual temperature. For example, when fan speeds are increased, the cooling coefficient should not be increased until the airspeed at the DIMM is sure to have reached the steady state value associated with the fan speed command. It is acceptable to reduce the cooling coefficient immediately on a fan speed decrease.

The thermal throttling logic implements this equation every DCLK:

T(t+1) = T(t) * (1 - c*2-36) + w*2-37

where:

T is virtual temperature

t is time

c is cooling coefficient

w is the weight of the command executed in cycle t

2.1.14.5.3 Virtual Temperature Counter

A counter will track the temperature above ambient of the hottest RAM in each rank. On each DRAM cycle, it will be incremented to reflect heat produced by DRAM activity and decremented to reflect cooling. This counter is saturating, it does not add when all ones, nor does it subtract when all zeros.

2.1.14.5.4 Per Command Energy

On each DCLK, the virtual temperature counter is increased to model the heat produced by the command issued in a previous DCLK. Eight-bit configurable values will be provided for Read, Write, Activate, and Idle with CKE on and off. The energy for an Activate-recharge cycle will be associated with the activate. Only common commands are represented. Other commands should use the idle value with the appropriate CKE state. Average refresh power should be included in the idle powers.

The per command energies are calculated from IDD values supplied by each DRAM vendor. The BIOS can determine the values on the basis of SPD information.

2.1.14.5.5 Cooling Coefficient

Over a series of 8 DCLKs, a portion of the temperature will be subtracted to model heat loss proportional to temperature. The portion is determined by a configurable cooling coefficient that represents the thermal resistance and capacitance of the DIMM.

The cooling coefficient is an 8-bit constant. In order to avoid multiplication of the current temperature and c in each cycle, the multiplication is done serially over 8 cycles. The following table describes how different amounts are subtracted on each of the 8 iterations. After the 8th iteration, the sequence repeats.

Firmware or BIOS can modulate the MC_COOLING_COEF dynamically to reflect better or worse system cooling capacity for memory. In case the fan controller is unable to update the cooling coefficient due to corner conditions or failure, the Integrated Memory Controller will load the SAFE_COOL_COEF value into the cooling coefficient if MC_COOLING_COEF is not updated in 0.5 seconds.

A thermal control agent can modulate the cooling coefficient to minimize the error between the virtual temperature and actual memory temperature. The agent must run a control loop at least twice a second to avoid application of failsafe values by the throttling logic. If it fails, throttling may occur due to conservative failsafe values and some performance might be lost. The agent should monitor DIMM temperature, Cycles Throttled and Virtual Temperature to minimize the difference between

DIMMtemp + DRAMdieToDIMMmargins - Ambient

and

VirtualTemp * (T64 - Ambient)/ThrottlePoint

2.1.14.5.6 Throttle Point

The throttle point is set by the ThrottleOffset parameter. When the virtual temperature exceeds the throttling threshold, throttling is triggered. As an artifact of closed loop throttling using DRAM die temperature sampling, the MSB of the virtual temperature is compared to 0 and VT[36:29] are compared to the ThrottleOffset parameter. Since Virtual Temperature cannot exceed the throttlepoint by very much before throttling is triggered, the effective range of Virtual Temperature is only 37, not 38 bits. It is recommended that Throttle Point be set to 255 for all usages.

2.1.14.5.7 Response to Throttling Trigger

When throttling is triggered, CKE will be de-asserted. DRAM issue will be suppressed to the hot rank except for refreshes. After 256 DCLKs, command issue will be allowed according to the MinThrottleDutyCycle field. After that, command issue will be permitted if temperature is above threshold.

This should be the case as the worst case cooling after 256 DCLKs of precharge power down should be sufficient to allow many commands. In the steady state under TDP load, 256 DCLKs of inactivity will be followed by however many DCLKs of high activity can be thermally sustained, and then another 256 DCLKs of inactivity and the sequence repeats.

Throttling is not re-triggerable, that is, multiple throttling triggers during the 256 DCLK interval will have no effect.

To support Lockstep mode, ranks on each channel will be throttled if throttling is required for the corresponding rank on the other channel.

2.1.14.5.8 DDR_THERM# Pin Response

A DDR_THERM# pin on the socket enables the throttle response.

Table 23. DDR_THERM# Responses

Register	Parameter	Bits	One Per	Description
MC_DDR_THERM_COMMAND	THROTTLE	1	Socket. (appears in each of the three channels)	While DDR_THERM# is high, Duty Cycle throttling will be imposed on all channels. The platform should ensure DDR_THERM# is high when any DIMM is over T64.

The interrupt capability is intended to allow modulation of throttling parameters by software that cannot perform a control loop several times a second. There is no PCI device to associate with the interrupts generated. The DDR_THERM# status registers have to be examined in response to these interrupts to determine whether the memory controller has triggered the interrupt.

2.1.14.6 Closed Loop Thermal Throttling (CLTT)

Basic closed loop thermal throttling can be achieved using the DDR_THERM# signal. Temperature sensors in Tcrit only mode will be placed on or near the DIMMs attached to the socket. The EVENT# pin of all DIMMs will be wired-OR to produce a DDR_THERM# signal to the Memory Controller. The temperature sensors will be configured to trip at the lowest temperature at which the associated DRAMs might exceed T64 or T32 (double or single refresh DRAM temperature spec) case temperature. BIOS or firmware will calculate and configure DIMM Event# temperature limit (Tcrit) based on DIMM thermal/power property, system cooling capacity, and DRAM/register temperature specifications. These temperature sensors are generally updated a minimum of eight times per second, but the more often they update, the smoother the throttling will be.

When one of the temperature sensors trips, the memory controller will throttle all ranks according to the duty cycle configured in the MinThrottleDutyCycle fields. This field should be set to allow a percentage of full bandwidth supported by the minimum fan speed at worst case operating conditions. Command issue will be blocked for the Off portion of the duty cycle whether or not commands have been issued during the On portion. This will generally result in over-throttling until the temperature sensors re-evaluate. This will result in choppy throttling. For a temp sensor update interval of 1/8 second. There will be 125 ms of very low bandwidth followed by n*125 ms of

unrestrained BW such that the average over many seconds is that which is thermally supported. The choppiness will be lessened by configuring MinThottleDutyCycle no lower than that required by the specific DIMMs plugged into each system.

2.1.14.7 Advanced Throttling Options

There are two closed loop throttling considerations which can be addressed by a thermal control agent (Management Hardware via PECI or SW periodically running on a core).

Option 1 is that all channels are throttled when any DIMM is hot. The CPU prefetchers adapt to longer latencies caused by throttling one rank by reducing the bandwidth to that rank. So it is better to throttle only the hot ranks.

Option 2 is the closed "transient margin": The DIMM temperature sensor lags the DRAM die temperature, so throttling must be triggered at a lower temperature than T64. This results in a loss of bandwidth for a given cooling capability.

A thermal control agent which monitors the DIMM temperature serially via SPD can collect higher granularity (as opposed to binary hot/cold via DDR_THERM#) information. The thermal control agent can set the ThrottleNow bits for ranks that are nearing max temperature. Per rank throttling only limits bandwidth to hot DIMMs.

Both concerns can be addressed by a thermal control agent modulating the virtual temperature sensor as described in Section 2.1.14.5.5, “Cooling Coefficient” . When this is done, closed loop throttling can be enabled at a higher DIMM temperature that does not include the transient margin can be feedback should not be needed, but can be added for safety.

2.1.14.8 2X Refresh

Some DRAMs can be operated above 85 degrees if refresh rate is doubled. The DDR3 DRAM spec refers to this capability as Extended Temperature Range (ETR). Some DRAMs have the capability to self refresh at a rate appropriate for their temperature. The DDR3 spec defines this as the Automatic Self Refresh (ASR) feature. When all DRAM on a channel have ASR enabled, the MC_CHANNEL_X_REFRESH_THROTTLE_SUPPORT.ASR_PRESENT bit should be set. Some platforms may support Extended Temperature Range operation, others may not.

The following recommendations are predicated on the assumption that BIOS set the DRAM ASR enable for any DIMM with SPD that indicates ASR support.

Table 24. Refresh for Different DRAM Types

Type	Open Loop	Closed Loop
	No indication of ETR	ETR is indicated by DDR_THERM# or MC_CLOSED_LOOP.REF_2X_NOW. The temperature indication can be directly from a thermal sensor, or via Baseboard Management Controller (BMC), or software.

Table 24. Refresh for Different DRAM Types

Type	Open Loop	Closed Loop
Platform or some DRAM on the socket does not support ETR. DRAM temperatures are limited to 85 degrees.	Refresh interval is always tREFI. There is no 2x refresh response. Self refresh entry is not delayed. Refresh rate is always 1x.	
All DRAM on the socket and the platform support ERT. Throttling does not limit DRAM temperature below 88 degrees.	BIOS will halve tREFI and double the parameters controlling maintenance operations. The memory controller and the DRAM are configured to refresh at 2x rate. There is no dynamic 2x refresh response.	When DDR_THERM# pin defined to respond with 2x refresh is inactive, refresh interval returns to tREFI.
	Non-ASR DRAM that supports Extended Temperature will be configured to self refresh at 2x rate. Non-ASR DRAM that does not support Extended Temperature is configured to self-refresh at 1x rate. ASR DRAM will be configured to adjust its self-refresh rate according to temperature.	

Integrated Memory Controller can double the refresh rate dynamically in two cases:

- When MC_CLOSED_LOOP.REF_2X_NOW configuration bit is set.
- When DDR_THERM# pin is asserted and bit[2] of the MC_DDR_THERM_COMMANDX register is set.

Memory controller delays refresh when doubling 2x refresh rate and ASR_PRESENT bit is set.

Memory controller supports a register based dynamic 2X Refresh via the REF_2X_NOW bit in the MC_CLOSED_LOOP register. See the MC_CLOSED_LOOP register in Table 27 for more details. In a system ensuring the DRAM never exceeds T64, it is conceivable the DIMM temperature sensor be used for this purpose. However, most platforms reduce fan speed during idle periods, and fan speed cannot be increased fast enough to keep up with DRAM temperature. Therefore, DIMM temperature sensors are probably used for T64. A temperature sensor near the DIMMs can be used to control 2X refresh.

Alternatively, code running on a processor or an external agent via PECI can set the MC_CLOSED_LOOP.REF_2X_NOW configuration bit on a channel basis. An agent which monitors the DIMM temperature serially via SPD can track this temperature. The agent must account for its worst case update interval and max rate of DRAM temperature increase to make sure the DRAM does not exceed T32 between updates. There is no failsafe logic to apply 2X Refresh if updates are not received often enough.

If the agent cannot reliably monitor this information, the refresh rate should be statically doubled by setting refresh parameters for extended temperature DRAM.

2.1.14.9 Demand Observation

In order to smooth fan speed transitions, the fan control agent needs to know how memory activity demanded by the current application is related to the throttling point.

By observing the trend of Virtual Temperature relative to throttling point, the fan controller can determine the trend of demand before the throttling point is exceeded. However, once the throttling point has been exceeded, the Virtual Temperature will remain at the throttling point and only provides the information that demand exceeds the throttling limit.

If the fan controller could determine the throttling duty cycle, it could determine how much demand exceeds the throttling limit. Therefore, the CYCLES_THROTTLED field gives an indication of throttling duty cycle. A 32-bit counter accumulates the number of DCLKs each rank has been throttled. Each time the ZQ interval completes, the 16 most significant bits of the counter are loaded into the status register and the counter is cleared. The register thus holds the number of cycles throttled in the last 128 ms (give or take a factor of 2; the thermal sensor sample rate is configurable).

2.1.14.10 Rank Sharing

Throttling logic is shared when more than four ranks are present as described in the following tables.

With 1 or 2 Single or Dual rank DIMMs on a channel, there are no more than four ranks present. Each throttler is associated with a single rank. The logical ranks are not consecutive due to the motherboard routing required to support three DIMMs on a channel.

Table 25. 1 or 2 Single / Dual Rank Throttling

Throttler	Command Accumulation	Which Logical Rank Throttled
0	accumulates the command power of rank 0	0
1	accumulates the command power of rank 4	4
2	accumulates the command power of rank 1	1
3	accumulates the command power of rank 5	5

When more than four ranks are present, sharing is required. Adjacent logical ranks are shared as they are on the same DIMM. Ranks on the same DIMM share the same DIMM planar thermal mass. CKE sharing is across DIMMs. Since it is best to de-assert CKE when throttling and CKE shared across DIMMs cannot be de-asserted until commands stop going to ranks on both DIMMs, it is simpler to throttle all ranks at the same time so that CKE may be de-asserted. Although CKE may be de-asserted on some ranks but not others when there is no throttling, this lower command energy will not be captured.

When only one quad rank is present, CKE is shared due to DIMM connector limitations. In this case, logical ranks 0, 1, 2, and 3 are present, but only Throttler 0 and 1 are used.

Table 26. 1 or 2 Quad Rank or 3 Single / Dual Rank Throttling

Throttler	Command Accumulation	Which Logical Rank Throttled
0	If a read/write/act is issued to rank 0 or 1, accumulate the read or write power. else, if any rank has CKE asserted, accumulate CKE asserted idle power else accumulate CKE de-asserted idle power	all
1	If a read/write/act is issued to rank 2 or 3, accumulate the read or write power. else, if any rank has CKE asserted, accumulate CKE asserted idle power else accumulate CKE de-asserted idle power	all
2	If a read/write/act is issued to rank 4 or 5, accumulate the read or write power. else, if any rank has CKE asserted, eccumulate CKE asserted idle power else accumulate CKE de-asserted idle power	all
3	If a read/write/act is issued to rank 6 or 7, accumulate the read or write power. else, if any rank has CKE asserted, accumulate CKE asserted idle power else accumulate CKE de-asserted idle power	All

2.1.14.11 Registers

Table 27 describes the parameters for this function. The size and association of each parameter is described. If the parameter is per channel, there are three per socket. If the parameter is per rank, there are 12 per socket. There is a limit of four throttlers per rank as described in the preceding section.

Table 27. Thermal Throttling Control Fields (Sheet 1 of 2)

Register	Dynamically Validated	Parameter	Bits	One per	Description
MC_THERMAL_CONTROL		THROTTLE_MODE	2	Channel	Defines the source of throttling information to be DDR_THERM# signal, virtual temperature sensor, or Throttle_Now configuration bit. Throttling can also be disabled with this field.
MC_THERMAL_CONTROL		THROTTLE_EN	1	Channel	DRAM commands will be throttled.
MC_CLOSED_LOOP	YES	THROTTLE_NOW	4	Throttler	Throttle according to Min Throttle Duty Cycle. This parameter may be modified during operation.
MC_CLOSED_LOOP	YES	MIN_THROTTLE_ DUTY_CYC	10	Channel	The minimum number of DCLKs of operation allowed after throttling is 4x this parameter. In order to provide actual command opportunities, the number of clocks between CKE de-assertion and first command should be considered. This parameter may be modified during operation.
MC_THERMAL_PARAMS_B		SAFE_DUTY_CYC	10	Channel	This value replaces Min Throttle Duty Cycle if it has not been updated for 4 sample periods.
MC_THERMAL_PARAMS_B		SAFE_COOLING_ COEF	8	Channel	Any rank that has received eight temperature samples since the last cooling coefficient update will load this value.
MC_THERMAL_CONTROL		APPLY_SAFE	1	Channel	If set, the Safe cooling coefficient will be applied after eight temperature sample intervals. Parameter appears on B-x stepping silicon.
MC_CHANNEL_X_ZQ_TIMING		ZQ_INTERVAL	21	Channel	This field defines the sample interval, formerly used for on-die temperature sensor samples, but currently only used to apply failsafe values when it has been too long between updates. Nominally set to 128 ms.
MC_THERMAL_DEFEATURE		THERM_REG_LO CK	1	Channel	Prevents further modification of all parameters in this table. This should not be set if parameters are to be modified during operation. On A-x stepping, unless set, safe cooling coefficient and DutyCycle will be applied when the associated register is not updated for 4 sample intervals. In the long run, safe cooling coefficient will be enabled by the APPLY_SAFE bit.
MC_THERMAL_PARAMS_A		RDCMD_ ENERGY	8	Channel	Energy of a read including data transfer
MC_THERMAL_PARAMS_A		WRCMD_ENERGY	8	Channel	Energy of a write including data transfer
MC_THERMAL_PARAMS_A		CKE_DEASSERT_ ENERGY	8	Channel	Energy of having CKE de-asserted when no command is issued.
MC_THERMAL_PARAMS_A		CKE_ASSERT_EN ERGY	8	Channel	Energy of having CKE asserted when no command is issued
MC_THERMAL_PARAMS_B		ACTCMD_ENERG Y	8	Channel	Energy of an Activate/Precharge Cycle

Table 27. Thermal Throttling Control Fields (Sheet 2 of 2)

Register	Dynamically Validated	Parameter	Bits	One per	Description
MC_THROTTLE_OFFSET		RANK	8	Throttler	Compared against bits [36:29] of virtual temperature to determine the throttle point. Recommended value is 255.
MC_COOLING_COEF	YES	RANK	8	Throttler	Heat removed from DRAM in 8 DCLKs. This should be scaled relative to the per command weights and the initial value of the throttling threshold. This includes idle command and refresh energies. If 2X refresh is supported, the worst case of 2X refresh must be assumed. This parameter may be modified during operation.
MC_CLOSED_LOOP	YES	REF_2X_NOW	1	Throttler	When set, refresh rate is doubled. This parameter may be modified during operation.

Table 28. Thermal Throttling Status Fields

Register	Parameter	Bits	One Per	Description
MC_DDR_THERM_STATUS	STATE	1	Socket. (appears in each of the 3 channels)	DDR_THERM# rising edge was detected since this bit was last reset. Parameter appears on B-x stepping silicon.
				DDR_THERM# falling edge was detected since this bit was last reset.
				Current value of DDR_THERM# pin
MC_THERMAL_STATUS	RANK_TEMP	4	Channel	Bit specifies whether the rank is above throttling threshold.
MC_THERMAL_STATUS	CYCLES_THROTTLED	16	Channel	The number of throttle cycles triggered in all ranks since last temperature sample
MC_RANK_VIRTUAL_TEMP	RANK	8	Throttler	Most significant bits of Virtual Temperature of the selected rank. The difference between the Virtual Temperature and the Sensor temperature can be used to determine how fast fan speed should be increased.

2.2 Platform Environment Control Interface (PECI)

The Platform Environment Control Interface (PECI) uses a single wire for self-clocking and data transfer. The bus requires no additional control lines. The physical layer is a self-clocked one-wire bus that begins each bit with a driven, rising edge from an idle level near zero volts. The duration of the signal driven high depends on whether the bit value is a logic ‘0’ or logic ‘1’. PECI also includes variable data transfer rate established with every message. In this way, it is flexible even though underlying logic is simple.

The interface design was optimized for interfacing to Intel processor and chipset components in both single processor and multiple processor environments. The single wire interface provides low board routing overhead for the multiple load connections in the congested routing area near the processor and chipset components. Bus speed, error checking, and low protocol overhead provides adequate link bandwidth and reliability to transfer critical device operating conditions and configuration information.

The PECI bus offers:

- A wide speed range from 2 Kbps to 2 Mbps.

- CRC check byte used to efficiently and atomically confirm accurate data delivery.
- Synchronization at the beginning of every message minimizes device timing accuracy requirements.

Generic PECI specification details are out of the scope of this document and instead can be found in *RS - Platform Environment Control Interface (PECI) Specification*, Revision 2.0. What follows is a processor-specific PECI client definition, and is largely an addendum to the PECI Network Layer and Design Recommendations sections for the PECI 2.0 Specification.

Note: The PECI commands described in this document apply to the Intel® Xeon® processor C5500/C3500 series only. See Table 29 for the list of PECI commands supported by the Intel® Xeon® processor C5500/C3500 series PECI client.

Table 29. Summary of Processor-Specific PECI Commands

Command	Supported on Intel® Xeon® Processor C5500/C3500 Series CPU
Ping()	Yes
GetDIB()	Yes
GetTemp()	Yes
PCIConfigRd()	Yes
PCIConfigWr()	Yes
MbxSend() [1]	Yes
MbxGet() [1]	Yes

Note:
1. See Table 34 for a summary of mailbox commands supported by the Intel® Xeon® processor C5500/C3500 series CPU.

2.2.1 PECI Client Capabilities

The Intel® Xeon® processor C5500/C3500 series PECI client is designed to support the following sideband functions:

- Processor and DRAM thermal management.
- Platform manageability functions including thermal, power and electrical error monitoring.
- Processor interface tuning and diagnostics capabilities (Intel® Interconnect BIST [Intel® IBIST]).

2.2.1.1 Thermal Management

Processor fan speed control is managed by comparing PECI thermal readings against the processor-specific fan speed control reference point, or $T_{CONTROL}$. Both $T_{CONTROL}$ and PECI thermal readings are accessible via the processor PECI client. These variables are referenced to a common temperature, the TCC activation point, and are both defined as negative offsets from that reference. Algorithms for fan speed management using PECI thermal readings and the $T_{CONTROL}$ reference are documented in Section 2.2.2.6.

PECI-based access to DRAM thermal readings and throttling control coefficients provide a means for Board Management Controllers (BMCs) or other platform management devices to feed hints into on-die memory controller throttling algorithms. These control coefficients are accessible using PCI configuration space writes via PECI reference are documented in Section 2.2.2.5.

2.2.1.2 Platform Manageability

PECI allows full read access to error and status monitoring registers within the processor's PCI configuration space. It also provides insight into thermal monitoring functions such as TCC activation timers and thermal error logs.

2.2.1.3 Processor Interface Tuning and Diagnostics

The processor Intel® IBIST allows for in-field diagnostic capabilities in Intel® QuickPath Interconnect and memory controller interfaces. PECI provides a port to execute these diagnostics via its PCI Configuration read and write capabilities.

2.2.2 Client Command Suite

2.2.2.1 Ping()

Ping() is a required message for all PECI devices. This message is used to enumerate devices or determine if a device has been removed, been powered-off, etc. A Ping() sent to a device address always returns a non-zero Write FCS if the device at the targeted address is able to respond.

2.2.2.1.1 Command Format

The Ping() format is as follows:

Write Length: 0

Read Length: 0

Figure 12. Ping()

Byte #	0	1	2	3
Byte Definition	Client Address	Write Length 0x00	Read Length 0x00	FCS

An example Ping() command to PECI device address 0x30 is shown below.

Figure 13. Ping() Example

Byte #	0	1	2	3
Byte Definition	0x30	0x00	0x00	0xe1

2.2.2.2 GetDIB()

The processor PECI client implementation of GetDIB() includes an 8-byte response and provides information regarding client revision number and the number of supported domains. All processor PECI clients support the GetDIB() command.

2.2.2.2.1 Command Format

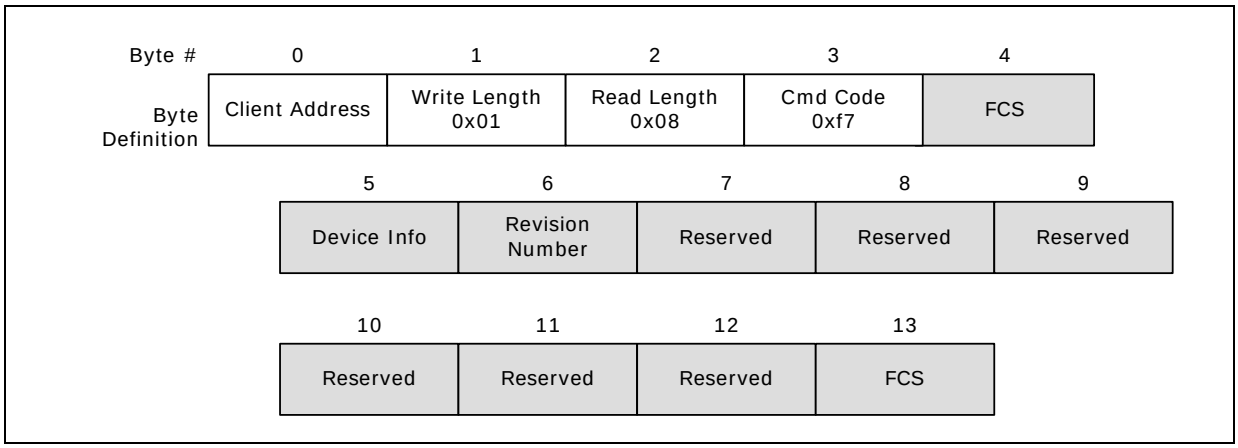
The GetDIB() format is as follows:

Write Length: 1

Read Length: 8

Command: 0xf7

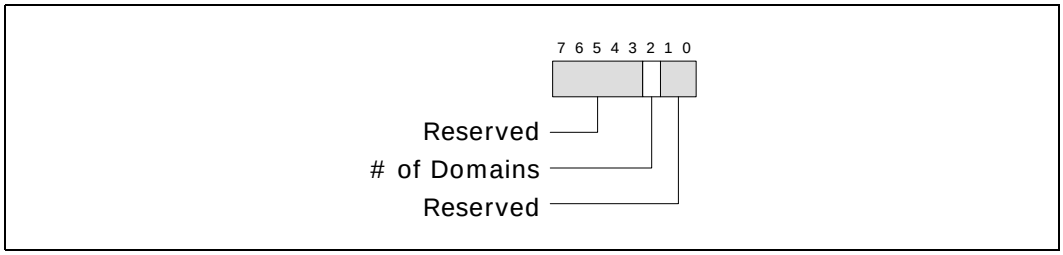
Figure 14. GetDIB()



2.2.2.2.2 Device Info

The Device Info byte gives details regarding the PECI client configuration. At a minimum, all clients supporting GetDIB will return the number of domains inside the package via this field. With any client, at least one domain (Domain 0) must exist. Therefore, the Number of Domains reported is defined as the number of domains in addition to Domain 0. For example, if the number 0b1 is returned, that would indicate that the PECI client supports two domains.

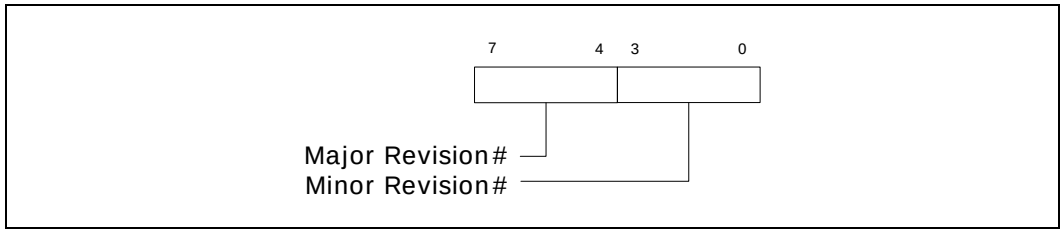
Figure 15. Device Info Field Definition



2.2.2.2.3 Revision Number

All clients that support the GetDIB command also support Revision Number reporting. The revision number may be used by a host or originator to manage different command suites or response codes from the client. Revision Number is always reported in the second byte of the GetDIB() response. The Revision Number always maps to the revision number of the supported PECI Specification.

Figure 16. Revision Number Definition



For a client that is designed to meet the Revision 2.0 *RS - Platform Environment Control Interface (PECI) Specification*, the Revision Number it returns will be '0010 0000b'.

2.2.2.3 GetTemp()

The GetTemp() command is used to retrieve the temperature from a target PECI address. The temperature is used by the external thermal management system to regulate the temperature on the die. The data is returned as a negative value representing the number of degrees centigrade below the Thermal Control Circuit Activation temperature of the PECI device. A value of zero represents the temperature at which the Thermal Control Circuit activates. The actual value that the thermal management system uses as a control set point (Tcontrol) is also defined as a negative number below the Thermal Control Circuit Activation temperature. TCONTROL may be extracted from the processor by issuing a PECI Mailbox MbxGet() (see Section 2.2.2.8), or using a RDMSR instruction.

See Section 2.2.6 for details regarding temperature data formatting.

2.2.2.3.1 Command Format

The GetTemp() format is as follows:

Write Length: 1

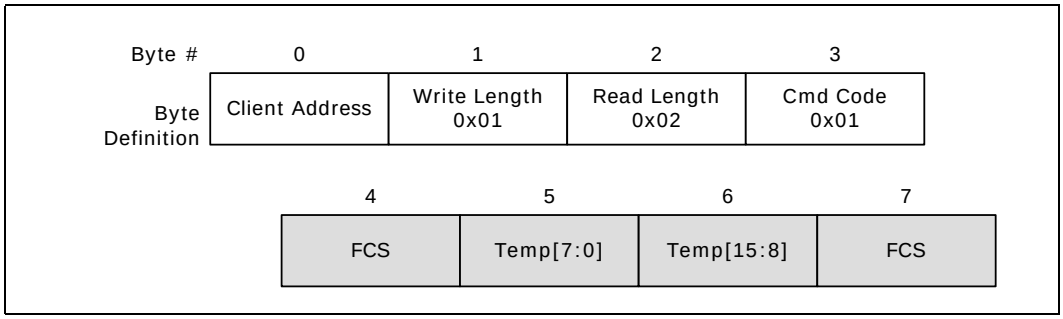
Read Length: 2

Command: 0x01

Multi-Domain Support: Yes (see Table 41)

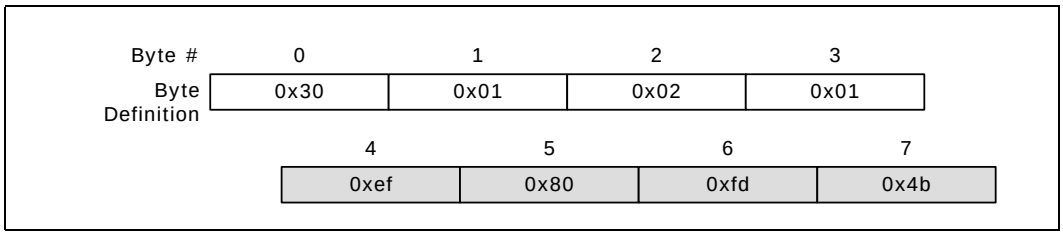
Description: Returns the current temperature for addressed processor PECI client.

Figure 17. GetTemp()



Example bus transaction for a thermal sensor device located at address 0x30 returning a value of negative 10° C:

Figure 18. GetTemp() Example



2.2.2.3.2 Supported Responses

The typical client response is a passing FCS and good thermal data. Under some conditions, the client's response will indicate a failure.

Table 30. GetTemp() Response Definition

Response	Meaning
General Sensor Error (GSE)	Thermal scan did not complete in time. Retry is appropriate.
0x0000	Processor is running at its maximum temperature or is currently being reset.
All other data	Valid temperature reading, reported as a negative offset from the TCC activation temperature.

2.2.2.4 PCIConfigRd()

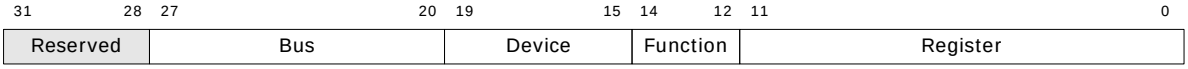
The PCIConfigRd() command gives sideband read access to the entire PCI configuration space maintained in the processor, but the PECI commands do not suport the IIO PCI space. This capability does not include support for route-through to downstream devices or sibling processors. Intel® Xeon® processor C5500/C3500 series PECI originators may conduct a device/function/register enumeration sweep of this space by issuing reads in the same manner that BIOS would. A response of all 1's indicates that the device/function/register is unimplemented.

PCI configuration addresses are constructed as shown in the following diagram. Under normal in-band procedures, the Bus number (including any reserved bits) would be used to direct a read or write to the proper device. Since there is a one-to-one mapping between any given client address and the bus number, any request made with a bad Bus number is ignored and the client will respond with a 'pass' completion code but all 0's in the data. The bus number for the processor PCI registers will be programmed to

255 for legacy processor and 254 for non-legacy processor. The client will return all 1's in the data response and 'pass' for the completion code for all of the following conditions:

- Unimplemented Device
- Unimplemented Function
- Unimplemented Register

Figure 19. PCI Configuration Address



PCI configuration reads may be issued in byte, word, or dword granularities.

2.2.2.4.1 Command Format

The PCIConfigRd() format is as follows:

Write Length: 5

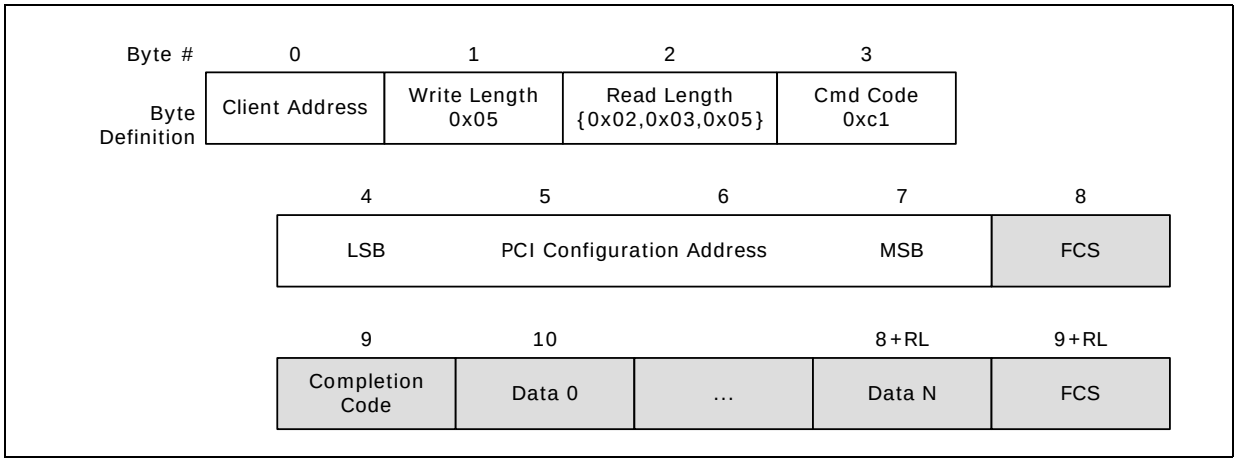
Read Length: 2 (byte data), 3 (word data), 5 (dword data)

Command: 0xc1

Multi-Domain Support: Yes (see Table 41)

Description: Returns the data maintained in the PCI configuration space at the PCI configuration address sent. The Read Length dictates the desired data return size. This command supports byte, word, and dword responses as well as a completion code. All command responses are prepended with a completion code that includes additional pass/fail status information. See Section 2.2.4.2 for details regarding completion codes.

Figure 20. PCIConfigRd()



The 4-byte PCI configuration address defined above is sent in standard PECI ordering with LSB first and MSB last.

2.2.2.4.2 Supported Responses

The typical client response is a passing FCS, a passing Completion Code (CC) and valid Data. Under some conditions, the client's response will indicate a failure.

Table 31. PCIConfigRd() Response Definition

Response	Meaning
Abort FCS	Illegal command formatting (mismatched RL/WL/Command Code)
CC: 0x40	Command passed, data is valid
CC: 0x80	Error causing a response timeout. Either due to a rare, internal timing condition or a processor RESET or processor S1 state. Retry is appropriate outside of the RESET or S1 states.

2.2.2.5 PCIConfigWr()

The PCIConfigWr() command gives sideband write access to the PCI configuration space maintained in the processor. The exact listing of supported devices, functions is defined below in Table 32. PECI originators may conduct a device/function/register enumeration sweep of this space by issuing reads in the same manner that BIOS would.

Table 32. PCIConfigWr() Device/Function Support

Writable		Description
Device	Function	
2	1	Intel® QuickPath Interconnect Link 0 Intel® IBIST
2	5	Intel® QuickPath Interconnect Link 1 Intel® IBIST
3	4	Memory Controller Intel® IBIST[1]
4	3	Memory Controller Channel 0 Thermal Control / Status
5	3	Memory Controller Channel 1 Thermal Control / Status
6	3	Memory Controller Channel 2 Thermal Control / Status

1. Currently not available for access through the PECI PCIConfigWr() command.

PCI configuration addresses are constructed as shown in Figure 19, and this command is subject to the same address configuration rules as defined in Section 2.2.2.4. PCI configuration reads may be issued in byte, word, or dword granularities.

Because a PCIConfigWr() results in an update to potentially critical registers inside the processor, it includes an Assured Write FCS (AW FCS) byte as part of the write data payload. See the *RS - Platform Environment Control Interface (PECI) Specification*, Revision 2.0 for a definition of the AW FCS protocol. In the event that the AW FCS mismatches with the client-calculated FCS, the client will abort the write and will always respond with a bad Write FCS.

2.2.2.5.1 Command Format

The PCIConfigWr() format is as follows:

Write Length: 7 (byte), 8 (word), 10 (dword)

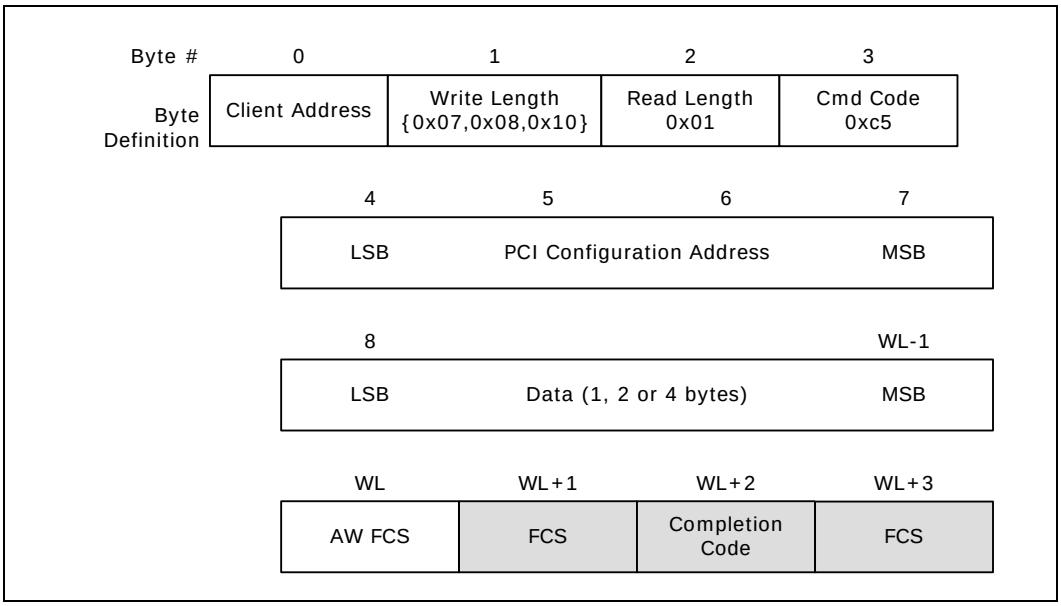
Read Length: 1

Command: 0xc5

Multi-Domain Support: Yes (see Table 41)

Description: Writes the data sent to the requested register address. Write Length dictates the desired write granularity. The command always returns a completion code indicating the pass/fail status information. Write commands issued to illegal Bus Numbers, or unimplemented Device / Function / Register addresses are ignored but return a passing completion code. See Section 2.2.4.2 for details regarding completion codes.

Figure 21. PCIConfigWr()



The 4-byte PCI configuration address and data defined above are sent in standard PECI ordering with LSB first and MSB last.

2.2.2.5.2 Supported Responses

The typical client response is a passing FCS, a passing Completion Code and valid Data. Under some conditions, the client's response will indicate a failure.

Table 33. PCIConfigWr() Response Definition

Response	Meaning
Bad FCS	Electrical error or AW FCS failure
Abort FCS	Illegal command formatting (mismatched RL/WL/Command Code)
CC: 0x40	Command passed, data is valid
CC: 0x80	Error causing a response timeout. Either due to a rare, internal timing condition or a processor RESET condition or processor S1 state. Retry is appropriate outside of the RESET or S1 states.

2.2.2.6 Mailbox

The PECI mailbox ("Mbx") is a generic interface to access a wide variety of internal processor states. A Mailbox request consists of sending a 1-byte request type and 4-byte data to the processor, followed by a 4-byte read of the response data. The following sections describe the Mailbox capabilities as well as the usage semantics for the MbxSend and MbxGet commands which are used to send and receive data.

2.2.2.6.1 Capabilities

Table 34. Mailbox Command Summary

Command Name	Request Type Code (byte)	MbxSend Data (dword)	MbxGet Data (dword)	Description
Ping	0x00	0x00	0x00	Verify the operability / existence of the mailbox.
Thermal Status Read/Clear	0x01	Log bit clear mask	Thermal Status Register	Read the thermal status register and optionally clear any log bits. The thermal status has status and log bits indicating the state of processor TCC activation, external PROCHOT# assertion, and Critical Temperature threshold crossings.
Counter Snapshot	0x03	0x00	0x00	Snapshots all PECI-based counters
Counter Clear	0x04	0x00	0x00	Concurrently clear and restart all counters.
Counter Read	0x05	Counter Number	Counter Data	Returns the counter number requested. 0: Total reference time 1: Total TCC Activation time counter
Icc-TDC Read	0x06	0x00	Icc-TDC	Returns the specified Icc-TDC of this part, in Amps.
Thermal Config Data Read	0x07	0x00	Thermal config data	Reads the thermal averaging constant.
Thermal Config Data Write	0x08	Thermal Config Data	0x00	Writes the thermal averaging constant.
Tcontrol Read	0x09	0x00	Tcontrol	Reads the fan speed control reference temperature, Tcontrol, in PECI temperature format.
Machine Check Read	0x0A	Bank Number / Index	Register Data	Read CPU Machine Check Banks.
T-state Throttling Control Read	0xB	0x00	ACPI T-state Control Word	Reads the PECI ACPI T-state throttling control word.
T-state Throttling Control Write	0xC	ACPI T-state Control Word	0x00	Writes the PECI ACPI T-state throttling control word.

Any MbxSend request with a request type not defined in Table 34 will result in a failing completion code.

More detailed command definitions follow.

2.2.2.6.2 Ping

The Mailbox interface may be checked by issuing a Mailbox ‘Ping’ command. If the command returns a passing completion code, it is functional. Under normal operating conditions, the Mailbox Ping command should always pass.

2.2.2.6.3 Thermal Status Read / Clear

The Thermal Status Read provides information on package level thermal status. Data includes:

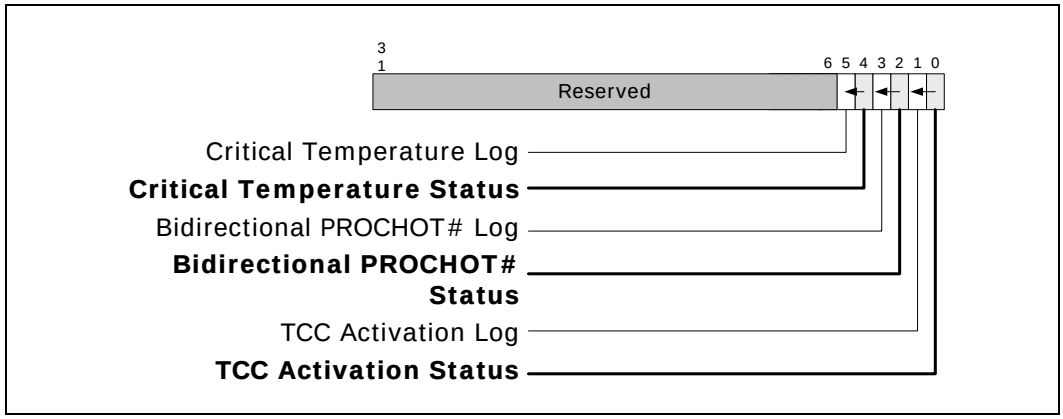
- The status of TCC activation
- Bidirectional PROCHOT# assertion
- Critical Temperature

These status bits are a subset of the bits defined in the IA32_THERM_STATUS MSR on the processor, and more details on the meaning of these bits may be found in the *Intel® 64 and IA-32 Architectures Software Developer's Manual*, Vol. 3B.

Both status and sticky log bits are managed in this status word. All sticky log bits are set upon a rising edge of the associated status bit, and the log bits are cleared only by Thermal Status reads or a processor reset. A read of the Thermal Status Word always includes a log bit clear mask that allows the host to clear any or all log bits that it is interested in tracking.

A bit set to 0b0 in the log bit clear mask will result in clearing the associated log bit. If a mask bit is set to 0b0 and that bit is not a legal mask, a failing completion code will be returned. A bit set to 0b1 is ignored and results in no change to any sticky log bits. For example, to clear the TCC Activation Log bit and retain all other log bits, the Thermal Status Read should send a mask of 0xFFFFFFFD.

Figure 22. Thermal Status Word



2.2.2.6.4 Counter Snapshot / Read / Clear

A reference time and 'Thermally Constrained' time are managed in the processor. These two counters are managed via the Mailbox. These counters are valuable for detecting thermal runaway conditions where the TCC activation duty cycle reaches excessive levels.

The counters may be simultaneously snapshot, simultaneously cleared, or independently read. The simultaneous snapshot capability is provided in order to guarantee concurrent reads even with significant read latency over the PECI bus. Each counter is 32 bits wide.

Table 35. Counter Definition

Counter Name	Counter Number	Definition
Total Time	0x00	Counts the total time the processor has been executing with a resolution of approximately 1ms. This counter wraps at 32 bits.
Thermally Constrained Time	0x01	Counts the total time the processor has been operating at a lowered performance due to TCC activation. This timer includes the time required to ramp back up to the original P-state target after TCC activation expires. This timer does not include TCC activation time as a result of an external assertion of PROCHOT#.

2.2.2.6.5 Icc-TDC Read

Icc-TDC is the Intel® Xeon® processor C5500/C3500 series TDC current draw specification. This data may be used to confirm matching Icc profiles of processors in DP configurations. It may also be used during the processor boot sequence to verify processor compatibility with motherboard Icc delivery capabilities.

This command returns Icc-TDC in units of 1 Amp.

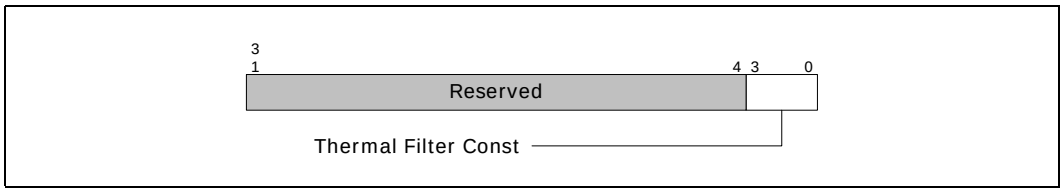
2.2.2.6.6 TCONTROL Read

TCONTROL is used for fan speed control management. The TCONTROL limit may be read over PECI using this Mailbox function. Unlike the in-band MSR interface, this TCONTROL value is already adjusted to be in the native PECI temperature format of a 2-byte, 2's complement number.

2.2.2.6.7 Thermal Data Config Read / Write

The Thermal Data Configuration register allows the PECI host to control the window over which thermal data is filtered. The default window is 256 ms. The host may configure this window by writing a Thermal Filtering Constant as a power of two. E.g., sending a value of 9 results in a filtering window of 2^9 or 512 ms.

Figure 23. Thermal Data Configuration Register



2.2.2.6.8 Machine Check Read

PECI offers read access to processor machine check banks 0, 1, 6, and 8.

Because machine check bank reads must be delivered through the Intel® Xeon® processor C5500/C3500 series Power Control Unit, it is possible that a fatal error in that unit will prevent access to other machine check banks. Host controllers may read Power Control Unit errors directly by issuing a PCIConfigRd() command of address 0x000000B0.

Figure 24. Machine Check Read MbxSend() Data Format

Byte #	0	1	2	3	4
Data	0x0A	Bank Index	Bank Number	Reserved	
	Request Type		Data[31:0]		

Table 36. Machine Check Bank Definitions

Bank Number	Bank Index	Meaning
0	0	MC0_CTL[31:0]
0	1	MC0_CTL[63:32]
0	2	MC0_STATUS[31:0]
0	3	MC0_STATUS[63:32]
0	4	MC0_ADDR[31:0]
0	5	MC0_ADDR[63:32]
0	6	MC0_MISC[31:0]
0	7	MC0_MISC[63:32]
1	0	MC1_CTL[31:0]
1	1	MC1_CTL[63:32]
1	2	MC1_STATUS[31:0]
1	3	MC1_STATUS[63:32]
1	4	MC1_ADDR[31:0]
1	5	MC1_ADDR[63:32]
1	6	MC1_MISC[31:0]
1	7	MC1_MISC[63:32]
6	0	MC6_CTL[31:0]
6	1	MC6_CTL[63:32]
6	2	MC6_STATUS[31:0]
6	3	MC6_STATUS[63:32]
6	4	MC6_ADDR[31:0]
6	5	MC6_ADDR[63:32]
6	6	MC6_MISC[31:0]
6	7	MC6_MISC[63:32]
8	0	MC8_CTL[31:0]
8	1	MC8_CTL[63:32]
8	2	MC8_STATUS[31:0]
8	3	MC8_STATUS[63:32]
8	4	MC8_ADDR[31:0]
8	5	MC8_ADDR[63:32]
8	6	MC8_MISC[31:0]
8	7	MC8_MISC[63:32]

2.2.2.6.9 T-State Throttling Control Read / Write

PECI offers the ability to enable and configure ACPI T-state (core clock modulation) throttling. ACPI T-state throttling forces all CPU cores into duty cycle clock modulation where the core toggles between C0 (clocks on) and C1 (clocks off) states at the specified duty cycle. This throttling reduces CPU performance to the duty cycle specified and, more importantly, results in processor power reduction.

The Intel® Xeon® processor C5500/C3500 series supports software initiated T-state throttling and automatic T-state throttling as part of the internal Thermal Monitor response mechanism (upon TCC activation). The PECI T-state throttling control register read/write capability is managed only in the PECI domain. In-band software may not manipulate or read the PECI T-state control setting. In the event that multiple agents are requesting T-state throttling simultaneously, the CPU always gives priority to the lowest power setting, or the numerically lowest duty cycle.

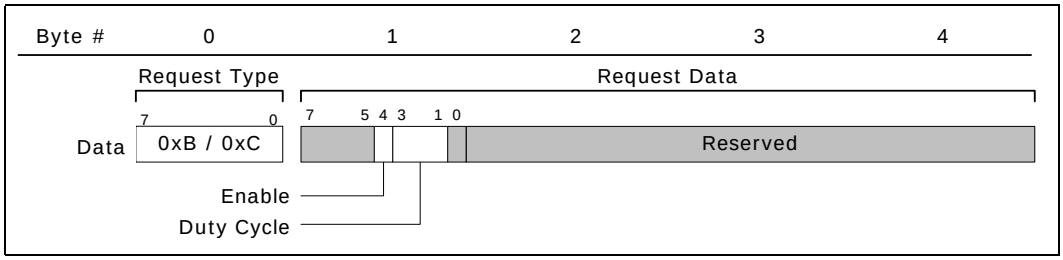
On the Intel® Xeon® processor C5500/C3500 series, the only supported duty cycle is 12.5% (12.5% clocks on, 87.5% clocks off). It is expected that T-state throttling will be engaged only under emergency thermal or power conditions. Future products may support more duty cycles, as defined in the following table.

Table 37. ACPI T-State Duty Cycle Definition

Duty Cycle Code	Definition
0x0	Undefined
0x1	**12.5% clocks on / 87.5% clocks off**
0x2	25% clocks on / 75% clocks off
0x3	37.5% clocks on / 62.5% clocks off
0x4	50% clocks on / 50% clocks off
0x5	62.5% clocks on / 37.5% clocks off
0x6	75% clocks on / 25% clocks off
0x7	87.5% clocks on / 12.5% clocks off

The T-state control word is defined as follows:

Figure 25. ACPI T-State Throttling Control Read / Write Definition



2.2.2.7 MbxSend()

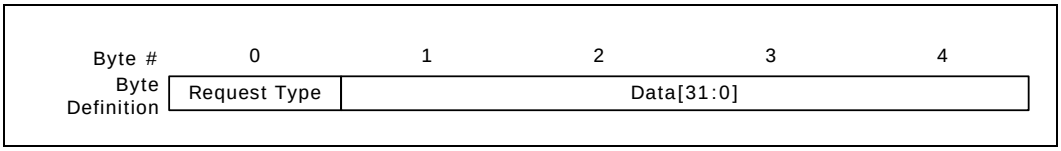
The MbxSend() command is utilized for sending requests to the generic Mailbox interface. Those requests are in turn serviced by the processor with some nominal latency and the result is deposited in the mailbox for reading. MbxGet() is used to retrieve the response and details are documented in Section 2.2.2.8.

The details of processor mailbox capabilities are described in Section 2.2.2.6.1, and many of the fundamental concepts of Mailbox ownership, release, and management are discussed in Section 2.2.2.9.

2.2.2.7.1 Write Data

Regardless of the function of the mailbox command, a request type modifier and 4-byte data payload must be sent. For Mailbox commands where the 4-byte data field is not applicable (e.g., the command is a read), the data written should be all zeroes.

Figure 26. MbxSend() Command Data Format



Because a particular MbxSend() command may specify an update to potentially critical registers inside the processor, it includes an Assured Write FCS (AW FCS) byte as part of the write data payload. See the *RS - Platform Environment Control Interface (PECI) Specification*, Revision 2.0 for a definition of the AW FCS protocol. In the event that the AW FCS mismatches with the client-calculated FCS, the client will abort the write and will always respond with a bad Write FCS.

2.2.2.7.2 Command Format

The MbxSend() format is as follows:

Write Length: 7

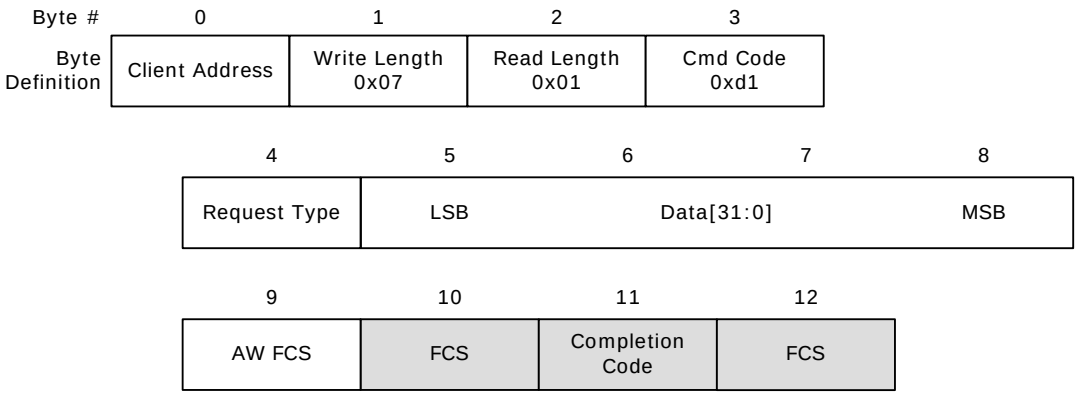
Read Length: 1

Command: 0xd1

Multi-Domain Support: Yes (see Table 41)

Description: Deposits the Request Type and associated 4-byte data in the Mailbox interface and returns a completion code byte with the details of the execution results. See Section 2.2.4.2 for completion code definitions.

Figure 27. MbxSend()



The 4-byte data defined above is sent in standard PECI ordering with LSB first and MSB last.

Table 38. MbxSend() Response Definition

Response	Meaning
Bad FCS	Electrical error
CC: 0x4X	Semaphore is granted with a Transaction ID of 'X'
CC: 0x80	Error causing a response timeout. Either due to a rare, internal timing condition or a processor RESET condition or processor S1 state. Retry is appropriate outside of the RESET or S1 states.
CC: 0x86	Mailbox interface is unavailable or busy

If the MbxSend() response returns a bad Read FCS, the completion code can't be trusted and the semaphore may or may not be taken. In order to clean out the interface, an MbxGet() must be issued and the response data should be discarded.

2.2.2.8 MbxGet()

The MbxGet() command is utilized for retrieving response data from the generic Mailbox interface as well as for unlocking the acquired mailbox. See Section 2.2.2.7 for details regarding the MbxSend() command. Many of the fundamental concepts of Mailbox ownership, release, and management are discussed in Section 2.2.2.9.

2.2.2.8.1 Write Data

The MbxGet() command is designed to retrieve response data from a previously deposited request. In order to guarantee alignment between the temporally separated request (MbxSend) and response (MbxGet) commands, the originally granted Transaction ID (sent as part of the passing MbxSend() completion code) must be issued as part of the MbxGet() request.

Any mailbox request made with an illegal or unlocked Transaction ID will get a failed completion code response. If the Transaction ID matches an outstanding transaction ID associated with a locked mailbox, the command will complete successfully and the response data will be returned to the originator.

Unlike MbxSend(), no Assured Write protocol is necessary for this command because this is a read-only function.

2.2.2.8.2 Command Format

The MbxGet() format is as follows:

Write Length: 2

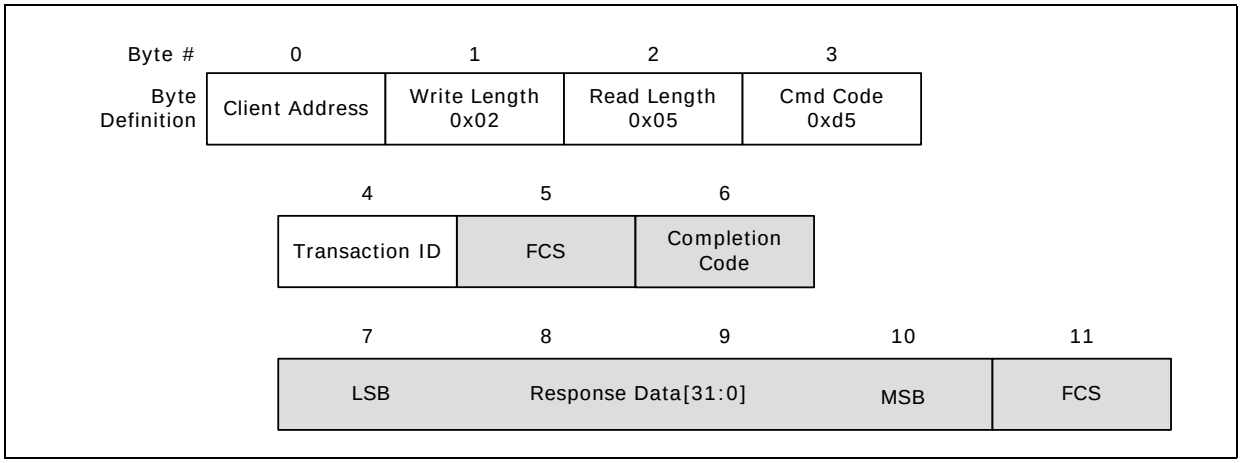
Read Length: 5

Command: 0xd5

Multi-Domain Support: Yes (see Table 41)

Description: Retrieves response data from mailbox and unlocks / releases that mailbox resource.

Figure 28. MbxGet()



The 4-byte data response defined above is sent in standard PECI ordering with LSB first and MSB last.

Table 39. MbxGet() Response Definition

Response	Meaning
Aborted Write FCS	Response data is not ready. Command retry is appropriate.
CC: 0x40	Command passed, data is valid.
CC: 0x80	Error causing a response timeout. Either due to a rare, internal timing condition or a processor RESET condition or processor S1 state. Retry is appropriate outside of the RESET or S1 states.
CC: 0x81	Thermal configuration data was malformed or exceeded limits.
CC: 0x82	Thermal status mask is illegal.
CC: 0x83	Invalid counter select.
CC: 0x84	Invalid Machine Check Bank or Index.
CC: 0x85	Failure due to lack of Mailbox lock or invalid Transaction ID.
CC: 0x86	Mailbox interface is unavailable or busy.
CC: 0xFF	Unknown/Invalid Mailbox Request.

2.2.2.9 Mailbox Usage Definition

2.2.2.9.1 Acquiring the Mailbox

The MbxSend() command is used to acquire control of the PECI mailbox and issue information regarding the specific request. The completion code response indicates whether or not the originator has acquired a lock on the mailbox, and that completion code always specifies the Transaction ID associated with that lock (see Section 2.2.2.9.2).

Once a mailbox has been acquired by an originating agent, future requests to acquire that mailbox will be denied with an ‘interface busy’ completion code response.

The lock on a mailbox is not achieved until the last bit of the MbxSend() Read FCS is transferred (in other words, it is not committed until the command completes). If the host aborts the command at any time prior to that bit transmission, the mailbox lock will be lost and it will remain available for any other agent to take control.

2.2.2.9.2 Transaction ID

For all MbxSend() commands that complete successfully, the passing completion code (0x4X) includes a 4-bit Transaction ID ('X'). That ID is the key to the mailbox and must be sent when retrieving response data and releasing the lock by using the MbxGet() command.

The Transaction ID is generated internally by the processor and has no relationship to the originator of the request. On the Intel® Xeon® processor C5500/C3500 series, only a single outstanding Transaction ID is supported. Therefore, it is recommended that all devices requesting actions or data from the mailbox complete their requests and release their semaphore in a timely manner.

In order to accommodate future designs, software or hardware utilizing the PECI mailbox must be capable of supporting Transaction IDs between 0 and 15.

2.2.2.9.3 Releasing the Mailbox

The mailbox associated with a particular Transaction ID is only unlocked / released upon successful transmission of the last bit of the Read FCS. If the originator aborts the transaction prior to transmission of this bit (presumably due to an FCS failure), the semaphore is maintained and the MbxGet() command may be retried.

2.2.2.9.4 Mailbox Timeouts

The mailbox is a shared resource that can result in artificial bandwidth conflicts among multiple querying processes that are sharing the same originator interface. The interface response time is quick, and with rare exception, back to back MbxSend() and MbxGet() commands should result in successful execution of the request and release of the mailbox. In order to guarantee timely retrieval of response data and mailbox release, the mailbox semaphore has a timeout policy. If the PECI bus has a cumulative '0 time of 1ms since the semaphore was acquired, the semaphore is automatically cleared. In the event that this timeout occurs, the originating agent will receive a failed completion code upon issuing a MbxGet() command, or even worse, it may receive corrupt data if this MbxGet() command so happens to be interleaved with an MbxSend() from another process. See Table 39 for more information regarding failed completion codes from MbxGet() commands.

Timeouts are undesirable, and the best way to avoid them and guarantee valid data is for the originating agent to always issue MbxGet() commands immediately following MbxSend() commands.

Alternately, mailbox timeout can be disabled. BIOS may write MSR MISC_POWER_MGMT (0x1AA), bit 11 to 0b1 in order to force a disable of this automatic timeout.

2.2.2.9.5 Response Latency

The PECI mailbox interface is designed to have response data available within plenty of margin to allow for back-to-back MbxSend() and MbxGet() requests. However, under rare circumstances that are out of the scope of this specification, it is possible that the response data is not available when the MbxGet() command is issued. Under these circumstances, the MbxGet() command will respond with an Abort FCS and the originator should re-issue the MbxGet() request.

2.2.3 Multi-Domain Commands

The Intel® Xeon® processor C5500/C3500 series does not support multiple domains, but it is possible that future products will, and the following tables are included as a reference for domain-specific definitions.

Table 40. Domain ID Definition

Domain ID	Domain Number
0b01	**0**
0b10	1

Table 41. Multi-Domain Command Code Reference

Command Name	Domain 0 Code	Domain 1 Code
GetTemp()	**0x01**	0x02
PCIConfigRd()	**0xC1**	0xC2
PCIConfigWr()	**0xC5**	0xC6
MbxSend()	**0xD1**	0xD2
MbxGet()	**0xD5**	0xD6

2.2.4 Client Responses

2.2.4.1 Abort FCS

The Client responds with an Abort FCS (See the *RS - Platform Environment Control Interface (PECI) Specification*) under the following conditions:

- The decoded command is not understood or not supported on this processor (this includes good command codes with bad Read Length or Write Length bytes).
- Data is not ready.
- Assured Write FCS (AW FCS) failure. Under most circumstances, an Assured Write failure will appear as a bad FCS. However, when an originator issues a poorly formatted command with a miscalculated AW FCS, the client will intentionally abort the FCS in order to guarantee originator notification.

2.2.4.2 Completion Codes

Some PECI commands respond with a completion code byte. These codes are designed to communicate the pass/fail status of the command and also provide more detailed information regarding the class of pass or fail. For all commands listed in Section 2.2.2 that support completion codes, each command's completion codes is listed in its respective section. What follows are some generalizations regarding completion codes.

An originator that is decoding these commands can apply a simple mask to determine pass or fail. Bit 7 is always set on a failed command, and is cleared on a passing command.

Table 42. Completion Code Pass/Fail Mask

0xxx xxxxb	Command passed
1xxx xxxxb	Command failed

Table 43. Device Specific Completion Code (CC) Definition

Completion Code	Description
0x00..0x3F	Device specific pass code
0x40	Command Passed
0x4X	Command passed with a transaction ID of 'X' (0x40 | Transaction_ID[3:0])
0x50..0x7F	Device specific pass code
CC: 0x80	Error causing a response timeout. Either due to a rare, internal timing condition or a processor RESET condition or processor S1 state. Retry is appropriate outside of the RESET or S1 states.
CC: 0x81	Thermal configuration data was malformed or exceeded limits.
CC: 0x82	Thermal status mask is illegal
CC: 0x83	Invalid counter select
CC: 0x84	Invalid Machine Check Bank or Index
CC: 0x85	Failure due to lack of Mailbox lock or invalid Transaction ID
CC: 0x86	Mailbox interface is unavailable or busy
CC:0xFF	Unknown/Invalid Mailbox Request

Note: The codes explicitly defined in this table may be useful in PECI originator response algorithms. All reserved or undefined codes may be generated by a PECI client device, and the originating agent must be capable of tolerating any code. The Pass/Fail mask defined in Table 42 applies to all codes and general response policies may be based on that limited information.

2.2.5 Originator Responses

The simplest policy that an originator may employ in response to receipt of a failing completion code is to retry the request. However, certain completion codes or FCS responses are indicative of an error in command encoding and a retry will not result in a different response from the client. Furthermore, the message originator must have a response policy in the event of successive failure responses.

See the definition of each command in Section 2.2.2 for a specific definition of possible command codes or FCS responses for a given command. The following response policy definition is generic, and more advanced response policies may be employed at the discretion of the originator developer.

Table 44. Originator Response Guidelines

Response	After One Attempt	After Three Attempts
Bad FCS	Retry	Fail with PECI client device error
Abort FCS	Retry	Fail with PECI client device error. May be due to illegal command codes.
CC: Fail	Retry	Either the PECI client doesn't support the current command code, or it has failed in its attempts to construct a response.
None (all 0's)	Force bus idle (1ms low), retry	Fail with PECI client device error. Client may be dead or otherwise non-responsive (in RESET or S1, for example).
CC: Pass	Pass	n/a
Good FCS	Pass	n/a

2.2.6 Temperature Data

2.2.6.1 Format

The temperature is formatted in a 16-bit, 2's complement value representing a number of 1/64 degrees centigrade. This format allows temperatures in a range of ±512°C to be reported to approximately a 0.016°C resolution.

Figure 29. Temperature Sensor Data Format

MSB Upper nibble					MSB Lower nibble					LSB Upper nibble					LSB Lower nibble			
S	x	x	x		x	x	x	x		x	x	x	x		x	x	x	x
Sign	Integer Value (0-511)											Fractional Value (~0.016)						

2.2.6.2 Interpretation

The resolution of the processor's Digital Thermal Sensor (DTS) is approximately 1°C, which can be confirmed by a RDMSR from IA32_THERM_STATUS MSR (0x19C) where it is architecturally defined. PECI temperatures are sent through a configurable low-pass filter prior to delivery in the GetTemp() response data. The output of this filter produces temperatures at the full 1/64°C resolution even though the DTS itself is not this accurate.

Temperature readings from the processor are always negative in a 2's complement format, and imply an offset from the reference TCC activation temperature. As an example, assume that the TCC activation temperature reference is 100°C. A PECI thermal reading of -10 indicates that the processor is running approximately 10°C below the TCC activation temperature, or 90°C. PECI temperature readings are not reliable at temperatures above TCC activation (since the processor is operating out of specification at this temperature). Therefore, the readings are never positive.

Changes in PECI data counts are approximately linear in relation to changes in temperature in degrees centigrade. A change of '1' in the PECI count represents roughly a temperature change of 1 degree centigrade. This linearity is approximate and cannot be guaranteed over the entire range of PECI temperatures, especially as the delta from the maximum PECI temperature (zero) increases.

2.2.6.3 Temperature Filtering

The processor digital thermal sensor (DTS) provides an improved capability to monitor device hot spots, which inherently leads to more varying temperature readings over short time intervals. Coupled with the fact that typical fan speed controllers may only read temperatures at 4 Hz, it is necessary for the thermal readings to reflect thermal trends and not instantaneous readings. Therefore, PECI supports a configurable low-pass temperature filtering function. By default, this filter results in a thermal reading that is a moving average of 256 samples taken at approximately 1msec intervals. This filter's depth, or smoothing factor, may be configured to between 1 sample and 1024 samples, in powers of 2. See the equation below for reference where the configurable variable is 'X'.

$$T_N = T_{N-1} + 1/2^X * (T_{SAMPLE} - T_{N-1})$$

See Section 2.2.2.6.7 for the definition of the thermal configuration command.

2.2.6.4 Reserved Values

Several values well out of the operational range are reserved to signal temperature sensor errors. These are summarized in the table below:

Table 45. Error Codes and Descriptions

Error Code	Description
0x8000	General Sensor Error (GSE)

2.2.7 Client Management

2.2.7.1 Power-up Sequencing

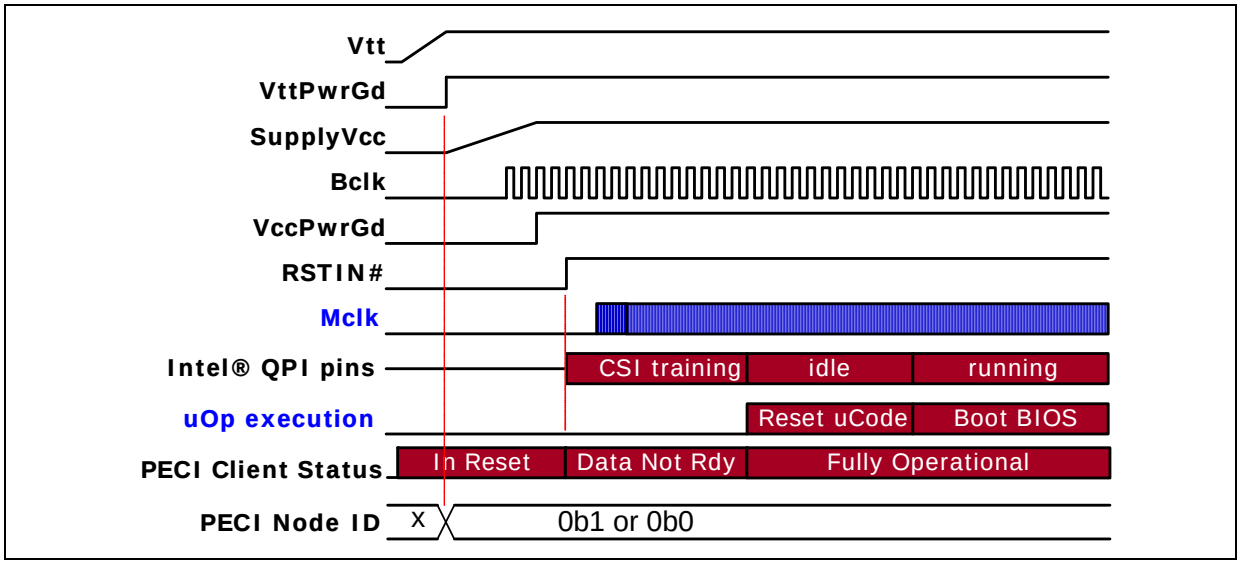
The PECI client is fully reset during processor RSTIN# assertion. This means that any transactions on the bus will be completely ignored, and the host will read the response from the client as all zeroes. After processor RSTIN# deassertion, the Intel® Xeon® processor C5500/C3500 series PECI client is operational enough to participate in timing negotiations and respond with reasonable data. However, the client data is not guaranteed to be fully populated until greater than 500 µS after processor RSTIN# is deasserted. Until that time, data may not be ready for all commands. The client responses to each command are as follows:

Table 46. PECI Client Response During Power-Up (During 'Data Not Ready')

Command	Response
Ping()	Fully functional
GetDIB()	Fully functional
GetTemp()	Client responds with a 'hot' reading, or 0x0000
PCIConfigRd()	Fully functional
PCIConfigWr()	Fully functional
MbxSend()	Fully functional
MbxGet()	Client responds with Abort FCS (if MbxSend() has been previously issued)

If the processor is tri-stated using power-on-configuration controls, the PECI client will also be tri-stated.

Figure 30. PECI Power-up Timeline



2.2.7.2 Device Discovery

The PECI client is available on all processors, and positive identification of the PECI revision number can be achieved by issuing the GetDIB() command. The revision number acts as a reference to the *RS - Platform Environment Control Interface (PECI) Specification*, Revision 2.0 document applicable to the processor client definition. See Section 2.2.2.2 for details on GetDIB response formatting.

2.2.7.3 Client Addressing

The PECI client assumes a default address of 0x30. If nothing special is done to the processor, all PECI clients will boot with this address. For DP enabled parts, a special PECI_ID# pin is available to strap each PECI socket to a different node ID. The package pin strap is evaluated at the assertion of VCCPWRGOOD (as depicted in Figure 30). Since PECI_ID# is active low, tying the pin to ground results in a client address of 0x31, and tying it to V_{TT} results in a client address of 0x30.

The client address may not be changed after VCCPWRGOOD assertion, until the next power cycle on the processor. Removal of a processor from its socket or tri-stating a processor in a DP configuration will have no impact to the remaining non-tri-stated PECI client address.

2.2.7.4 C-States

The Intel® Xeon® processor C5500/C3500 series PECI client is fully functional under all core and package C-states. Support for package C-states is a function of processor SKU and platform capabilities. All package C-states (C1/C1E, C3, and C6) are annotated here for completeness, but actual processor support for these C-states may vary.

Because the processor takes aggressive power savings actions under the deepest C-states (C1/C1E, C3, and C6), PECI requests may have an impact to platform power. The impact is documented below:

- Ping(), GetDIB(), GetTemp() and MbxGet() have no measurable impact on processor power under C-states.
- MbxSend(), PCIConfigRd() and PCIConfigWr() usage under package C-states may result in increased power consumption because the processor must temporarily return to a C0 state in order to execute the request. The exact power impact of a pop-up to C0 varies by product SKU, the C-state from which the pop-up is initiated, and the negotiated T_{BIT}.

Table 47. Power Impact of PECI Commands vs. C-states

Command	Power Impact
Ping()	Not measurable
GetDIB()	Not measurable
GetTemp()	Not measurable
PCIConfigRd()	Requires a package ‘pop-up’ to a C0 state
PCIConfigWr()	Requires a package ‘pop-up’ to a C0 state
MbxSend()	Requires a package ‘pop-up’ to a C0 state
MbxGet()	Not measurable

2.2.7.5 S-States

The PECI client is always guaranteed to be operational under S0 and S1 sleep states. Under S3 and deeper sleep states, the PECI client response is undefined and therefore unreliable.

Table 48. PECI Client Response During S1

Command	Response
Ping()	Fully functional
GetDIB()	Fully functional
GetTemp()	Fully functional
PCIConfigRd()	Fully functional
PCIConfigWr()	Fully functional
MbxSend()	Fully functional
MbxGet()	Fully functional

2.2.7.6 Processor Reset

The Intel® Xeon® processor C5500/C3500 series PECI client is fully reset on all RSTIN# assertions. Upon deassertion of RSTIN#, where power is maintained to the processor (otherwise known as a 'warm reset'), the following are true:

- The PECI client assumes a bus Idle state.
- The Thermal Filtering Constant is retained.
- PECI Node ID is retained.
- GetTemp() reading resets to 0x0000.
- Any transaction in progress is aborted by the client (as measured by the client no longer participating in the response).
- The processor client is otherwise reset to a default configuration.

2.3 SMBus

The Intel® Xeon® processor C5500/C3500 series has two V2.0 SMBus interfaces, one slave and one master. A third V2.0 SMBus master is provided by the PCH. The slave interface is a two signal/pin interface supporting a clock and a data line. The master interface is a two signal/pin interface supporting a clock, and a data line.

2.3.1 Slave SMBus

The IIO 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 IIO. The IIO's SMBus interface is capable of both accessing IIO registers and generating in-band 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 IIO's internal configuration registers. See the *SMBus Specification*, Revision 2.0 for the specific bus protocol, timings, and waveforms.

Warning: Since the IIO clock frequency is changed during the boot sequence, access to/from the IIO through the SMbus is not permitted during boot up.

SMBus features:

- The SMBus allows access to any register within the IIO portion of the Intel® Xeon® processor C5500/C3500 series, whether the CSR exists in PCI (bus, device,

function) space or in memory mapped space. The PECI registers are not within the IIO portion of the processor, and therefore cannot be accessed from the SMBus.

  — In a dual processor configuration, the SMBus Master (BMC for example) must use the SMBus slave local to each processor to access the IIO registers in that processor as remote peer to peer IO and configuration cycles are not supported.

- The SMBus interface acts as a side-band configuration access and must service all SMBus config transactions even in the presence of a processor deadlock condition.
- The slave SMBus supports Packet Error Checking (can be disabled) as defined in the SMBus 2.0 specification.
- The SMBus requires the SMBus master to poll the busy bit to determine if the previous transaction has completed. For reads, this is after the repeated start sequence.

2.3.2 Master SMBus

The IIO also includes a SMBUS master for PCIe hot plug. See Section 11.7.2, “PCIe Hot Plug” for further information.

2.3.3 SMBus Physical Layer

The component fabrication process does not support the pull-up voltage required by the SMBus protocol. Therefore, it will be required that voltage translators be placed on the platform to accommodate the differences in driving voltages. The IIO SMBus pads will operate at voltage of 1.1v. The IIO complies with the SMBus SCL frequency of 100 kHz.

2.3.4 SMBus Supported Transactions

The IIO supports six SMBus commands associated in read/write groups with three data sizes. Read transactions require two SMBus sequences: writing requested read address to internal register stack and the read command to extract data once it is available. Write transactions are a single sequence containing both address and data.

Supported transactions:

- Block Write (Dword sized data packet)
- Block Read (Dword sized data packet)
- Word Write (Word sized data packet)
- Word Read (Word sized data packet)
- Byte Write (Byte sized data packet)
- Byte Read (Byte sized data packet)

To support longer PCIe time-outs the SMBus master is required to poll the busy bit to know when data in the stack contains the desired data. This applies to both reads and writes. The protocol diagrams (Figure 31 through Figure 37) only show the polling in read transactions. This is due to the length of PCIe time-outs, which may be as long as several seconds. This will violate the SMBus spec of a maximum of 25 ms. To overcome this limitation, the SMBus slave will request the config master for access, once granted the slave asserts its busy bit and releases the link. The SMBus master is free to address other devices on the link or poll the busy bit until the IIO has completed the transaction.

Sequencing these commands initiates internal accesses to the component’s configuration registers. For high reliability, the interface supports the optional Packet Error Checking feature (CRC-8) and is enabled or disabled with each transaction.

Every configuration read or write first consists of an SMBus write sequence which initializes the Bus Number, Device, and so on. The term sequence is used since these variables may be written with a single block write or multiple word or byte writes. Once these parameters are initialized, the SMBus master can initiate a read sequence (which performs a configuration register read) or a write sequence (which performs a configuration register write).

Each SMBus transaction has an 8-bit command the master sends as part of the packet to instruct the IIO on handling data transfers. Command format is illustrated in Table 49 and subsequent bulleted list of sub-field encodings.

Table 49. 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 found in Section 2.3.9.1, “SMBus Configuration and Memory Block-Size Reads” on page 99 through Section 2.3.9.7, “SMBus Configuration and Memory Byte Writes” on page 104 illustrate when this bit should be set.
- The End bit indicates the last transaction of the read or write sequence. The examples in Section 2.3.9.1, “SMBus Configuration and Memory Block-Size Reads” on page 99 through Section 2.3.9.7, “SMBus Configuration and Memory Byte Writes” on page 104 best describe when this bit should be set.
- The MemTrans bit indicates the configuration request is a memory mapped addressed register or a PCI (bus, device, function, offset) addressed register. 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, then there would be no PEC generated or checked by the slave.
- The Internal Command field specifies the internal command to be issued by the SMBus slave. The IIO 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 length of transfer so that 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. Table 50 provides a list of the bytes in the stack and their descriptions.

Table 50. Internal SMBus Protocol Stack

SMBus Stack usage for bus/dev/func commands (cmd[5] = 0)	SMBus Stack usage for memory region commands (cmd[5] = 1)	Description
Command	Command	Command byte
Byte Count	Byte Count	The number of bytes for this transaction when Block command is used.
Bus Number	Memory region	Bus number for bus/dev config space command type. Memory region for memory config space command type.
Device/Function	Address [23:16]	Device[4:0] and Function[2:0] for cmd[5] = 0 type of config transaction. Address[23:16] for cmd[5] = 1 type of memory config transaction.
Register Offset	Address [15:8]	The following fields are further defined for cmd[5]=0: Address High[7:4] = Reserved[3:0] Address High [3:0] = Register Offset[11:8]: This is the high order PCIe address field. The following fields are further defined for cmd[5]=1: Address[15:8]
Register Offset	Address [7:0]	The following fields are further defined for cmd[5]=0: Lower order 8-bit register offset (Address[7:0]) The following fields are further defined for cmd[5]=1: Address [7:0]
Data3	Data3	Data byte 3
Data2	Data2	Data byte 2
Data1	Data1	Data byte 1
Data0	Data0	Data byte 0

2.3.5 Addressing

The slave address that each component claims is dependent on the DMI_PE_CFG# pin strap (sampled on the assertion of PWRGOOD). The IIO claims SMBus accesses with address[7:1] = 1110_1X0. The X's represent inversion of the DMI_PE_CFG# strap pin on the IIO. See Table 51 for the mapping of strap pins to the bit positions of the slave address.

Note: The slave address is dependent on the DMI_PE_CFG# strap pin only and cannot be reprogrammed.

Table 51. SMBus Slave Address Format

Slave Address Field Bit Position	Slave Address Source
[7]	1
[6]	1
[5]	1

Table 51. SMBus Slave Address Format

Slave Address Field Bit Position	Slave Address Source
[4]	0
[3]	1
[2]	Inversion of DMI_PE_CFG# strap pin
[1]	0
[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 (MemTrans) bit as described in Table 49, "SMBus Command Encoding" is cleared then 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 52 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 then 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 IIO slave will perform the following:

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

Table 52. Memory Region Address Field

Bit Field	Memory Region Address Field
0Fh	LT_QPII/LT_LT BAR
0Eh	LT_PR_BAR
0Dh	LT_PB_BAR
0Ch	NTB Secondary memory BAR, (SBAR01BASE)
0Bh	NTB Primary memory BAR, (PBAR01BASE)
0Ah	DMI RC memory BAR, (DMIRCBAR)
09h	IOAPIC memory BAR, (MBAR/ABAR)
08h	Intel® VT-d memory BAR, (VTBAR)
07h	Intel® QuickData Technology memory BAR 7,(CB_BAR7)
06h	Intel® QuickData Technology memory BAR 6,(CB_BAR6)
05h	Intel® QuickData Technology memory BAR 5,(CB_BAR5)
04h	Intel® QuickData Technology memory BAR 4,(CB_BAR4)
03h	Intel® QuickData Technology memory BAR 3,(CB_BAR3)
02h	Intel® QuickData Technology memory BAR 2,(CB_BAR2)
01h	Intel® QuickData Technology memory BAR 1,(CB_BAR1)
00h	Intel® QuickData Technology memory BAR 0 (CB_BAR0)

2.3.6 SMBus Initiated Southbound Configuration Cycles

The platform SMBus master agent that is connected to an IIO 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 IIO (i.e. not the IIO's bus number), it sends the request to port with the bus address. All requests outside ofthis range are sent to the legacy ESI port for a master abort condition.

2.3.7 SMBus Error Handling

SMBus Error Handling feature list:

- Errors are reported in the status byte field.
- Errors in Table 53 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 IIO 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 must be routed to the FERR and NERR registers for error reporting. The status field encoding is defined in Table 53. This field reports if an error occurred. If bits[2:0] are 000b then transaction was successful only to the extent that the IIO 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 53. Status Field Encoding for SMBus Reads

Bit	Description
7	Busy
6:3	Reserved
2:0	100-111: Reserved 011: Master Abort. An error that is reported by the IIO 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 a memory region is not valid. 000: Successful

2.3.8 SMBus Interface Reset

The slave interface state machine can be reset in several ways. The first two items are defined in the SMBus rev2.0 specification.

- 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 us.
- Force a platform reset.

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 in reset.

2.3.9 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. This is due to the PCIe time-outs that may be long, causing an SMBus spec violation. The SMBus master must poll the busy bit to determine when the pervious read transaction has completed.

If an error occurs then the status byte will report the results. This 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 PEC (Packet Error Code) 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, see 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.

2.3.9.1 SMBus Configuration and Memory Block-Size Reads

Figure 31. SMBus Block-Size Configuration Register Read

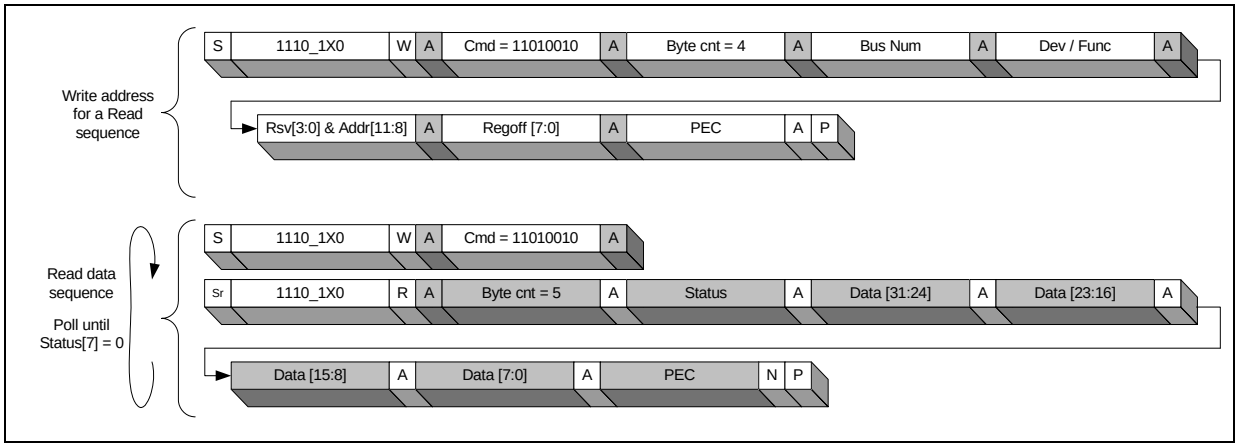
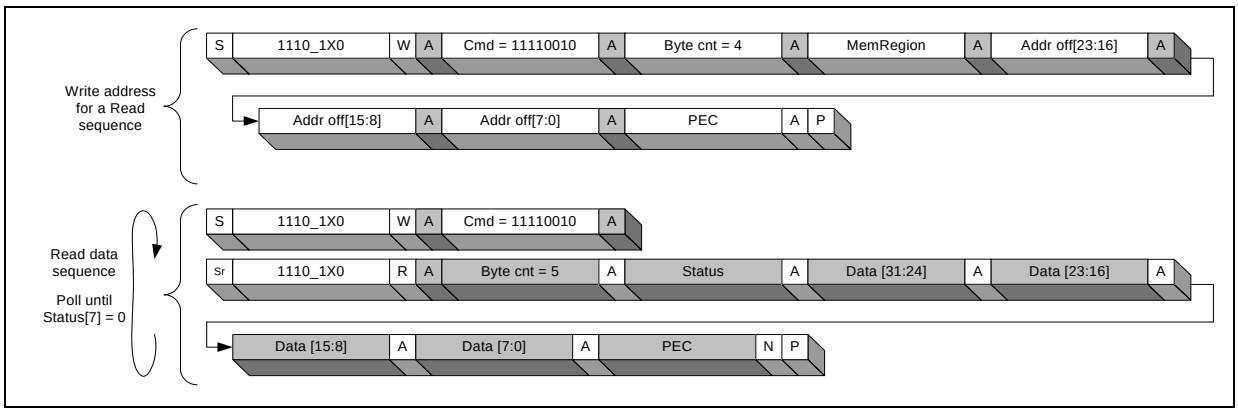


Figure 32. SMBus Block-size Memory Register Read



2.3.9.2 SMBus Configuration and Memory Word-Size Reads

Figure 33. SMBus Word-Size Configuration Register Read

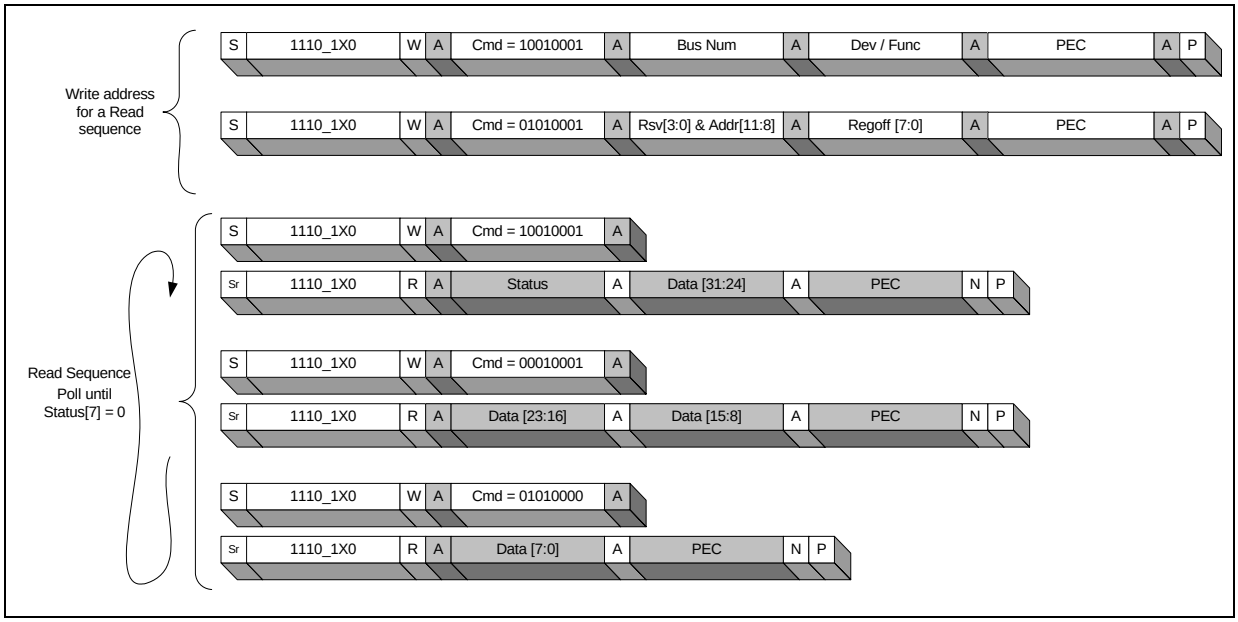
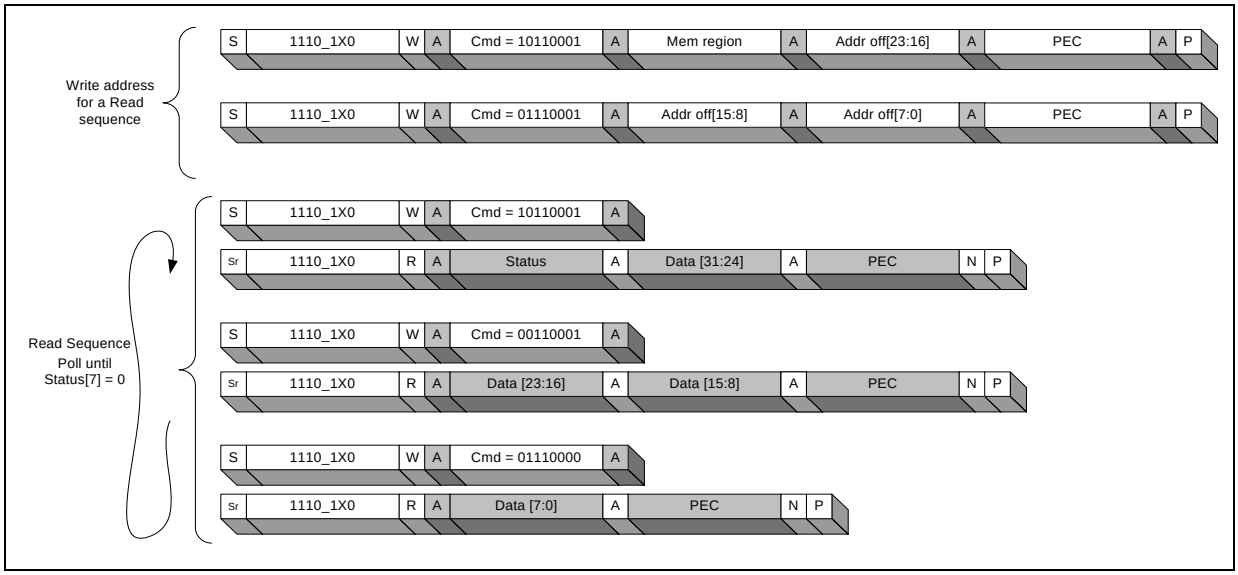


Figure 34. SMBus Word-Size Memory Register Read



2.3.9.3 SMBus Configuration and Memory Byte Reads

Figure 35. SMBus Byte-Size Configuration Register Read

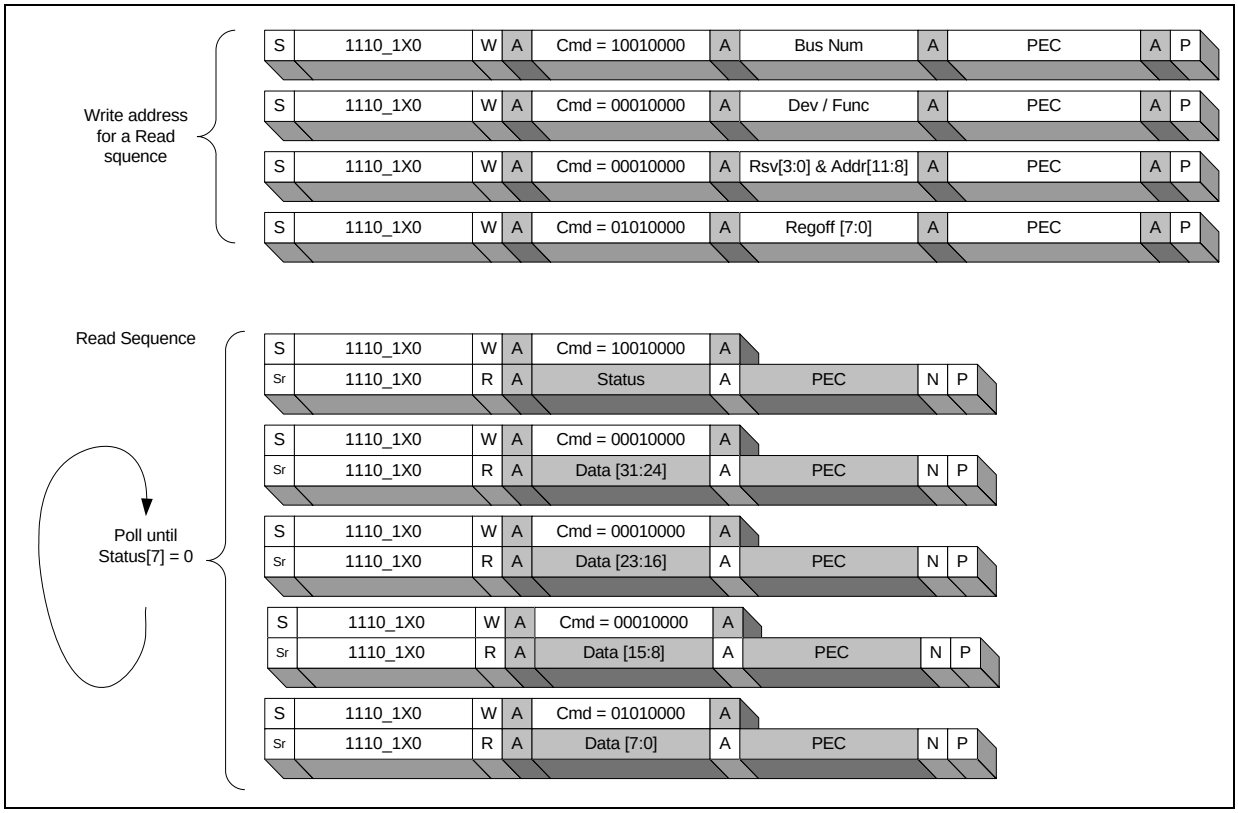
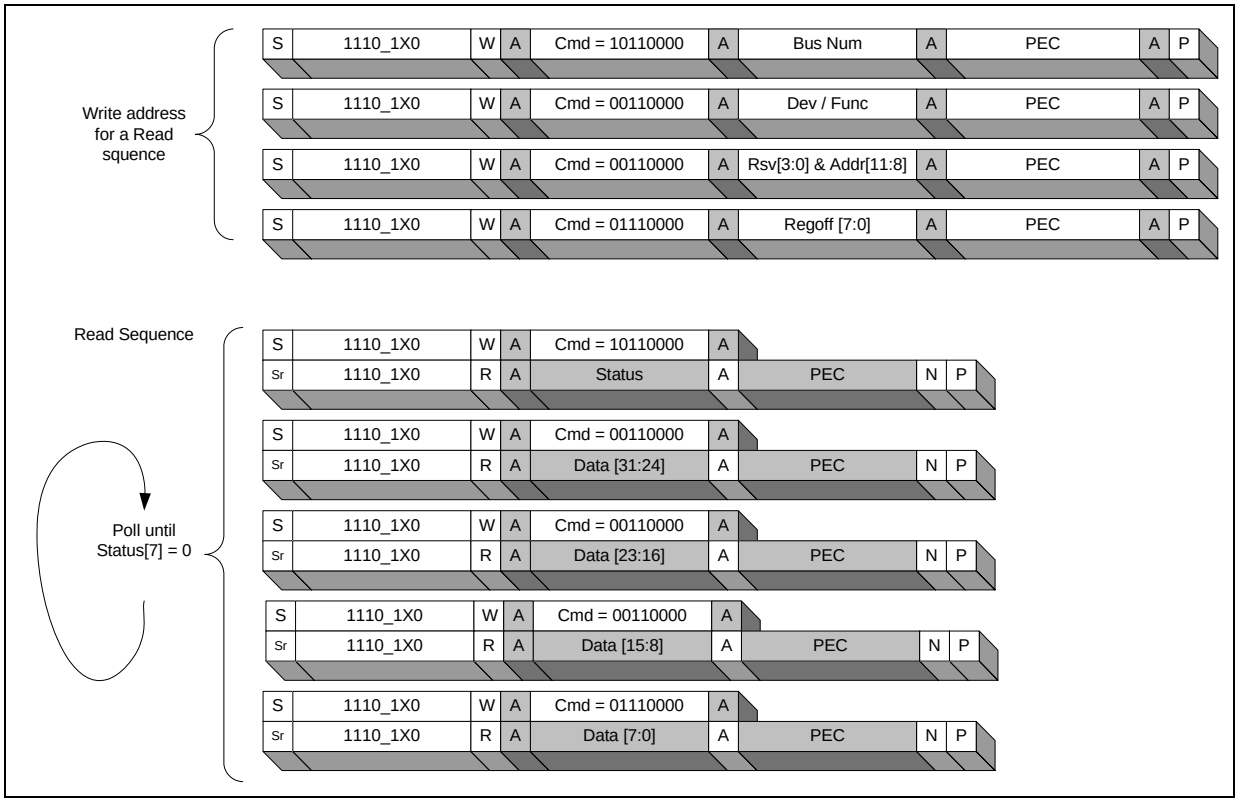


Figure 36. SMBus Byte-Size Memory Register Read



2.3.9.4 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 the 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 that 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 occurred, the SMBus interface NACKs the last write operation just before the stop bit.

The busy bit will be set for the write transaction. A config write to the IIO will most likely complete before the SMBus master can poll the busy bit. If the transaction is destined to a chip on a PCIe link then it could take several more clock cycle to complete the outbound transaction being sent.

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

2.3.9.5 SMBus Configuration and Memory Block Writes

Figure 37. SMBus Block-Size Configuration Register Write

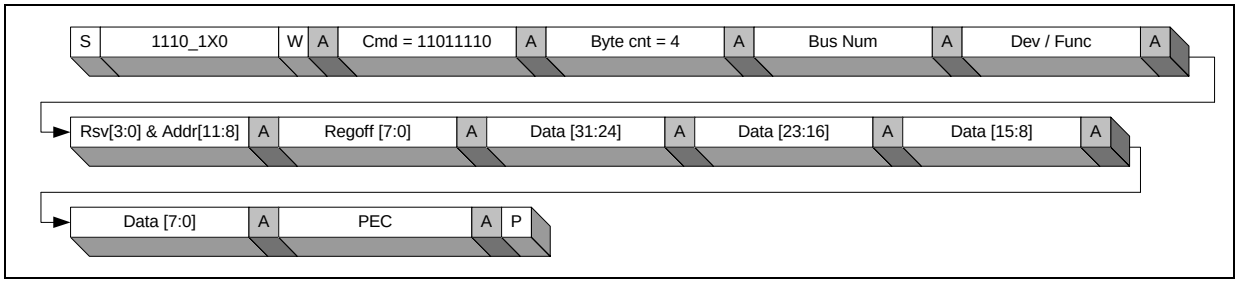
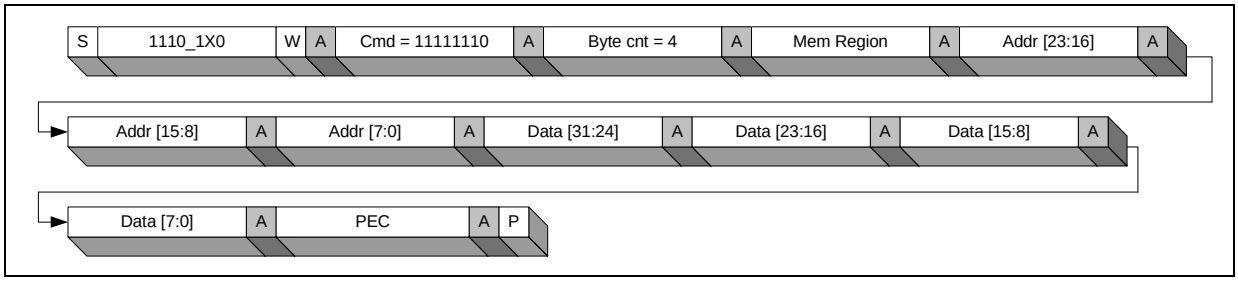


Figure 38. SMBus Block-Size Memory Register Write



2.3.9.6 SMBus Configuration and Memory Word Writes

Figure 39. SMBus Word-Size Configuration Register Write

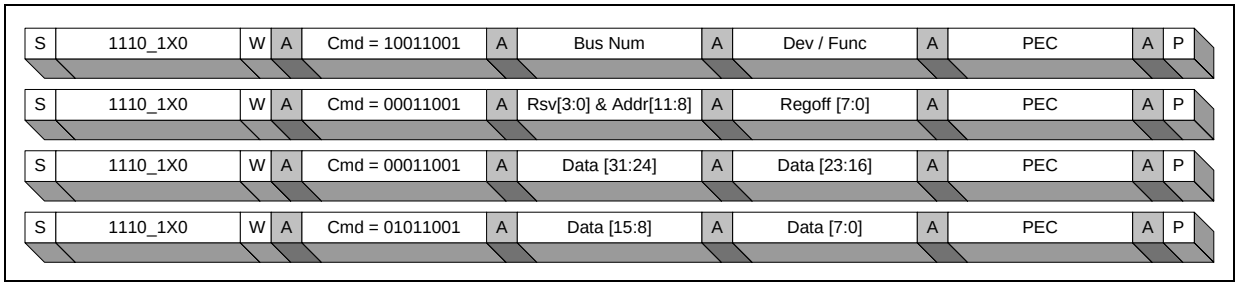
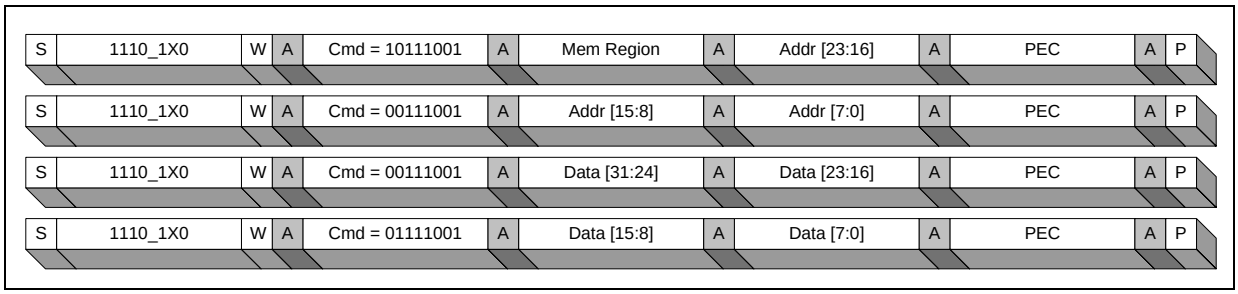


Figure 40. SMBus Word-Size Memory Register Write



2.3.9.7 SMBus Configuration and Memory Byte Writes

Figure 41. SMBus Configuration (Byte Write, PEC enabled)

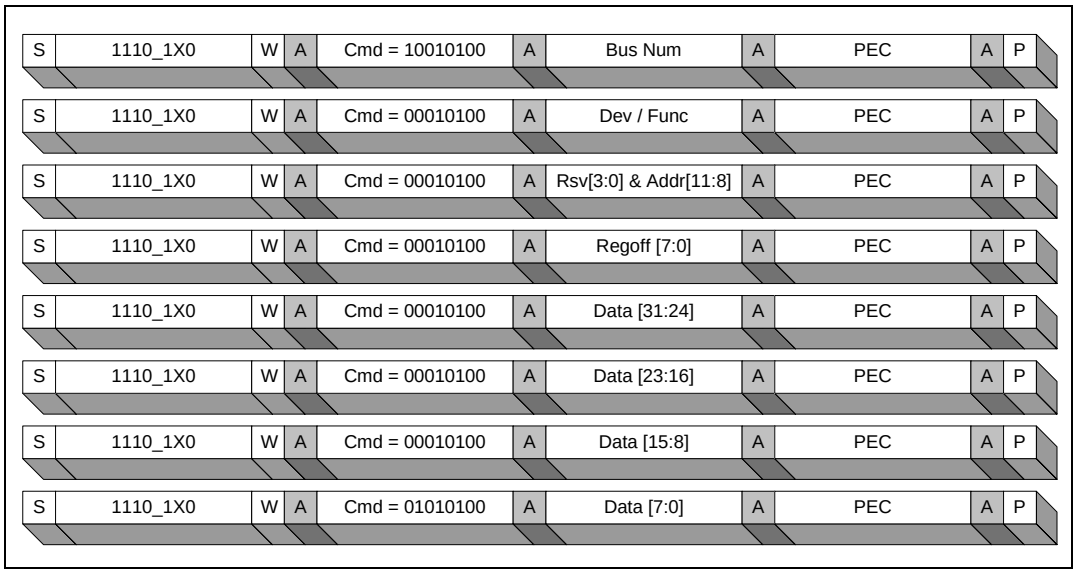
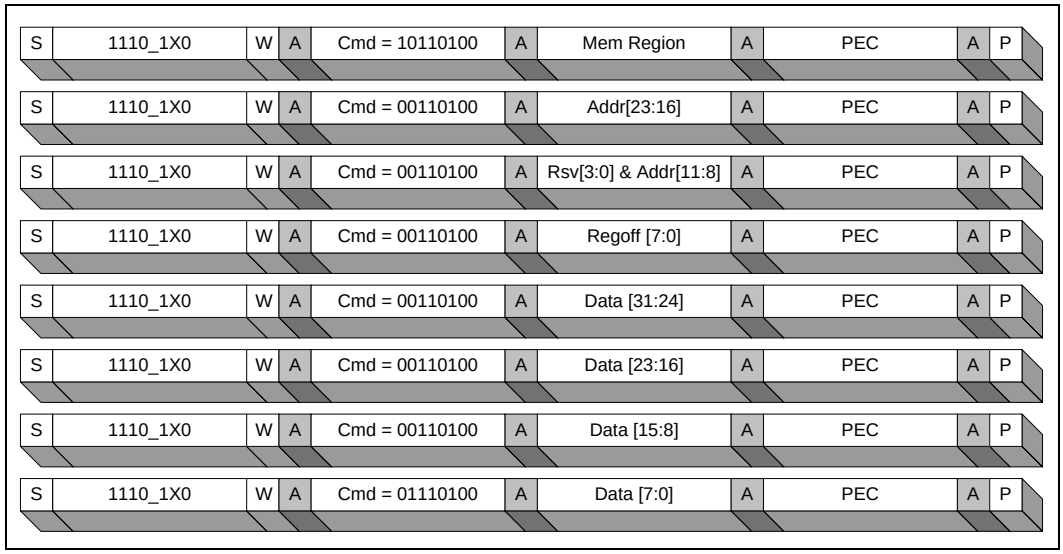


Figure 42. SMBus Memory (Byte Write, PEC enabled)



2.4 Intel® QuickPath Interconnect (Intel® QPI)

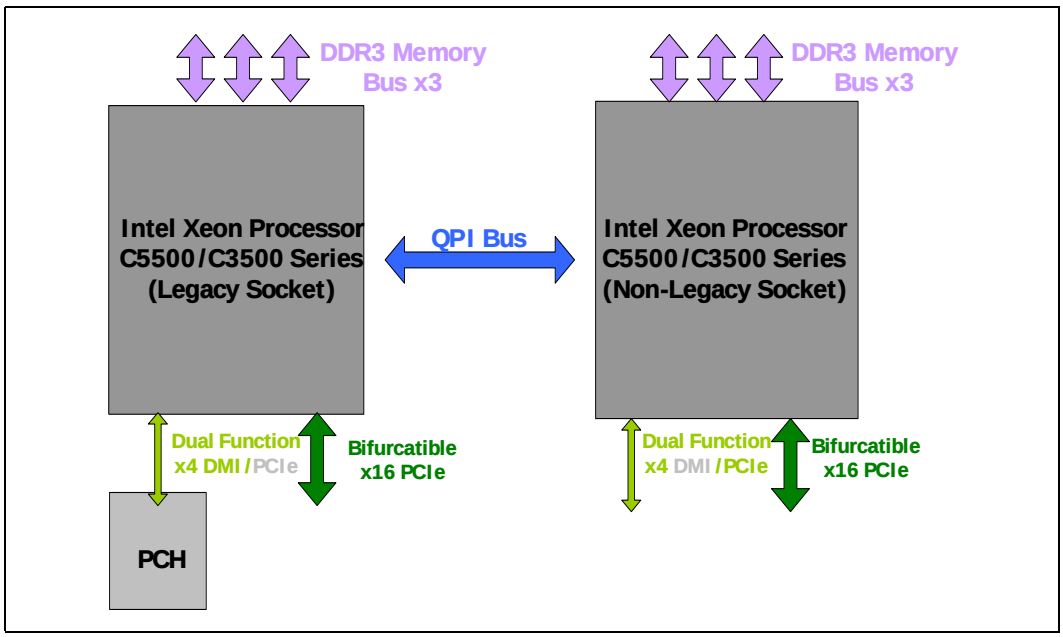
Intel® QuickPath Interconnect (Intel® QPI) is an Intel-developed cache-coherent, link based interface used for interconnecting processors, chipsets, bridges, and various acceleration devices. There is one internal link between the core complex and the IIO module and there may be one external link to the non-legacy processor or to an IOH. This section discusses the external link.

2.4.1 Processor's Intel® QuickPath Interconnect Platform Overview

Figure 43 represents a simplified block diagram of a dual processor (DP) Intel® Xeon® processor C5500/C3500 series platform, showing how the Intel® QPI bus is used to interconnect the two processors. The QPI bus is used to seamlessly interconnect the resources from one processor to another, whether it be one processor accessing data in the memory managed by the alternate processor, one processor accessing a PCIe port residing on the alternate processor, or a PCIe P2P transfer involving two PCIe ports residing on different processors. The Intel® QPI physical layer, link layer, and protocol layer are implemented in hardware. No special software or drivers are required, other than firmware to initialize the Intel® QPI link.

The Intel® QPI link is a serial point-to-point connection, consisting of 20 lanes in each direction. The Intel® QPI bus is the only method of exchanging data between the two processors, there are no additional sideband signals. The Intel® QPI bus is gluelessly connected between processors, no additional hardware required.

Figure 43. Intel® Xeon® Processor C5500/C3500 Series Dual Processor Configuration Block Diagram



Data needing to go between processors is converted into packets that are transmitted serially over the Intel® QPI bus, rather than previous Intel architectures that used the parallel 64-bit Front Side Bus (FSB). Intel® Xeon® processor C5500/C3500 series SKUs are available supporting various Intel® QPI link data rates.

The Intel® QuickPath Interconnect architecture is partitioned into five layers, one of which is optional depending on the platform specifics. Section 2.4.2 through Section 2.4.9.1 provide an overview of each of the layers.

2.4.2 Physical Layer Implementation

The physical layer of the Intel® QPI bus is the physical entity between two components, it uses a differential signalling scheme, and is responsible for the electrical transfer of data.

2.4.2.1 Processor's Intel® QuickPath Interconnect Physical Layer Attributes

The processor's Intel® QuickPath Interconnect Physical layer attributes are summarized in Table 54 below.

Table 54. Processor's Intel® QuickPath Interconnect Physical Layer Attributes

Feature	Supported	Notes
Support for full width (20 bit) links	Yes	
Support for half width (10 bit) links	No	
Support for quarter width (5 bit) links	No	
Link Self Healing	No	
Clock channel failover	No	
Lane Reversal	Yes	
Polarity Reversal	Yes	
Hot-Plug support	No	
Independent control of link width in each direction	No	
Link Power Management - L0s	Yes	An adaptive L0s scheme, where the idle threshold is continually adjusted by hardware. The associated parameters are fixed at the factory and do not require software programming.
Link Power Management - L1	Yes	

2.4.3 Processor's Intel® QuickPath Interconnect Link Speed Configuration

Intel® QuickPath Interconnect link initialization is performed following a VCCPWRGOOD reset. At reset, the Intel® QuickPath Interconnect links come up in slow mode (66 MT/s). BIOS must then determine the speed at which to run the Intel® QuickPath Interconnect links in full speed mode, program the transmitter equalization parameters and issue a processor-only reset to bring the Intel® QuickPath Interconnect links to full speed. The equalization parameters are dependent on the specific board design, and it is expected these parameters will be hard coded in the BIOS. Once the Intel® QuickPath Interconnect links transition to full speed, they cannot go back to slow mode without a VCCPWRGOOD reset.

The maximum supported Intel® QuickPath Interconnect link speed is processor SKU dependent.

2.4.3.1 Detect Intel® QuickPath Interconnect Speeds Supported by the Processors

The BIOS can detect the minimum and maximum Intel® QuickPath Interconnect data rate supported by a processor. This information is indicated by the following processor CSRs: QPI_0_PLL_STATUS and QPI_1_PLL_STATUS. The BSP can also read the CSRs of the other processor, without assistance from the other processor. Both processors must be initialized to the same Intel® QuickPath Interconnect data rates.

2.4.4 Intel® QuickPath Interconnect Probing Considerations

When a Logic Analyzer probe is present on the Intel® QuickPath Interconnect links (for hardware debug purposes), the characteristics of the Intel® QuickPath Interconnect link are changed. This requires slightly different transmitter equalization parameters and retraining period. It is expected that these alternate parameters will be stored in BIOS. There is no mechanism for automatically detecting the presence of probes. Therefore, the BIOS must be told if the probes are present in order to load the correct equalization parameters. Using the incorrect set of equalization parameters (with probes and without probes) will cause the platform to not boot reliably.

2.4.5 Link Layer

The Link layer abstracts the physical layer from the upper layers, and provides reliable data transfer and flow control between two directly connected Intel® QuickPath Interconnect entities. It is responsible for virtualization of a physical channel into multiple virtual channels and message classes.

2.4.5.1 Link Layer Attributes

Intel® QuickPath Interconnect Link layer attributes are summarized in Table 55 below.

Table 55. Intel® QuickPath Interconnect Link Layer Attributes

Feature	Support	Notes
Number of Node IDs supported	4	Four for a DP system, two for a UP system.
Packet Format	DP	
Extended Header Support	No	
Virtual Networks Supported	VN0, VNA	
Viral indication	No	
Data Poisoning	Yes	
Simple CRC (8 bit)	Yes	
Rolling CRC (16 bit)	No	

2.4.6 Routing Layer

The Routing layer provides a flexible and distributed way to route Intel® QuickPath Interconnect packets from source to destination. The routing is based on the destination. It relies on the virtual channel and message class abstraction of the link layer to specify the Intel® QuickPath Interconnect port(s) and virtual network(s) on which to route a packet. The mechanism for routing is defined through implementation of routing tables.

2.4.6.1 Routing Layer Attributes

The Intel® QuickPath Interconnect Routing layer attributes are summarized in Table 56 below.

Table 56. Intel® QuickPath Interconnect Routing Layer Attributes

Feature	Support
Through routing capability for processors	Yes

2.4.7 Intel® QuickPath Interconnect Address Decoding

On past FSB platforms, the processors and I/O subsystem could direct all memory and I/O accesses to the North Bridge. The processor's Intel® QuickPath Interconnect is more distributed in nature. The memory controller is integrated inside the processor. Therefore, a processor may be able to resolve memory accesses locally or may have to send it to another processor.

Each Intel® QuickPath Interconnect agent that is capable of accessing a system resource (system memory, MMIO, etc) needs a way to determine the Intel® QuickPath Interconnect agent that owns that resource. This is accomplished by Source Address Decoders (SAD). Each Intel® QuickPath Interconnect agent contains a Source Address Decoder whereby a lookup process is used to convert a physical address to the Node ID of the Intel® QuickPath Interconnect agent that owns that address.

In some Intel® QuickPath Interconnect implementations, each Intel® QuickPath Interconnect agent may have multiple Intel® QuickPath Interconnect links and needs to know which of the links can be used to reach the target agent. This job is handled via a Routing Table (RT). The Routing Table takes the target Node ID and provides a link number. The target agent may then need to perform another level of lookup to determine how to satisfy the request (e.g., a memory controller may need to determine which of many memory channels contains the target address). This lookup structure is called Target Address Decode (TAD).

The Intel® Xeon® processor C5500/C3500 series implements a fixed Intel® QuickPath Interconnect routing topology that simplifies the SAD, RT and TAD structures and also simplifies programming of these structures. Memory SAD entries in the processor directly refer to a target package number and not a Node ID. The processor knows which package is local and which is remote and therefore, either satisfies the request internally, or sends it to the remote package over the processor-processor Intel® QuickPath Interconnect link.

2.4.8 Transport Layer

The Intel® QuickPath Interconnect Transport Layer is not implemented on the Intel® Xeon® processor C5500/C3500 series. The Transport layer is optional in the Intel® QuickPath architecture as defined.

2.4.9 Protocol Layer

The Protocol layer implements the higher level communication protocol between nodes, such as cache coherence (reads, writes, invalidates), ordering, peer-to-peer I/O, interrupt delivery etc. The write-invalidate protocol implements the MESIF states, where the MESI states have the usual connotation (Modified, Exclusive, Shared, Invalid), and the F state indicates a read-only forwarding state.

2.4.9.1 Protocol Layer Attributes

The processor's Intel® QuickPath Interconnect Protocol layer attributes are summarized in Table 57 through Table 62 below.

2.4.9.2 Intel® QuickPath Interconnect Coherent Protocol Attributes

Table 57. Processor's Intel® QuickPath Interconnect Coherent Protocol Attributes

Coherence Protocol	Support
Supports Coherence protocol with in-order home channel	Yes
Supports Coherence protocol with out-of-order home channel	No
Supports Snoopy Caching agents	Yes
Supports Directory Caching agents	No
Supports Critical Chunk data order for coherent transactions	Yes
Generates Buried HITM transaction cases	No
Supports Receiving Buried HITM cases	Yes

2.4.9.3 Intel® QuickPath Interconnect Non-Coherent Protocol Attributes

Table 58. Picket Post Platform Intel® QuickPath Interconnect Non-Coherent Protocol Attributes

Non-Coherence Protocol	Support
Peer-to-peer tunnel transactions	Yes
Virtual Legacy Wire (VLW) transactions	Yes
Special cycle transactions	N/A
Locked accesses	Yes

2.4.9.4 Interrupt Handling

Table 59. Intel® QuickPath Interconnect Interrupts Attributes

Interrupt Attribute	Support
Processor initiated Int Transaction on Intel® QuickPath Interconnect link	Yes
Logical interrupts (IntLogical)	Yes
Broadcast of logical and physical mode interrupts	Yes
Logical Flat Addressing Mode (<= 8 threads)	Yes
Logical Cluster Addressing Mode (<= 60 threads)	Yes
EOI	Yes
Support for INIT, NMI, SMI, and ExtINT through Virtual Legacy Wire (VLW) transaction	Yes
Support for INIT, NMI, and ExtINT through Int transaction	Yes
Limit on number of threads supported for inter-processor interrupts	8

2.4.9.5 Fault Handling

Table 60. Intel® QuickPath Interconnect Fault Handling Attributes

Interrupt Attribute	Support
Machine check indication through Int	No
Time-out hierarchy for fault diagnosis	Only via 3-strike counter
Packet elimination for error isolation between partitions	No
Abort time-out response	Only via 3-strike counter

2.4.9.6 Reset / Initialization

Table 61. Intel® QuickPath Interconnect Reset / Initialization Attributes

Interrupt Attribute	Support
NodeID Assignment	strap assignment
Processor accepting external configuration (NcRd, NcWr, CfgRd, CfgWr going to CSRs) requests	Yes
Separation of reset domains between link and physical layer for link self-healing	N/A
Separation of reset domains between routing/protocol and link layer for hot-plug	N/A
Separation of reset domains between Intel® QuickPath Interconnect entities and routing layer to allow sub-socket partitioning	No
Product specific fixed and configurable power on configuration values-configurable through link parameter exchange.	Yes
Flexible firmware location through discovery during link initialization	No
Packet routing during initialization before route table and address decoder is initialized	Configurable through link init parameter

2.4.9.7 Other Attributes

Table 62. Intel® QuickPath Interconnect Other Attributes

General System Management	Support
Protected system configuration region	No
Support for various partitioning models	No
Support for Link level power management	Yes

2.5 IIO Intel® QPI Coherent Interface and Address Decode

2.5.1 Introduction

This section discusses the internal coherent interface between the CPU complex and the IIO complex. It is based on the Intel® QuickPath Interconnect. IIO address decoding mechanisms are also discussed.

2.5.2 Link Layer

There are 128 Flit (Flow control unit of transfer) link layer credits to be split between VN0 and VNA virtual channels from the IIO. One VN0 credit is used per Intel® QPI message class in the normal configuration, which consumes a total of 26 Flits in the Flit buffer. For UP systems, with the six Intel® QPI message classes supported, this will leave the remaining 102 Flits to be used for VNA credits. For DP systems, the route through VN0 traffic requires a second VN0 credit per channel to be allocated, making a minimum of 52 Flits consumed by CPU and route through traffic, leaving 76 Flits remaining to be split between CPU and Route Through VNA traffic. Bias register is implemented to allow configurability of the 72 Flits split between CPU and route through traffic. The default sharing of the VNA credits will be 36/36, but biasing registers can be used to give more credits to either normal or route through traffic.

2.5.2.1 Link Error Protection

Error detection is done in the link layer using CRC. 8-bit CRC is supported. However, link layer retry (LLR) is not supported and must be disabled by the BIOS.

2.5.2.2 Message Class

The link layer defines six Message Classes. The IIO supports four of those channels for receiving and six for sending. Table 63 shows the message class details.

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

Table 63. Supported Intel® QPI Message Classes

Message Class	VC Description	Send Support	Receive Support
SNP	Snoop Channel. Used for snoop commands to caching agents.	Yes	No
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	No
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

2.5.2.3 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 VN0 credits are tracked on a packet granularity.

2.5.2.4 Ordering

The IIO link layer keeps each message class ordering independent. Credit management is kept independent on VN0. This ensures that each message class may bypass the other in blocking conditions.

Ordering is not assumed within a single Message Class, except for the Home Message Class. The Home Message Class coherence conflict resolution requires ordering between transactions corresponding to the same cache line address.

VNA and VN0 follow similar ordering to the message class being transported on it. With Home message class requiring ordering across VNA/VN0 for the same cache line, all other message classes have no ordering requirement.

2.5.3 Protocol Layer

The protocol layer is responsible for translating requests from the core into the Intel® QPI domain, and for maintaining protocol semantics. The IIO is a Intel® QPI caching agent. It is also a fully-compliant 'IO' (home) agent for non-coherent I/O traffic. Source Broadcast mode supports up to two peer caching agents. In DP, there are three peer caching agents, but the other IIO is not snooped due to the invalidating write back flow. Lock arbiter support in IA-32 systems is provided for up to eight processor lock requesters.

The protocol layer supports 64 B cache lines. All transactions from PCI Express are broken up into 64 B aligned requests to match Intel® QPI packet size and alignment requirements. Transactions of less than a cache line are also supported using the 64 B packet framework in Intel® QPI.

2.5.4 Snooping Modes

The IIO contains an 8b vector to indicate peer caching agents that specifies up to eight peer agents that are involved in coherency. In UP profile this vector is always empty.

In DP system, there will be three peer caching agents: Home CPU, non-Home CPU, and remote IIO. With the Invalidating Write Back flow, only both CPUs need to be snooped so two bits are set. The IIO Intel® QPI logic handles the masking of snooping the home agent.

2.5.5 IIO Source Address Decoder (SAD)

Every inbound request going to Intel® QPI 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 remote peer to peer MMIO accesses, the inbound request must also look at the SAD to determine the node ID of the other IIO. These are not home NodeID requests.

In UP profile, all inbound requests are sent to a single target NodeID. When in this mode the SAD is only used to decode legal ranges and the Target NodeID is ignored.

In DP profile the source address decoder is only used for decode of the DRAM address ranges and APIC targets to find the correct home NodeID. In the DP profile the SAD also decodes peer IIO address ranges. Other ranges including any protected memory holes are decoded elsewhere. See Chapter 6.0, "System Address Map" for more details.

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

- Memory Address - Memory address range used for coherent and non-coherent DRAM, MMIO, CSR.
- Physical Address (PA) - This is the address field seen on Intel® QPI. (Differentiates the virtual address seen on PCIe with Intel® VT-d and in the virtual address seen in processor cores).

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

2.5.5.1 NodeID Generation

This section contains an overview of how 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 64 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)
3. NodeID[5:0] is directly assigned from the TargetID

2.5.5.2 Memory Decoder

A single Decoder entry defines a contiguous memory range. Low order address interleaving is provided to distribute this range across up to two home agents. All ranges must be non-overlapping and aligned to 64 MB.

A miss of the SAD results in an error. Outbound snoops are dropped. Inbound requests return an unsupported request response. Protection of address ranges from inbound requests is done in range decoding prior to the SAD or can be done using holes in the SAD memory mapping if the range is aligned to 64 MB.

Note: The memory source address decoder in IIO contains no attribute, unlike the processor SAD. All attribute decode (MMIO, memory, Non-Coherent memory) is done with coarse range decoding prior to the request reaching the Source Address Decoder. See Chapter 6.0, “System Address Map” for details on the coarse address decode ranges.

Table 64. 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 QPIPINT: Intel® QPI Protocol 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 >0x4 - Reserved
TargetID List	48	A list of eight 6-bit TargetID values. Only two Home Node IDs are supported.

2.5.5.3 I/O Decoder

The MMIOL and MMIOH regions use standard memory decoders. The I/O decoder contains a number of special regions as shown below.

Table 65. I/O Decoder Entries

Field	Type	Address Base/Range	Size (Bytes)	attr	Interleave	CSR Register	Comments
VGA/CSeg	Memory	000A_0000	128K	MMIO	None	QPIPVSAD	Space can be disabled.
LocalxAPIC	Memory	FEE0_0000	1M	IPI	8 deep table	QPIPAPICSAD	Which bits of address select the table entry is variable.
Special Requests Targeting DMI.	Memory	N/A	N/A	Varies	None	QPIPSUBSAD	Peer-to-peer between PCie and DMI is not supported.
DCA Tag	NodeID	0	64	NodeID	Direct Mapping	QPIPDCASAD	Direct mapping modes for tag to NodeID.

2.5.5.3.1 APIC ID Decode

APIC ID decode is used to determine the Target NodeID for non-broadcast interrupts. Three bits of the APIC ID is used to select from eight targets. Selection of the APIC ID bits is dependent on the processor, so modes exist within the APIC ID decode register to select the appropriate bits. The bits that are used is also dependent on the type of interrupt (physical or extend logical).

2.5.5.3.2 Subtractive Decode

Requests that are subtractively decoded are sent to the legacy DMI port. When the legacy DMI port is located on a remote IIO over Intel® QPI this decoder simply specifies the NodeID of the peer legacy IIO 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 results in an error.

2.5.5.3.3 DCA Tag

DCA enabled writes result in a PrefetchHint message on Intel® QPI that is sent to a caching agent on Intel® QPI. The NodeID of the caching agent is determined by the PCI Express tag. The IIO supports a number of modes for which tag bits correspond to which NodeID bits. The tag bits may also translate to cache target information in the packet.

2.5.6 Special Response Status

Intel® QPI includes two response types: normal and failed. Normal is the default. Failed is discussed in this section.

On receiving a failed response status the IIO continues to process the request in the standard manner, but the failed status is forwarded with the completion. This response status is also logged as an error.

The IIO sends a failed response to Intel® QPI for some failed response types from PCI Express.

2.5.7 Illegal Completion/Response/Request

IIO explicitly checks all transaction for compliance to the request-response. If an illegal response is detected it is logged and the illegal packet is dropped.

2.5.8 Inbound Coherent

The IIO only sends a subset of the coherent transactions supported in Intel® QPI. This section describes 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.

The IIO supports only source broadcast snooping, with the Invalidating Write Back flow. In source broadcast mode, the IIO sends a snoop to all peer caching (it does not send a snoop to peer IIO caching agent) participants when initiating a new coherent request. The snoop is sent to non-Home node (CPU) only in DP systems. Which peer caching agents are snooped is determined by the snoop participant list. The snoop participant list comprises a list of NodeIDs which must receive snoops for a given coherent request. The IIO's NodeID is masked from the snoop participant list to prevent a snoop being sent back to the IIO", and following "The snoop participant list is programmed in QPIPSB.

2.5.9 Inbound Non-Coherent

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

There are three non-coherent broadcast lists

- The primary list is the "non-coherent broadcast list" which is used for power management, and Broadcast Interrupts. This list will be programmed to include all processors.
- The Lock Arbiter list of IIOs
- The Lock Arbiter list of processors

The broadcast lists are implemented with an 8-bit vector corresponding to NodeIDs 0-7. Each bit in this vector corresponds to a destination NodeID receiving the broadcast.

The Transaction ID (TID) allocation scheme used by the IIO results in a unique TID for each non-coherent request that is broadcast (i.e. for each broadcast interrupt, each request will use a unique TID). See Section 2.5.13 for additional details on the TID allocation.

Broadcast to the IIO's local NodeID will only be spawned internally and do not appear on the Intel® QPI bus.

2.5.9.1 Peer-to-Peer Tunneling

The IIO supports peer-to-peer tunneling of MRd, MWr, CplD, Cpl, MsgD and Msg PCI Express messages. Peer-to-peer traffic between PCIe and DMI is not supported.

2.5.10 Profile Support

The IIO will support UP and DP profiles set through configuration registers. Table 66 defines which register settings are required for each profile. There is not a single register setting for a given profile, but rather a set of registers that must be programmed to match Table 66.

Table 66. Profile Control

Feature	Register	Attribute	UP Profile	DP Profile	Notes
Source Address decoder enable	QPIPCTRL	RW	disable	enable	In UP profile all inbound requests are sent to a single target NodeID.
Address bits	QPIPMADDATA	RW	<=40 bits [39:0]		Can be reduced from the max to match a processor's support.
NodeID width	QPIPCTRL	RO	3-bit		Other NodeID bits will be set to zero, and will be interpreted as zero when received.
Remote P2P	<I/O SAD>[1]	RO	disable		All IO Decoder entries (except LocalxAPIC) will be disabled in DP Profile. See Table 65 for details.
Poison	QPIPCTRL	RW	<prog>	enable	When disabled any uncorrectable data error will be treated identically to a header parity.
Snoop Protocol	QPIPSB	RW	0x0h	0x6h[2]	The snoop vector controls which agents the IIO neeeds to broadcast snoops to.

1. See Table 65 for details on which registers are affected.
2. This value needs to be programmed in both IIOs.

2.5.11 Write Cache

The IIO 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.

2.5.11.1 Write Cache Depth

The write cache size is 72 entries.

2.5.11.2 Coherent Write Flow

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

IIO will issue an RFO command on Intel® QPI when it finds the write cache in I-state. The "Invalidating Write" flow uses InvWbMtoI command. These requests return E-state with no data. Once all the I/O ordering requirements have been met, the promotion phase occurs and the state of the line becomes M.

In the case where a RFO hits an M-state line in the write cache, ownership is granted immediately with no request appearing on Intel® QPI. This state will be referred to as MG (M-state with RFO Granted). An RFO hitting E-state or MG-state in the write cache indicates that another write has already received an RFO completion.

2.5.11.3 Eviction Policy

On reaching M-state the Write cache will evict the line immediately if no conflict is found. If a subsequent RFO is pending in the conflict queue and it is the first RFO conflict for this M-state line, then that write is given ownership. This is only allowed for a single conflicting RFO, which restricts the write combining policy so that only 2 writes may combine. This combining policy can be disabled with a configuration bit such that each inbound write will result in a RFO-EWB flow on Intel® QPI.

2.5.12 Outgoing Request Buffer (ORB)

When an inbound request is issued onto Intel® QPI an ORB entry is allocated. This list keeps all pertinent information about the transaction header needed to complete the request. It also stores the cache line address for coherent transactions to allow conflict checking with snoops (used for conflict checking for other requests).

When a request is issued, a RTID (Requestor Transaction ID) is assigned based on NodeID.

The ORB depth is 64 entries.

2.5.13 Time-Out Counter

Each entry in the ORB is tagged with a time-out value when it is allocated; the time-out value is dependent on the transaction type. This separation allows for isolating a failing transaction when dependence exists between transactions. Table 67 shows the four time-out levels of transactions the IIO supports. Levels 2 and 6 are for transactions that the IIO 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 single 8-bit time-out counter that increments at a programmable rate. This rate is programmable via configuration registers to a timeout between 2^8 cycles (IIO core) and 2^36 cycles. The time-out counter can also be disabled.

For each supported level there is a configuration value that defines the number of counter transitions for a given level before that transaction times-out. This value will be referred to as the “level time-out”. It provides a range of possible time-out values based on the counter speed and the “level time-out” in this configuration register. Table 67 shows the possible values for each level at a given counter speed. The configuration values should be programmed to increase as the level increases to support longer time-out values for the higher levels.

The ORB time-out tag is assigned when the entry is allocated. The value is based on current counter value + level time-out + 1. This tag supports an equal number of bits to the counter (8-bits).

On each increment of the counter every ORB tag is checked to see if the value is equal to the value of the counter. If a match is found on a valid transaction then it logged as a time-out. A failed response status is then sent to the requesting south agent for non-posted requests, and all Intel® QPI structures will be cleared of this request.

Table 67. Time-Out Level Classification for IIO

Level	Request Type
1	WbMtoI
2	None
3	NcRd, NonSnpRd, NonSnpWr, RdCode, InvWbMtoI, NcP2PB, IntPhysical, IntLogical, NcMsgS-StartReq1, NcMsgB-StartReq2, PrefetchHint
4	NcWr, NcMsgB-VLW, NcMsgB-PmReq
5	NcMsgS-StopReq1, NcMsgS-StopReq2
6	None

2.6 PCI Express Interface

This section describes the PCI Express* interface capabilities of the processor. See the latest *PCI Express Base Specification, Revision 2.0* for PCI Express details.

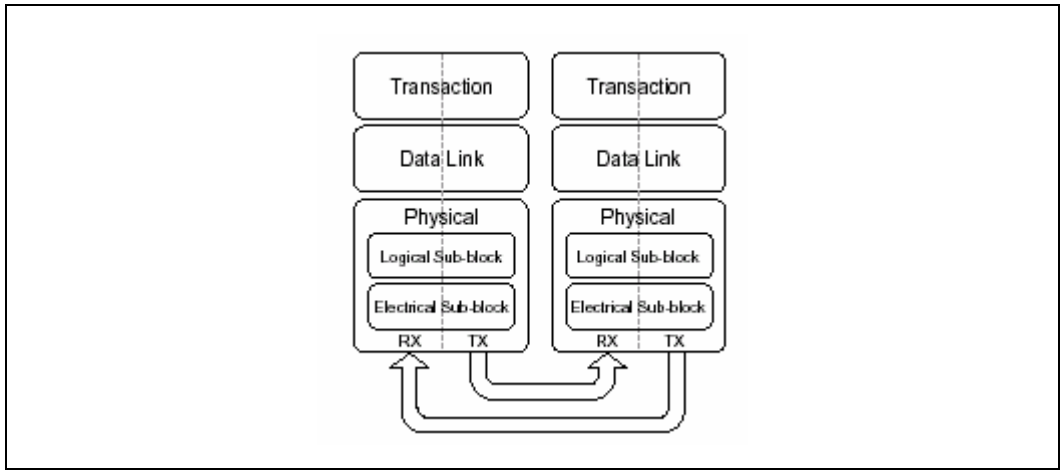
The processor has four PCI Express controllers, allowing the sixteen lanes to be controlled as a single x16 port, or two x8 ports, or one x8 port and two x4 ports, or four x4 ports.

2.6.1 PCI Express Architecture

PCI Express configuration uses standard mechanisms as defined in the PCI Plug-and-Play specification. The initial speed of 1.25 GHz results in 2.5 Gb/s per direction per lane. All processor PCI Express ports can negotiate between 2.5 GT/s and 5.0 GT/s speed per the inband mechanism defined in Gen2 PCI Express Specification. Note that the PCI Express port muxed with DMI will only support negotiation to 2.5 GT/s.

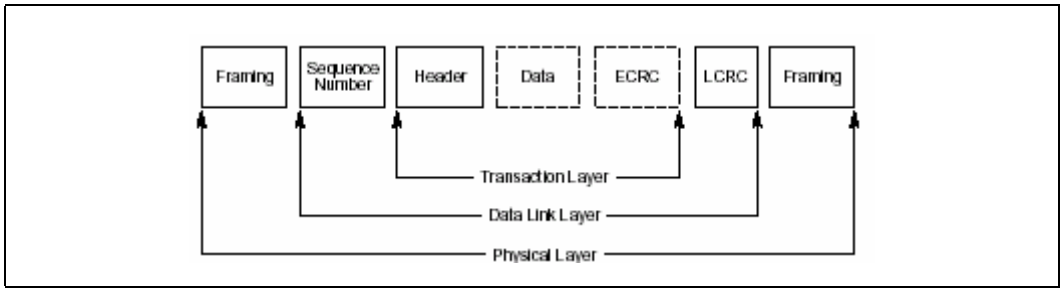
The PCI Express architecture is specified in three layers: Transaction Layer, Data Link Layer, and Physical Layer. The partitioning in the component is not necessarily along these same boundaries. See Figure 44.

Figure 44. PCI Express Layering Diagram



PCI Express uses packets to communicate information between components. Packets are formed in the Transaction and Data Link Layers to carry the information from the transmitting component to the receiving component. As the transmitted packets flow through the other layers, they are extended with additional information necessary to handle packets at those layers. At the receiving side the reverse process occurs and packets get transformed from their Physical Layer representation to the Data Link Layer representation and finally (for Transaction Layer Packets) to the form that can be processed by the Transaction Layer of the receiving device.

Figure 45. Packet Flow through the Layers



2.6.1.1 Transaction Layer

The upper layer of the PCI Express architecture is the Transaction Layer. The Transaction Layer's primary responsibility is the assembly and disassembly of Transaction Layer Packets (TLPs). TLPs are used to communicate transactions, such as read and write, as well as certain types of events. The Transaction Layer also manages flow control of TLPs.

2.6.1.2 Data Link Layer

The middle layer in the PCI Express stack, the Data Link Layer, serves as an intermediate stage between the Transaction Layer and the Physical Layer. Responsibilities of Data Link Layer include link management, error detection, and error correction.

The transmission side of the Data Link Layer accepts TLPs assembled by the Transaction Layer, calculates and applies data protection code and TLP sequence number, and submits them to Physical Layer for transmission across the Link. The receiving Data Link Layer is responsible for checking the integrity of received TLPs and for submitting them to the Transaction Layer for further processing. On detection of TLP error(s), this layer is responsible for requesting retransmission of TLPs until information is correctly received, or the Link is determined to have failed. The Data Link Layer also generates and consumes packets that are used for Link management functions.

2.6.1.3 Physical Layer

The Physical Layer includes all circuitry for interface operation, including driver and input buffers, parallel-to-serial and serial-to-parallel conversion, PLL(s), and impedance matching circuitry. It also includes logical functions related to interface initialization and maintenance. The Physical Layer exchanges data with the Data Link Layer in an implementation-specific format, and is responsible for converting this to an appropriate serialized format and transmitting it across the PCI Express Link at a frequency and width compatible with the device connected to the other side of the link.

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

2.6.2.1 Link Training

The Intel® Xeon® processor C5500/C3500 series supports 16 physical PCI Express lanes that can be grouped into 1, 2 or 4 independent PCIe ports. The processor PCI Express port will support the following Link widths: x16, x8, x4, x2 and x1.

During link training, the processor will attempt link negotiation starting from the highest defined link width and ramp down to the nearest supported link width that passes negotiation. For example, when x16 support is defined the port will first attempt negotiation as a single x16. If that fails, an attempt is made to negotiate as a single x8 link. If that fails an attempt is made to negotiate as a single x4 link. If that fails an attempt is made to negotiate as a single x2 link and finally if that fails it will attempt to train as a single x1 link.

Each of the widths (x16, x8, x4) are trained in both the non-lane-reversed and lane-reversed modes. Widths of (x2 and x1) are considered degraded special cases of a x4 port and have limited lane reversal as defined in Section 2.6.2.5, “Lane Reversal” . For example, x16 link width is trained in both the non-lane-reversed and lane-reversed modes before training for a single x8 configuration is attempted by the IIO. A x1 link is the minimum required link width that must be supported per the *PCI Express Base Specification,* Revision 2.0.

2.6.2.2 Port Bifurcation

IIO port bifurcation support is available via different means:

- Using the hardware strap pins (PECFGSEL[2:0]) as shown in Table 68.
- Via BIOS by appropriately programming the PCIE_PRT0_BIF_CTL register.

2.6.2.3 Port Bifurcation via BIOS

When the BIOS needs to control port bifurcation, the hardware strap needs to be set to “Wait_on_BIOS”. This instructs the LTSSM to not train until the BIOS explicitly enables port bifurcation by programming the PCIE_IOU0_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 the 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 IIO. Software can initiate link retraining within a sub-port or even change the width of a sub-port (by programming the PCIE_PRT/DMI_LANE_MSK register) any number of times without resetting the IIO.

The following is pseudo-code for how the register and strap work together to control port bifurcation. “Strap to ltssm” indicates the IIO internal strap to the Link Training and Status State Machine (LTSSM).

```
If (PCIE_IOU0_BIF_CTRL[2:0] == 111)
        If (<PECFGSEL[2:0]>!= 100 ) {
                Strap to ltssm = strap
        } else {
                Wait for “PCIE_IOU0_BIF_CTRL[3]” bit to be set
                Strap to ltssm = csr
        }
} else {
        Strap to ltssm = csr
}
```

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

Table 68. Link Width Strapping Options

PECFGSEL[2:0]	Behavior of PCIe Port
000	Reserved
001	Reserved
010	x4x4x8: Dev6(x4, lanes 15-12), Dev5(x4, lanes 11-8), Dev3(x8, lanes 7-0)
011	x8x4x4: Dev5(x8, lanes 15-8), Dev4(x4, lanes 7-4), Dev3(x4, lanes 3-0)
100	Wait-On-BIOS: optional when all RPs, must use if using NTB
101	x4x4x4x4: Dev6(x4, lanes 15-12), Dev5(x4, lanes 11-8), Dev4(x4, lanes 7-4), Dev3(x4, lanes 3-0)
110	x8x8: Dev5(x8, lanes 15-8), Dev3(x8, lanes 7-0)
111	x16: Dev3(x16, lanes 15-0)

2.6.2.4 Degraded Mode

Degraded mode is supported for x16, x8, and x4 link widths. Intel® Xeon® processor C5500/C3500 series supports degraded mode operation at half the original width, a quarter and an eighth of the original width or a x1. The IIO supported degradation modes are limited to the outer lanes only (including lane reversal). Lane degradation 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, the IIO attempts to negotiate at the link width that it is currnetly operating at and if that fails, the IIO attempts to negotiate a lower link width per the degraded mode operation.

Degraded modes are shown in Table 69 are supported. A higher width degraded mode will be attempted before trying any lower width degraded modes.

Table 69. Supported Degraded Modes in IIO

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 either lanes 7-4, 4-7, 3-0, 0-3
	x2 on either lanes 5-4, 4-5, 1-0, 0-1
	x1 on either lanes 0, 1, 2, 3, 4, 5, 6, 7

Table 69. Supported Degraded Modes in IIO

Original Link Width[1]	Degraded Mode Link Width and Lanes Numbers
x4	x2 on either lanes 1-0, 0-1
	x1 on either lanes 0, 1, 2, 3
x2	x1 on either lanes 0, 1

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

Entry into or exit from degraded mode are reported to software in the MISCCTRLSTS register, and also records which lane failed. Software can then report the flaky hardware behavior to the system operator for attention, by generating a system interrupt.

2.6.2.5 Lane Reversal

Lane reversal is supported on all PCI Express ports, regardless of the link width i.e. lane reversal works in x16, x8, and x4 link widths. See Table 69 for lane reversal combinations supported. A x2 card can be plugged into a x16, x8, or x4 slot and work as x2 only if lane-reversal is not done, otherwise it would operate in x1 mode.

2.6.3 Gen1/Gen2 Speed Selection

In general, Gen1 vs. Gen2 speed will be negotiated per the inband mechanism defined in the Gen2 PCI Express Specification. In addition, Gen2 speed can be prevented from negotiating if PE_GEN2_DISABLE# strap is set to 1 at reset deassertion. This strap controls all ports together. In addition the 'Target Link Speed' field in LNKCON2 register can be used by software to force a certain speed on the link.

2.6.4 Link Upconfigure Capability

Upconfigure is an optional *PCI Express Base Specification,* Revision 2.0 feature that allows the SW to increase or decrease the link width. Possible uses are for bandwidth matching and power savings.

The IIO supports link upconfigure capability. The IIO sends "1" during link training in Configuration state, in bit 6 of symbol 4 of a TS2 to indicate this capability when the upcfgcpable bit is set.

2.6.5 Error Reporting

PCI Express reports many error conditions through explicit error messages: ERR_COR, ERR_NONFATAL, ERR_FATAL. One of the following can be programmed when one of these error messages is received: (See "PCICMD: PCI Command" and "MSIXMSGCTL: MSI-X Messae Control" registers).

- Generate MSI
- Forward the messages to PCH

See 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.

2.6.5.1 Chipset-Specific Vendor-Defined

These vendor-defined messages are identified with a Vendor ID of 8086 in the message header and a specific message code. See the *Direct Media Interface Specification Rev 1.0* for details.

2.6.5.2 ASSERT_GPE / DEASSERT_GPE

General Purpose Event (GPE) consists of two messages: Assert_GPE and Deassert_GPE. Upon receipt of a Assert_GPE message from a PCI Express port, the IIO forwards the message to the PCH. When the GPE event has been serviced, the IIO will receive a Deassert_GPE message on the PCI Express port. At this point the IIO can send the deassert_GPE message on DMI.

2.6.6 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. The *PCI Express Base Specification, Revision 2.0* spec 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.

The following is a summary of when a configuration request will be re-issued:

- 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. The Configuration Retry Timeout timer is set via the “CTOCTRL: Completion Timeout Control” register. 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 coherent interface (except in the DP profile) at the expiry of every 48 ms from the time the request has been first sent on PCI Express till 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, IIO completes the transaction normally with a value of 0x01 in the data field and all 1s in any other bytes included in the read. See the *PCI Express Base Specification, Revision 2.0* for more details.

An Intel® Xeon® processor C5500/C3500 series-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.

2.6.7 Inbound Transactions

Inbound refers to the direction towards main memory from I/O.

2.6.7.1 Inbound PCI Express Messages Supported

Table 70 lists all inbound messages that may be received on a PCI Express downstream port (does not include DMI 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 70. Incoming PCI Express Message Cycles

PCI Express Transaction	Address Space or Message	IIO 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 DMI.
	ERR_COR ERR_NONFATAL ERR_FATAL	PCI Express error messages Propagate as an interrupt to system.
	PM_PME	Propagate as an interrupt/general purpose event to the system.
	PME_TO_ACK	Received PME_TO_ACK bit is set when IIO receives this message.
	PM_ENTER_L1 (DLLP)	Block subsequent TLP issue and wait for all pending TLPs to Ack. Then, send PM_REQUEST_ACK. See *PCI Express Base Specification*, Revision 2.0 for details of the L1 entry flow.
	ATC Invalidation Complete	When an end point device completes a ATC invalidation, it will send an Invalidate Complete message to the IIO (RC). This message will be tagged with information from the Invalidate message so that the IIO 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 hotplug event in PXH. Message forwarded to DMI port.
	MCTP	Management Control Transport Protocol messages - forwards MCTP messages received on its PCI-E ports to PCH over DMI interface.
	All Other Messages	Silently discard if message type is type 1 and drop and log error if message type is type 0

2.6.8 Outbound Transactions

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

2.6.8.1 Memory, I/O and Configuration Transactions Supported

Table 71 lists the possible outbound memory, I/O and configuration transactions.

Table 71. 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 PCI Express device.
	I/O	Legacy I/O write targeting PCI Express legacy device
	Configuration	Configuration write targeting PCI Express device.
Outbound Completions for Inbound Write Requests	I/O	Unsupported. Transaction will be returned as UR.
	Configuration (Type0 or Type1)	Unsupported. Transaction will be returned as UR.
Outbound Read Requests	Memory	Memory-mapped I/O read targeting PCI Express device.
	I/O	legacy I/O read targeting PCI Express device.
	Configuration	Configuration read targeting PCI Express device.
Outbound Completions for Inbound Read Requests	Memory	Response for an inbound read to main memory or a peer I/O device.
	I/O	Unsupported. Transaction will be returned as UR.
	Configuration (Type0 or Type1)	Unsupported. Transaction will be returned as UR.

2.6.9 Lock Support

For legacy PCI functionality, bus locks are supported through an explicit sequence of events. Intel® Xeon® processor C5500/C3500 series can receive a locked transaction sequence on the Intel® QuickPath Interconnect interface directed to a PCI Express port.

2.6.10 Outbound Messages Supported

Table 72 provides a list of all the messages supported as an initiator on a PCI Express port (DMI messages are not included in this table).

Table 72. 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 is set, send this message to the associated PCI Express port.
	PM_REQUEST_ACK (DLLP)	Acknowledges that the IIO received a PM_ENTER_L1 message. This message is continuously issued until the receiver link is idle. See the *PCI Express Base Specification, Revision 2.0* for details.
	PM_Active_State_Nak	When IIO receives a PM_Active_State_Request_L1.
	Set_Slot_Power_Limit	Message that is sent to PCI Express device when software wrote to the Slot Capabilities Register or the PCI Express link transitions to DL_Up state. See *PCI Express Base Specification, Revision 2.0* for more details.
	ATC Translation Invalidate	When a translation is changed in the TA and that translation might be contained within an ATC in an endpoint, the host system must send an invalidation to the ATC via IIO to maintain proper synchronization between the translation tables and the translation caches.
Intel Chipset-specifc Vendor-defined	EOI	End-of-interrupt cycle received on Intel® QPI. IIO broadcasts this message to all downstream PCI Express and DMI ports that have an I/OxAPIC below them.

2.6.10.1 Unlock

This message is transmitted by IIO at the end of a lock sequence. This message is transmitted irrespective of whether PCI Express lock was established or not and also regardless of whether the lock sequence terminated in an error or not.

2.6.10.2 EOI

EOI messages will be broadcast from the coherent interface to all the PCI Express interfaces/DMI ports that have an APIC below them. Presence of an APIC is indicated by the EOI enable bit in the MISCCTRLSTS: Misc Control and Status Register. This ensures that the appropriate interrupt controller receives the end-of-interrupt.

The IIO has the capability to NOT broadcast/multicast EOI message to any of the PCI Express/DMI ports and this is controlled via bit 0 in the EOI_CTRL register. When this bit is set, IIO simply drops the EOI message received from Intel® QPI and not send it to any south agent. But IIO does send a normal compare for the message on Intel® QPI.

2.6.11 32/64 bit Addressing

For inbound and outbound memory reads and writes, the IIO supports the 64-bit address format. If an outbound transaction's address is less than 4 GB, then the IIO will issue the transaction with a 32-bit addressing format on PCI Express. Only when the address is greater than 4 GB then IIO will initiate transaction with 64-bit addressing format.

2.6.12 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

2.6.12.1 Transaction ID

The Transaction ID uniquely identifies every transaction in the system. The Transaction ID comprises four sub-fields described in Table 73. This table provides details on how this field in the Express header is populated by IIO.

Table 73. PCI Express Transaction ID Handling

Field	Definition	IIO as Requester	IIO as Completer
Bus Number	Specifies the bus number that the requester resides on.	The IIO fills this field with the internal Bus Number that the PCI Express cluster resides on.	The IIO preserves this field from the request and copies it into the completion.
Device Number	Specifies the device number of the requester. NOTE: Normally, The 5-bit Device ID is required to be zero in the RID that consists of BDF, but when ARI is enabled, the 8-bit DF is now interpreted as an 8-bit Function Number with the Device Number equal to zero implied.	For CPU requests, the IIO fills this field with the Device Number that the PCI Express cluster owns. For DMA requests, the IIO fills this field with the device number of the DMA engine (Device#10)	
Function Number	Specifies the function number of the requester.	The IIO fills this field in with its Function Number that the PCI Express cluster owns (zero).	
Tag	Identifies 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.	NP tx: The IIO fills this field in with a value such that every pending request carries a unique Tag. NP Tag[7:5]=Intel® QPI Source NodeID[4:2]. Bits 7:5 can be non-zero only when 8-bit tag usage is enabled. Otherwise, IIO always zeros out 7:5. NP Tag[4:0]=Any algorithm that guarantees uniqueness across all pending NP requests from the port. P Tx: No uniqueness guaranteed. Tag[7:0]=Intel® QPI Source NodeID[7:0] for CPU requests. Bits 7:5 can be non-zero only when 8-bit tag usage is enabled. Otherwise, IIO always zeros out 7:5.	

2.6.12.2 Attributes

PCI Express supports two attribute hints described in Table 74. This table describes how these attribute fields are populated for requests and completions.

Table 74. PCI Express Attribute Handling

Attribute	Definition	IIO as Requester	IIO 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 for transactions generated on PCIe on behalf of a Intel® QPI request. On peer-to-peer requests, IIO forwards this attribute as-is.	This field is preserved 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.		

2.6.12.3 Traffic Class

The IIO does not optimize based on traffic class. IIO can receive a packet with TC != 0 and treat the packet as if it were TC = 0 from an ordering perspective. IIO forwards the TC filled as is on p2p requests and also returns the TC field from the original request on the completion packet sent back to the device.

2.6.13 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 75.

Table 75. PCI Express CompleterID Handling

Field	Definition	IIO as Completer
Bus Number	Specifies the bus number that the completer resides on.	The IIO fills this field in with its internal Bus Number that the PCI Express cluster resides on.
Device Number	Specifies the device number of the completer.	Device number of the root port sending the completion back to PCIe.
Function Number	Specifies the function number of the completer.	0

2.6.14 Miscellaneous

2.6.14.1 Number of Outbound Non-posted Requests

The x4 PCI Express interface supports up to 16 outstanding non-posted transactions outbound comprising transactions issued by the processors. x8 supports 32 and x16 supports 64.

2.6.14.2 MSIs Generated from Root Ports and Locks

Once lock has been established on the coherent interface, IIO cannot send any requests on the coherent interface, and this includes MSI transactions generated from the root port of the PCIe port that is locked. This requirement imposes that the MSI's from the root port should not block (locked read) completions from the PCI Express port moving to the coherent interface.

2.6.14.3 Completions for Locked Read Requests

LkRdCmp and RdCmp are aliased -i.e. either of these completion types can terminate a locked/non-locked read request.

2.6.15 PCI Express RAS

The PCI Express Advanced Error Reporting (AER) capability is supported. See the *PCI Express Base Specification, Revision 2.0* for details.

2.6.16 ECRC Support

ECRC is not supported. ECRC is ignored and dropped on all incoming packets and is not generated on any outgoing packet.

2.6.17 Completion Timeout

For all non-posted requests issued on PCI Express/DMI, a timer is maintained that tracks the max completion time for that request.

The OS selects a coarse range for the timeout value. The timeout value is programmable from 10 ms all the way up to 64s. See the DEVCAP2: PCI Express Device Capabilities register for additional control that provides for the 17s to 64s timeout range.

See Section 11.0, “Reliability, Availability, Serviceability (RAS)” for details of responses returned by IIO to various interfaces on a completion timeout event. AER-required error logging and escalation happen as well. In addition to the AER error logging, IIO also sets the locked read timeout bit in “MISCCTRLSTS: Misc Control and Status Registers”, if the completion timeout happened on a locked read request.

2.6.18 Data Poisoning

The IIO supports forwarding poisoned information between Intel® QPI and PCI Express and vice-versa. The IIO also supports forwarding poisoned data between peer PCI Express ports.

The IIO has a mode in which poisoned data is never sent out on PCI Express i.e. any packet with poisoned data is dropped internally in the IIO and an error escalation done.

2.6.19 Role-Based Error Reporting

The role-based error reporting that is specified in the *PCI Express Base Specification, Revision 2.0* spec is supported.

A Poisoned TLP that IIO receives on peer-to-peer packets is treated as an advisory non-fatal error condition i.e. ERR_COR signaled and poisoned information propagated peer-to-peer. Poisoned TLP that is received on packets that are destined towards DRAM memory or poisoned TLP packets that target the interrupt address range, are forwarded to the coherent interface with the poison bit set, provided the coherent interface is enabled to set the poisoned bit via QPIPC[12] bit. 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, then the received poisoned TLP condition is treated as a normal non-fatal error. The packet would be dropped if it is a posted transaction. A “master abort” response is sent on the coherent interface if a poisoned TLP is received for an outstanding non-posted request.

When a transaction times out or receives a UR/CA response on a request outstanding on PCI Express, recovery in hardware is not attempted. UR/CA received does not cause any error escalation via AER mechanism and cause error escalation. A completion timeout condition is treated as a normal non-fatal error condition (and not as an advisory condition). An unexpected completion received from PCI Express port is treated as an advisory non-fatal error if the severity of it is set to non-fatal. If the severity is set to fatal, then unexpected completions are NOT treated as advisory but as fatal.

2.6.20 Data Link Layer Specifics

2.6.20.1 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”. See the *PCI Express Base Specification,* Revision 2.0 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 IIO 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 Acked with a single Ack DLLP. The timer is programmable.

2.6.20.2 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 32b CRC (using the polynomial defined in the *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.

2.6.21 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. See the *PCI Express Base Specification, Revision 2.0* for details on this mechanism.

2.6.22 Flow Control

The PCI Express flow control types are described in Table 76.

Table 76. PCI Express Credit Mapping for Inbound Requests

Flow Control Type	Definition	Initial IIO 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.	16(x4) 32(x8) 64(x16)
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.	80(x4) 160(x8) 320(x16)
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.	18(x4) 36(x8) 72(x16)
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.	4
Completion Header Credits (CPH)	Tracks the number of completion headers the agent is capable of supporting.	infinite advertized 64 physical
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 advertized 16 physical

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 the *PCI Express Base Specification, Revision 2.0* .

Note: The credit advertisement in Table 76 does not necessarily imply the number of outstanding requests to memory.

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

The credit advertisement for downstream requests are described in Table 77.

Table 77. PCI Express Credit Mapping for Outbound Requests (Sheet 1 of 2)

Flow Control Type	Definition	Initial IIO 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.	4 (x4) 8 (x8) 16 (x16)
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.	8 (x4) 16 (x8) 32 (x16)
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.	4 (x4) 8 (x8) 16 (x16)

Table 77. PCI Express Credit Mapping for Outbound Requests (Sheet 2 of 2)

Flow Control Type	Definition	Initial IIO Advertisement
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.	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)

2.6.22.1 Flow Control Credit Return by IIO

After reset, credit information is initialized with the values indicated in Table 76 by following the flow control initialization protocol defined in the *PCI Express Base Specification, Revision 2.0* . Since the IIO supports only VC0, only this channel is initialized. As a receiver, the IIO 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 Update_FCs for that credit class, as per spec.
- For non-infinite credits advertised, we have a long timer running that will send Update_FCs if none was sent in the past say 28 usec (to comply with the spec's 30 usec rule). This 28 us is programmable to 6 us.
- If and only when there is credits to be released, IIO will wait for a configurable/ programmable number of cycles (in the order of 30-70 cycles) before update_FC is sent. This will be 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.

2.6.22.2 FC Update DLLP Timeout

The optional flow control update DLLP timeout timer is supported.

2.6.23 Physical Layer Specifics

2.6.23.1 Polarity Inversion

The PCI Express Base Specification, Version 0.9 of Revision 2.0 defines a concept called polarity inversion. Polarity inversion allows the board designer to connect the D+ and D- lines incorrectly between devices. Polarity inversion is supported.

2.6.24 Non-Transparent Bridge

PCI Express non-transparent bridge (NTB) acts as a gateway that enables high performance, low overhead communication between two intelligent subsystems, the local and the remote subsystems. The NTB allows a local processor to independently configure and control the local subsystem, provides isolation of the local host memory domain from the remote host memory domain while enabling status and data exchange between the two domains.

See "PCI Express Non-Transparent Bridge" for more information on the NTB.

2.7 Direct Media Interface (DMI2)

The Direct Media Interface in the IIO is responsible for sending and receiving packets/ commands to the PCH. The DMI is an extension of the standard PCI Express specification with special commands/features added to mimic the legacy Hub Interface. DMI2 is the second generation extension of DMI. See the DMI Specification, Revision 2.0, for more DMI2 details.

Note: Other references to DMI are referring to the same DMI2-compliant interface described above.

DMI connects the processor and the PCH chip-to-chip. DMI2 is supported. The DMI is similar to a four-lane PCI Express interface supporting up to 1 GB/s of bandwidth in each direction. Only DMI x4 configuration is supported.

In DP configurations, the DMI port of the non-legacy processor may be configured as a a single PCIe port, supporting PCIe Gen1 only.

2.7.1 DMI Error Flow

DMI can only generate SERR in response to errors, never SCI, SMI, MSI, PCI INT, or GPE. Any DMI related SERR activity is associated with Device 0.

2.7.2 Processor/PCH Compatibility Assumptions

The Intel® Xeon® processor C5500/C3500 series is compatible with the PCH and is not compatible with any previous (G)MCH or ICH products.

2.7.3 DMI Link Down

The DMI link going down is a fatal, unrecoverable error. If the DMI data link goes to data link down, after the link was up, then the DMI link hangs the system by not allowing the link to retrain to prevent data corruption. This is controlled by the PCH.

Downstream transactions that had been successfully transmitted across the link prior to the link going down may be processed as normal. No completions from downstream, non-posted transactions are returned upstream over the DMI link after a link down event.

§ §

3.0 PCI Express Non-Transparent Bridge

3.1 Introduction

PCI Express* non-transparent bridge (NTB) acts as a gateway that enables high performance, low overhead communication between two intelligent subsystems, the local and the remote subsystems. The NTB allows a local processor to independently configure and control the local subsystem, provides isolation of the local host memory domain from the remote host memory domain while enabling status and data exchange between the two domains.

When used in conjunction with Intel® VT-d2 both primary and secondary addresses are guest addresses. When Intel® VT-d2 is not used the secondary side of the bridge is a guest address and the primary side of the bridge is a physical address.

3.2 NTB Features Supported on Intel® Xeon® Processor C5500/C3500 Series

The Intel® Xeon® processor C5500/C3500 series supports the following NTB features. Details are specified in the subsequent sections of this document.

- PCIE Port 0 can be configured to be either a transparent bridge (TB) or an NTB.
  - NTB link width can support x4 or x8
- The NTB port supports Gen1 and Gen2 speed.
- The NTB supports two usage models
  - NTB attached to a Root Port (RP)
  - NTB attached to another NTB
- Supports 3 64b BARs
  - BAR 0/1 for configuration space
  - BAR 2/3 and BAR 4/5 are prefetchable memory windows that can access both 32b and 64b address space through 64 bit BARs.
  - BAR 2/3 and 4/5 support direct address translation
  - BAR 2/3 and 4/5 support limit registers
    - Limit registers can be used to limit the size of a memory window to less than the size specified in the PCI BAR. PCI BAR sizes are always a power of 2, e.g. 4GB, 8GB, 16GB. The limit registers allow the user to select any value to a 4KB resolution within any window defined by the PCI BAR. For example if the PCI BAR defines 8GB region the limit register could be used to limit that region to 6GB.
    - One use case for limit registers also provide a mechanism to allow separation of code space from data space.
- Supports posted writes and non-posted memory read transactions across NTB.

- Supports peer-to-peer transactions upstream and downstream across NTB. Capabilities for NTB are the same as defined for PCIE ports. See Section 3.7, "NTB Inbound Transactions" and Section 3.8, "Outbound Transactions" for details.
- Supports sixteen, 32-bit scratch pad registers, (total 64B) that are accessible through the BAR0 configuration space.
- Supports two, 16-bit doorbell registers (PDOORBELL and SDOORBELL) that are accessible through the BAR0 configuration space.
- Supports INTx, MSI and MSI-X mechanism for interrupts on both sides of the NTB in the upstream direction only.
  - For example a write to the PDOORBELL from the link partner attached to the secondary side of the NTB will result in a INTx, MSI or MSI-X in the upstream direction to the local Intel® Xeon® processor C5500/C3500 series.
  - A write from the local host on the Intel® Xeon® processor C5500/C3500 series to the SDOORBELL will result in a INTx, MSI or MSI-X in the upstream direction to the link partner connected to the secondary side of the NTB.
- Capability for passing doorbell/scratchpad across back-to-back NTB configuration.

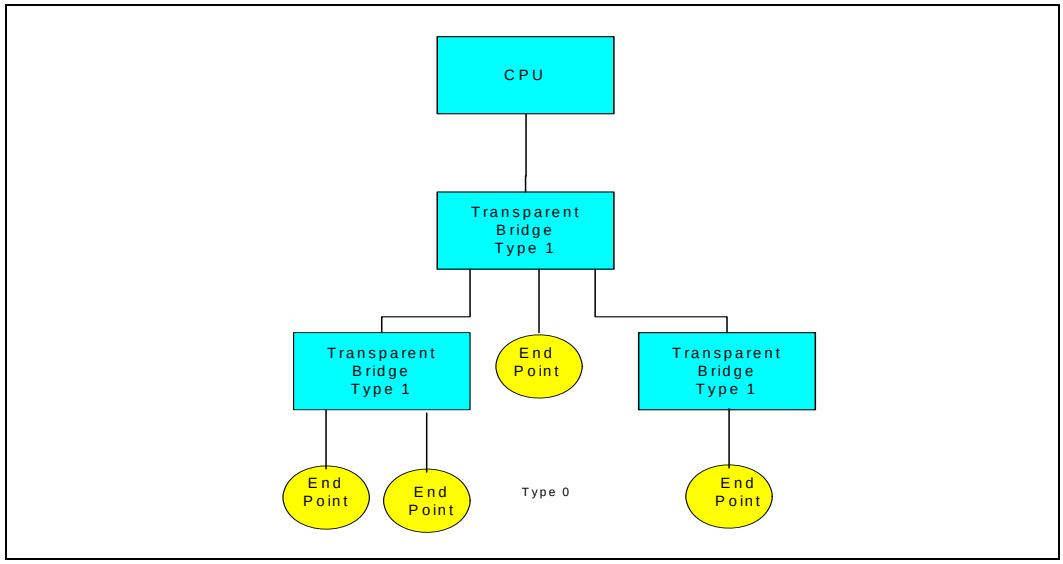
3.2.1 Features Not Supported on the Intel® Xeon® Processor C5500/C3500 Series NTB

- NTB does not support x16 link configuration
- NTB does not support IO space BARs
- NTB does not support vendor defined PCIE message transactions. These messages are silently dropped if received.

3.3 Non-Transparent Bridge vs. Transparent Bridge

A PCIE TB provides electrical isolation and enables design expansion for the host I/O subsystem. The host processor enumerates the entire system through discovery of TBs and Endpoint devices. The presence of a TB between the host and an Endpoint device is transparent to the device and the device driver associated with that device. The Intel® Xeon® processor C5500/C3500 series TB does not require a device driver of its own as it does not have any resources that must be managed by software during run time. The TB exposes Control and Status Register with Type 1 header, informing the host processor to continue enumeration beyond the bridge until it discovers Endpoint devices downstream from the bridge. The Endpoint devices will support Configuration Registers with Type 0 header and terminate the enumeration process. Figure 46 shows a system with TBs and Endpoint devices.

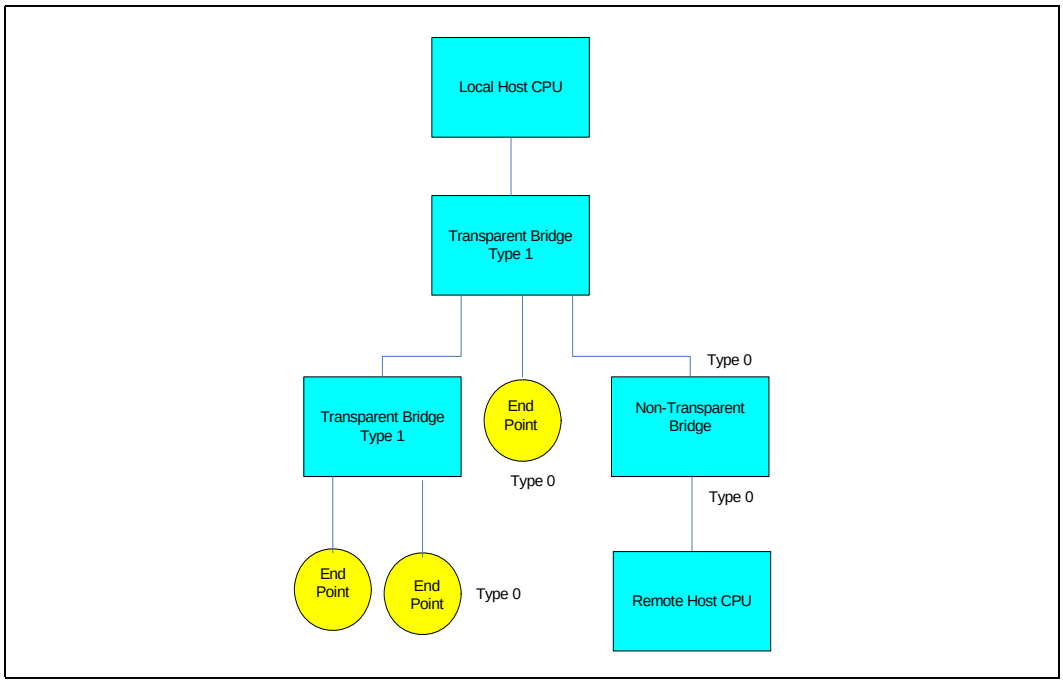
Figure 46. Enumeration in System with Transparent Bridges and Endpoint Devices



In contrast, a NTB provides logical isolation of resources in a system in addition to providing electrical isolation and system expansion capability. The NTB connects a local subsystem to a remote subsystem and provides isolation of memory space between the two subsystems. The local host discovers and enumerates all local Endpoint devices connected to the system. The NTB is discovered by the local host as a Root Complex Integrated Endpoint (RCiEP), the NTB then exposes its CSRs with Type 0 header to the local host. The local host stops enumeration beyond the NTB and marks the NTB as a logical Endpoint in its memory space. Similarly, the remote host discovers and enumerates all the Endpoint devices connected to it (directly or through TBs). When the remote host discovers the NTB, the NTB exposes a CSR with Type 0 header on the remote interface as well. Thus the NTB functions as an Endpoint to both domains, terminates the enumeration process from each side and isolates the two domains from each other.

Figure 47 shows a system with a NTB. The NTB provides address translation for transactions that cross from one memory space to the other.

Figure 47. Non-Transparent Bridge Based Systems



3.4 NTB Support in Intel® Xeon® Processor C5500/C3500 Series

When using the NTB capability the Intel® Xeon® processor C5500/C3500 series will support the NTB functionality on port 0 only in either the 1x4 or 1x8 configuration. The NTB functionality is not supported in the single x16 port configuration.

The BIOS must enable the NTB function. In addition, a software configuration enable bit will provide the ability to enable or disable the NTB port .

3.5 NTB Supported Configurations

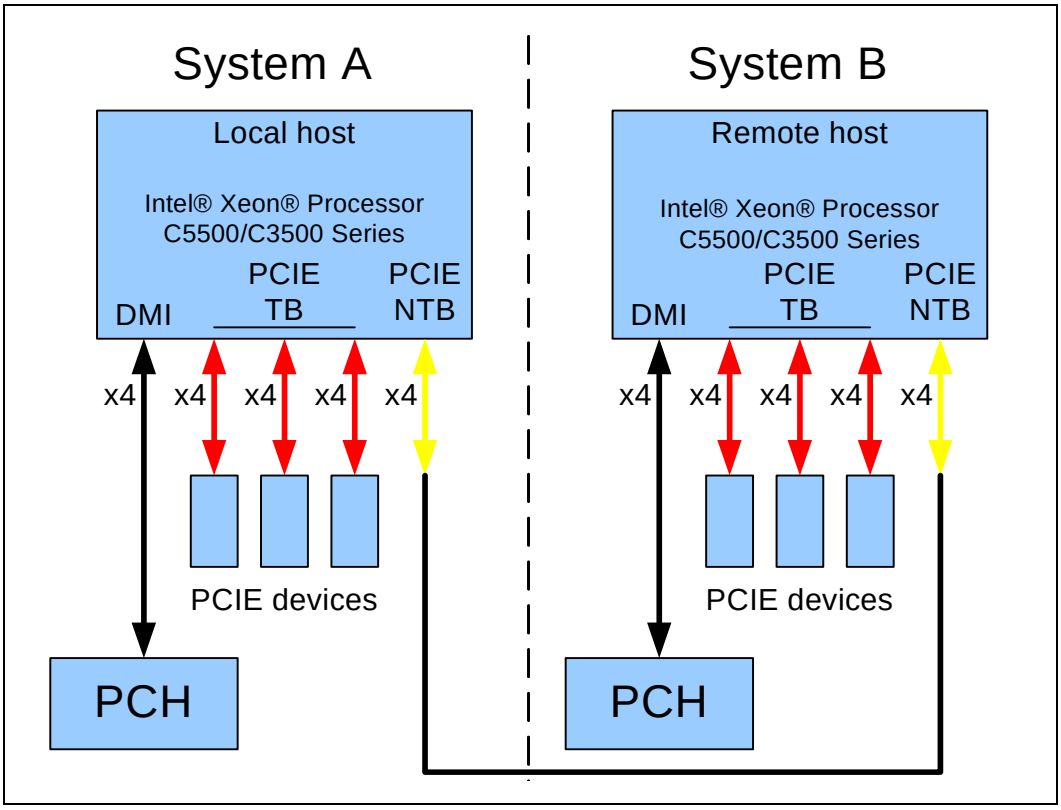
The following configurations are possible.

3.5.1 Connecting Intel® Xeon® Processor C5500/C3500 Series Systems Back-to-Back with NTB Ports

In this configuration, two Intel® Xeon® processor C5500/C3500 series UP systems are connected together through the NTB port of each system as shown in Figure 48. In the example each Intel® Xeon® processor C5500/C3500 series system supports one x4 PCIE port configured to be a NTB while the other three x4 ports are configured to be root ports. Each system is completely independent with its own reset domain.

Note: In this configuration, the NTB port can also be a x8 PCIE port.

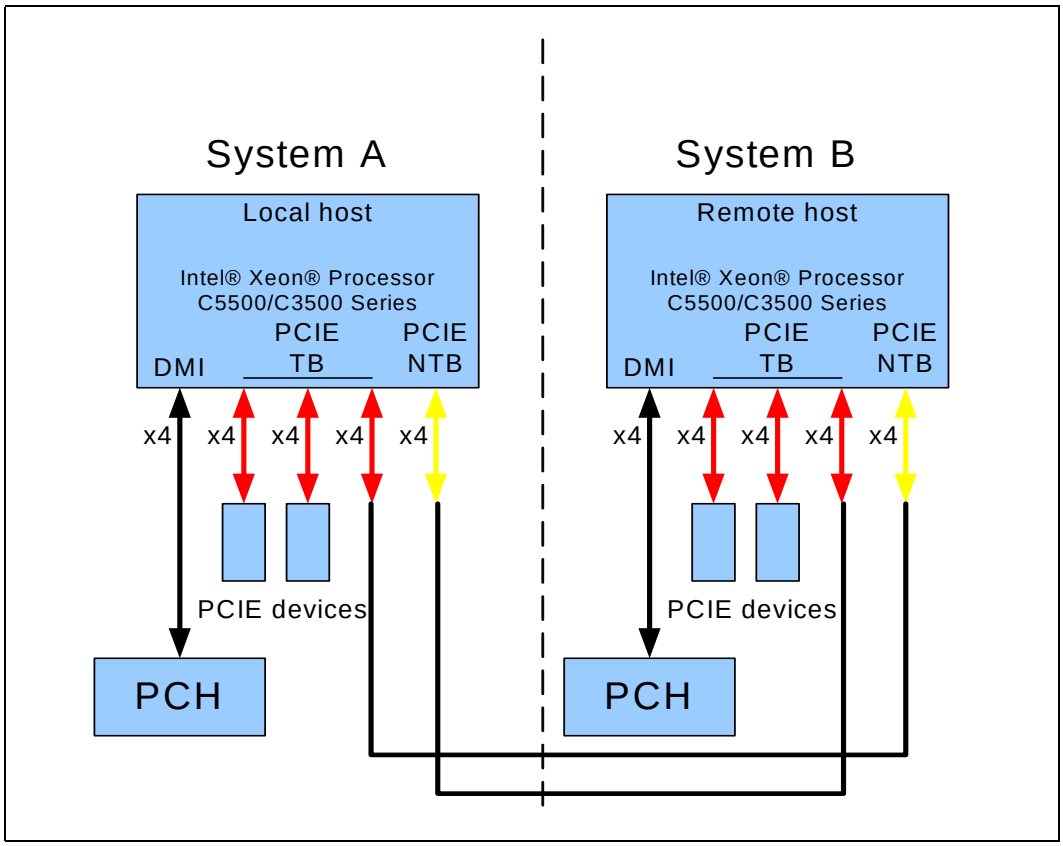
Figure 48. NTB Ports Connected Back-to-Back



3.5.2 Connecting NTB Port on Intel® Xeon® Processor C5500/C3500 Series to Root Port on Another Intel® Xeon® Processor C5500/C3500 Series System - Symmetric Configuration

In the configuration shown in Figure 49, the NTB port on one Intel® Xeon® processor C5500/C3500 series (the system on the left), is connected to the root port of the Intel® Xeon® processor C5500/C3500 series system on the right. The second system's NTB port is connected to the root port on the first system making this a fully symmetric configuration. This configuration provides full PCIE link redundancy between the two UP systems in addition to providing the NTB isolation. One limitation of this system is that two out of the four PCIE ports on the two Intel® Xeon® processor C5500/C3500 seriess are used for NTB interconnect, leaving only two other ports on each Intel® Xeon® processor C5500/C3500 series as generic PCIE root ports. The example is shown with x4 ports but the same is possible with x8 ports but leaving no other PCIE ports for attach points to the system.

Figure 49. NTB Port on Intel® Xeon® Processor C5500/C3500 Series Connected to Root Port - Symmetric Configuration



3.5.3 Connecting NTB Port on Intel® Xeon® Processor C5500/C3500 Series to Root Port on Another System - Non-Symmetric Configuration

In the configuration shown in Figure 50, the NTB port on one Intel® Xeon® processor C5500/C3500 series (the system on the left), is connected to the root port of the system on the right. Although the second system is shown as a Intel® Xeon® processor C5500/C3500 series system, it is not necessary for that system to be a Intel® Xeon® processor C5500/C3500 series system. Figure 51 shows a configuration where the NTB port on Intel® Xeon® processor C5500/C3500 series is connected to the root port of a non-Intel® Xeon® processor C5500/C3500 series system (any host that supports a PCIE root port). Hence this configuration is referred to as the non-symmetric usage model. Another valid configuration is to connect the root port on Intel® Xeon® processor C5500/C3500 series to an external non-Intel® Xeon® processor C5500/C3500 series NTB port on a device.

The non-symmetrical configuration has a more general usage model that allows the Intel® Xeon® processor C5500/C3500 series system to operate with another Intel® Xeon® processor C5500/C3500 series system through the NTB port on a single PCIE, or to operate with a non-Intel® Xeon® processor C5500/C3500 series system through the NTB port on Intel® Xeon® processor C5500/C3500 series or NTB port of the other device.

Figure 50. NTB Port on Intel® Xeon® Processor C5500/C3500 Series Connected to Root Port - Non-Symmetric

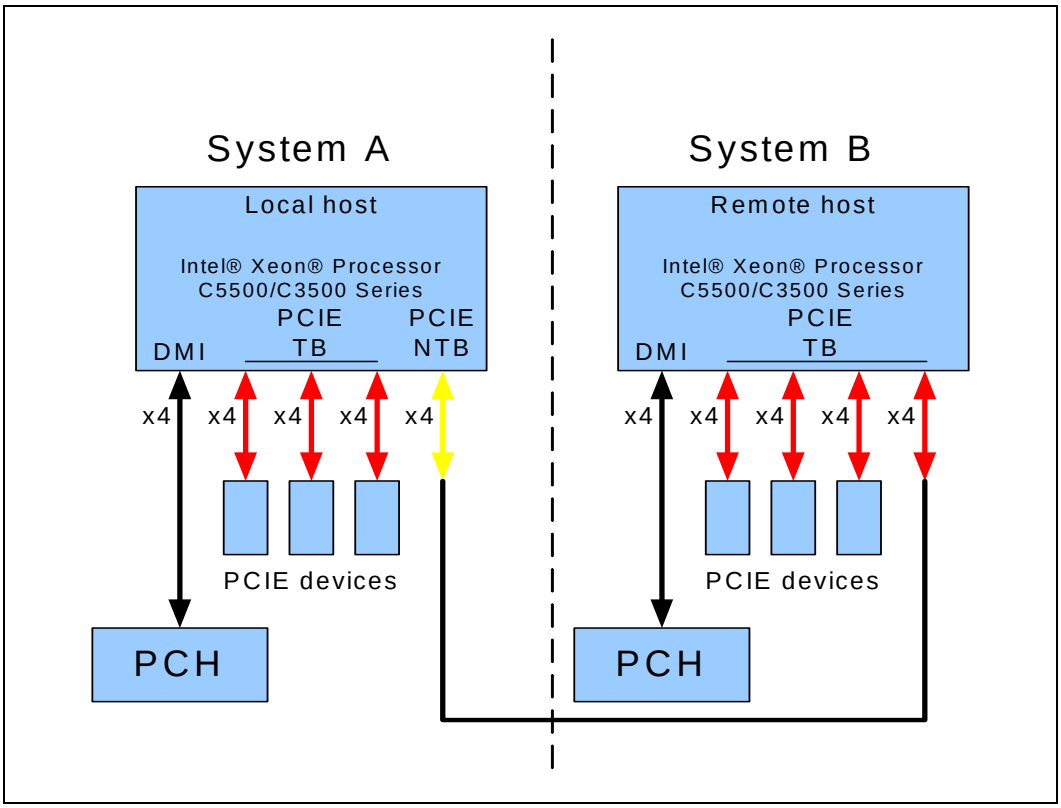
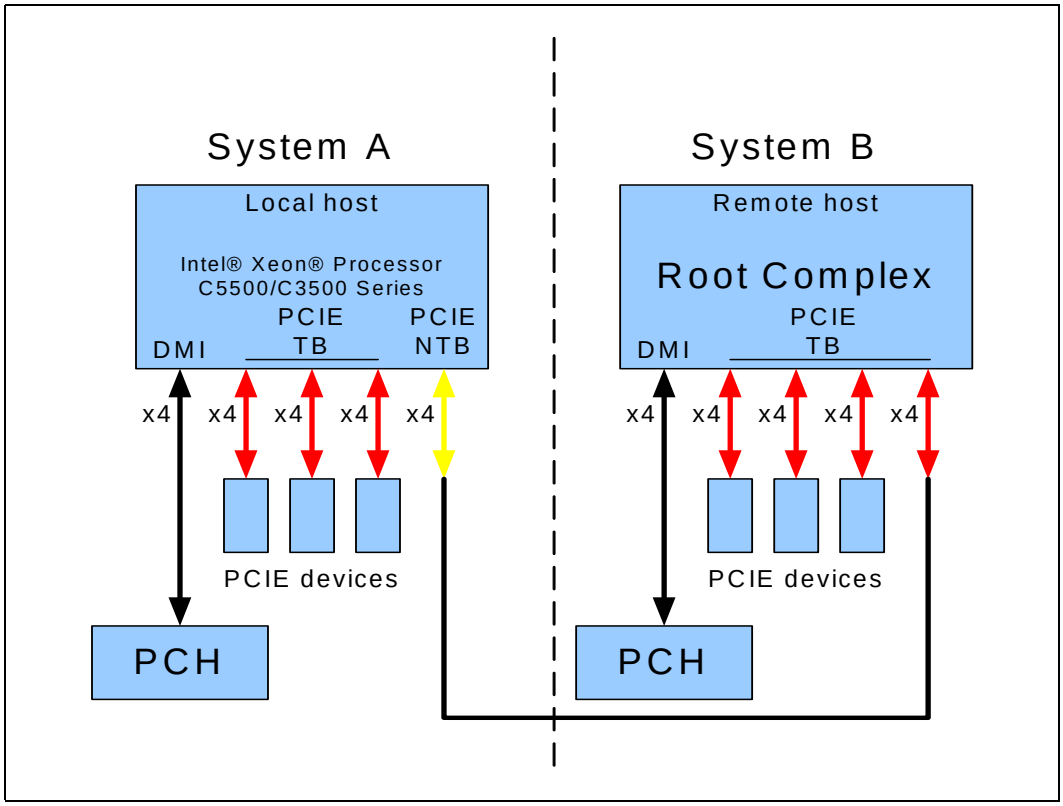


Figure 51. NTB Port Connected to Non-Intel® Xeon® Processor C5500/C3500 Series System - Non-Symmetric



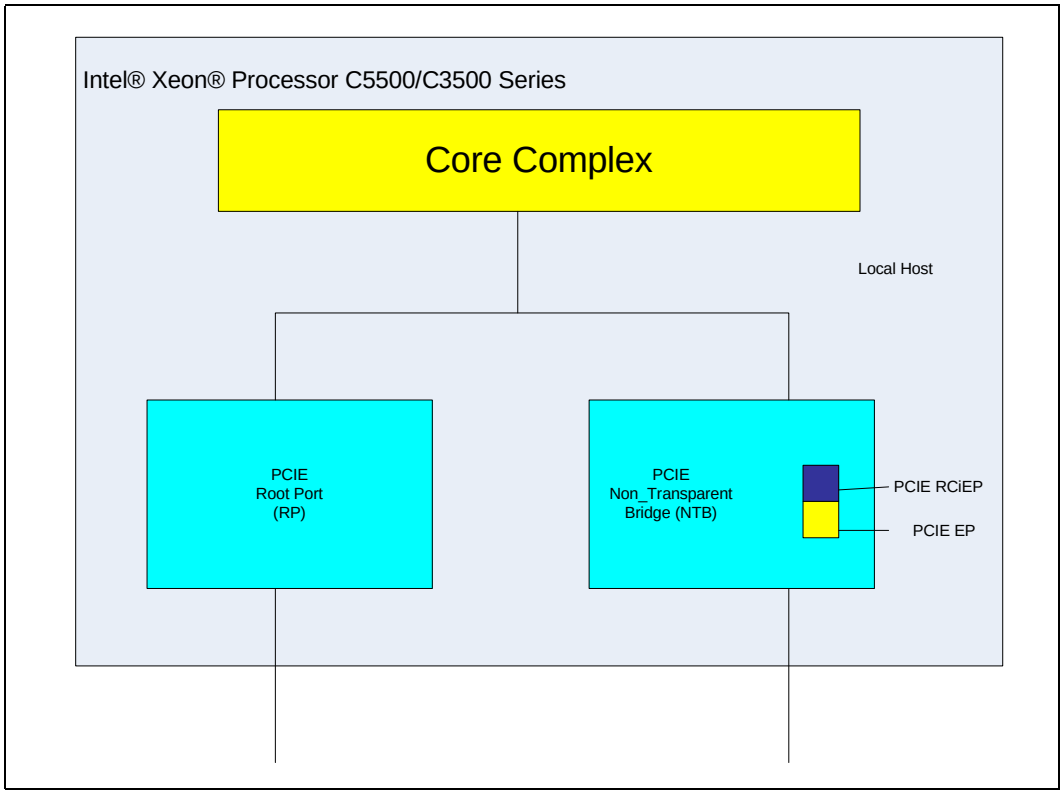
3.6 Architecture Overview

The NTB provides two interfaces and sets of configuration registers, one for each of the interfaces shown in Figure 52. The interface to the on-chip CPU complex is referred to as the local host interface. The external interface is referred to as the remote host interface. The NTB local host interface appears as a Root Complex Integrated Endpoint (RCiEP) to the local host and the NTB's remote interface appears as a PCI Express Endpoint to the remote host. Both sides expose Type 0 configuration header to discovery software, to both the local host and the remote host interface.

The NTB port supports the following sets of registers

- Type 0 configuration space registers with BAR definition on each side of the NTB.
- PCIE Capability Structure Configuration Registers registers with device capabilities.
- PCIE Capability Structure Configuration Registers registers with Device ID, Class Code and interface configuration with link layer attributes such as port width, max payload size etc.
- Configuration Shadowing – A set of registers present on each side of the NTB. Secondary side registers are visible to primary side. Primary side registers are not visible to the secondary side.
- Access Enable – A register is provided to enable blocking configuration register access from the secondary side of the NTB. See bit 0 in Section 3.21.1.12, "NTBCNTL: NTB Control" .
- Limit Registers– Limit registers can be used to limit the size of a memory window to less than the size specified in the PCI BAR. PCI BAR sizes are always a power of 2, e.g. 4GB, 8GB, 16GB. The limit registers allow the user to select any value to a 4KB resolution within any window defined by the PCI BAR. For example if the PCI BAR defines 8GB region the limit register could be used to limit that region to 6GB.
- Scratchpad – A set of 16, 32b registers used for inter-processor communication. These registers can be seen from both sides of the NTB.
- Doorbell – Two 16-bit doorbell registers (PDOORBELL and SDOORBELL) enabling each side of the NTB to interrupt the opposite side. There is one set on the primary side PDOORBELL and one set on the secondary side SDOORBELL.
- Semaphore – This is a single register that can be seen from both sides of the NTB. The semaphore register allows SW a mechanism of controlling write access into scratchpad. This semaphore has a "read 0 to set", "write 1 to clear" attribute and is visible from both sides of the NTB. This register is used for NTB/RP configuration.
- B2B Scratchpad – A set of 16, 32b registers used for inter-processor communication between two NTBs.
- B2B Doorbell – A 16-bit doorbell register (B2BDOORBELL) enabling interrupt passing between two NTBs.

Figure 52. Intel® Xeon® Processor C5500/C3500 Series NTB Port - Nomenclature



The NTB port supports the Type 0 configuration header. The first 10 DW of the Type 0 configuration header is shown in Table 78. The NTB sets the following parameters in the configuration header.

- The class code field is defined per PCI Specification Revision 3.0 and is set to 0x068000 as shown in Table 79.

Table 78. Type 0 Configuration Header for Local and Remote Interface

Byte3	Byte2	Byte1	Byte0	DW
Device ID		Vendor Id		00
Status Register		Command Register		01
Class code			Revision ID	02
BIST	Header Type	Latency Timer	Cache Line Size	03
Base Address 0				04
Base Address 1				05
Base Address 2				06
Base Address 3				07
BAse Address 4				08
Base Address 5				09

Table 79. Class Code

23:16	15:8	7:0
Class Code	Sub-Class Code	Programming Interface Byte
0x06 (bridge)	0x80 (other bridge type)	0x00

- Header type is set to type 0

The base address registers (BAR) specify the address decode functions that will be supported by the NTB.

- The Intel® Xeon® processor C5500/C3500 series NTB will support only 64b BARs. Intel® Xeon® processor C5500/C3500 series will not support 32b BARs.
- The Intel® Xeon® processor C5500/C3500 series NTB will support memory decode region only. Intel® Xeon® processor C5500/C3500 series will not support IO decode region.
  - Bit 0 in all Base Address registers is read-only and used to determine whether the register maps into Memory or I/O Space. Base Address registers that map to Memory Space must return a 0 in bit 0. Base Address registers that map to I/O Space must return a 1 in bit 0. The Intel® Xeon® processor C5500/C3500 series NTB only supports Memory Space so this bit is hard-coded to 0.
  - Bits [2:1] of each BAR indicate whether the decoder address is 32b (4GB memory space) or 64b (>4GB memory space)
    00 = Locate anywhere in 32-bit access space
    01 = Reserved
    10 = Locate anywhere in 64-bit access space
    11 = Reserved
    Intel® Xeon® processor C5500/C3500 series only supports 64b BARs so these bits will be hard-coded to “10”
  - Bit[3] of a memory BAR specifies whether the memory is prefetchable or not. 1=Prefetchable Memory
    0=Non-Prefetchable
- Primary side BAR 0/1 (internal side of the bridge) is a fixed 64KB prefetchable memory associated with MMIO space and will be used to map the 256B PCI configuration space of the secondary side, and the shared MMIO space of the NTB into the local host memory. Local host will have access to the configuration registers on primary side of the NTB, the shared MMIO space of the NTB, and the first 256B of the secondary side of the NTB through memory mapped IO transactions.

Note: BAR 0/1 Semaphore register has read side effects that must be properly handled by software

- Secondary side BAR 0/1 (external side of the bridge) is a fixed 32KB programmable as either prefetchable or non-prefetchable memory associated with configuration and MMIO space and will be used to map the configuration space of the secondary side and the shared MMIO space of the NTB into the remote host memory. The remote host will have access to the configuration registers on the secondary side of the NTB and the shared MMIO space of the NTB through memory mapped IO transactions. The remote host cannot see the configuration registers on the primary side of the bridge.
- BAR 2/3 and BAR 4/5 will provide two BARs for memory windows. These BARs will be for prefetchable memory only.
- Intel® Xeon® processor C5500/C3500 series will not support BARs for IO space.

Enumeration software can determine how much address space the device requires by writing a value of all 1's to the BAR and then reading the value back. Unimplemented Base Address registers are hardwired to zero.

The size of each BAR is determined based on the weight of the least significant bit that is writable in the BAR address bits b[63:7] for a 64b BAR. (The minimum memory address range defined in PCIE is 4KB). Table 80 shows the possible memory size that can be specified by the BAR.

Note: Programming a value of '0' or any other value other than (12-39) into any of the size registers (PBAR23SZ, PBAR45SZ, SBAR23SZ, SBAR45SZ)will result in the associated BAR being disabled.

Table 80. Memory Aperture Size Defined by BAR

least significant bit set to 1	Size of Memory Block
11	2KB
12	4KB
13	8KB
…	…
32	4GB
33	8GB
34	16GB
35	32GB
36	64GB
37	128GB
38	256GB
39	512GB

The NTB accepts only those configuration and memory transactions that are addressed to the bridge. It must return an unsupported request (UR) response to all other Configuration Register transactions.

3.6.1 "A Priori" Configuration Knowledge

The PCIE x4/x8 port 0 is capable of operating as a RP, NTB/RP or NTB/NTB. The chipset cannot dynamically make these determinations upon power up so this information must be provided by BIOS prior to enumeration.

3.6.2 Power On Sequence for RP and NTB

Intel® Xeon® processor C5500/C3500 series systems and connecting devices/systems through the RP/NTB will likely be cycled on at different times. The following sections describe the power-on sequence and its impact to enumeration.

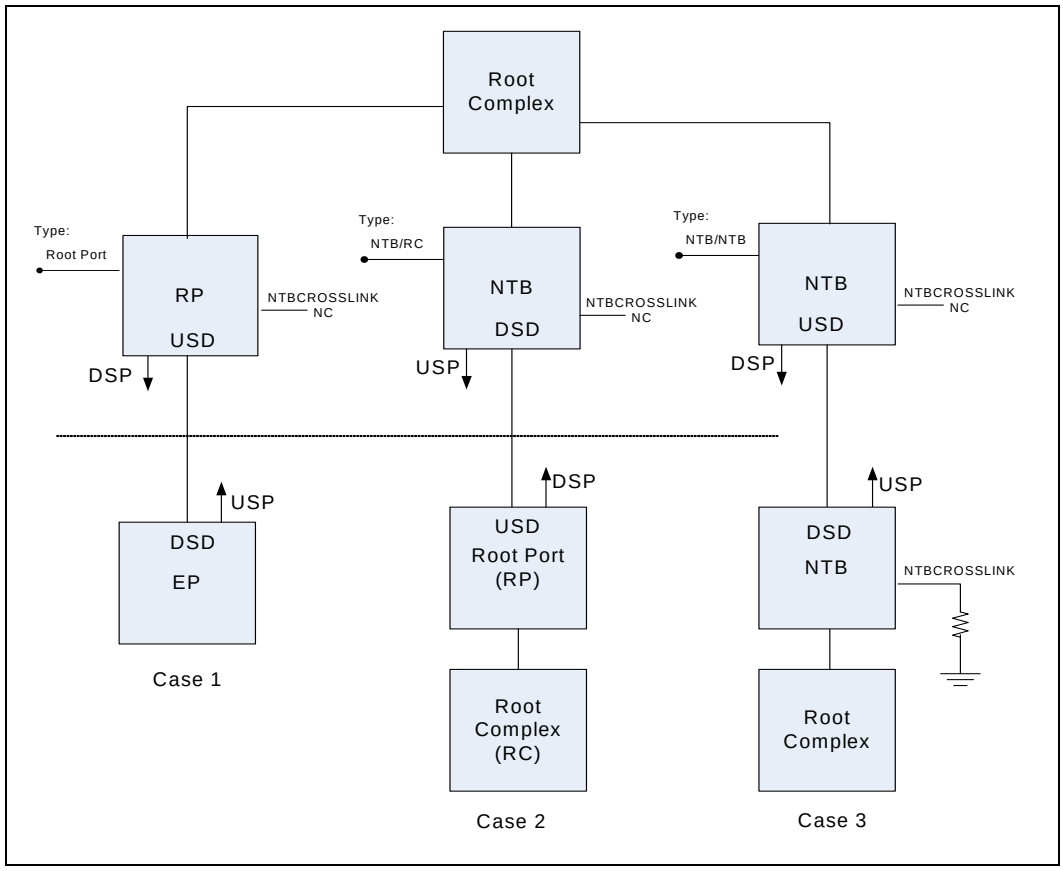
3.6.3 Crosslink Configuration

Crosslink configuration is required whenever two like PCIE ports are connected together. E.g. two downstream ports or two upstream ports.

Crosslink configuration is also only required when the PCIE port is configured as back to back NTB's. Hardware will resolve RP and NTB/RP cases based on PPD Port definition.

Figure 53 describes the three usage models and their behavior regarding crosslink training.

Figure 53. Crosslink Configuration



The following acronyms need to be understood to decode the crosslink figure shown above. Upstream device (USD)/Downstream port (DSP) and Downstream device (DSD)/Upstream port (USP).

This assumes both devices have been powered on and are capable of sending training sequences.

Case 1: Intel® Xeon® processor C5500/C3500 series Root Port (RP) connected to external endpoint (EP)

No Crosslink configuration required: Hardware will automatically strap the port as an USD/DSP when the PPD register, Port Definition field, is set to “00”b (RP).

The RP will train as USD/DSP and the EP will train as DSD/USP. No conflict occurs and link training proceeds without need for crosslink training.

Note: **When configured as a RP. the PE_NTBXL pin should be left as a no-connect (NTB logic does not look at the state of the PE_NTBXL pin when configured as a RP). The PPD Crosslink Control Override field bits 3:2 have no meaning when configured as a RP.**

Case 2: Intel® Xeon® processor C5500/C3500 series NTB connected to external RP

No Cross-link configuration is required: Hardware will automatically strap the port as an DSD/USP when the PPD register, Port Definition field, is set to “10”b (NTB/RP).

The Intel® Xeon® processor C5500/C3500 series NTB will train as DSD/USP and the external RP will train as USD/DSP. No conflict occurs and link training proceeds without need for crosslink training.

Note: **When configured as a NTB/RP. the PE_NTBXL pin should be left as a no-connect (NTB logic does not look at the state of the PE_NTBXL pin when configured as a NTB/RP). The PPD Crosslink Control Override field bits 3:2 have no meaning when configured as an NTB/RP.**

Case 3: Intel® Xeon® processor C5500/C3500 series NTB connected to another Intel® Xeon® processor C5500/C3500 series,

Crosslink configuration is required:

Two options are provided to give the end user flexibility in resolving crosslink in the case of back to back NTBs.

Option 1:

The first option is to use a pin strap to set the polarity of the port without requiring BIOS/SW interaction. The Intel® Xeon® processor C5500/C3500 series has provided the pin strap “PE_NTBXL” that is strapped at platform level to select the polarity of the NTB port.

The NTB port is forced to be an USD/DSP when the PE_NTBXL pin is left as no-connect. The NTB port is forced to be DSD/USP when the PE_NTBXL pin is pulled to ground through a resistor.

After one of the platforms NTB port is left floating (USD/DSP) and the other platforms NTB port is pulled to ground (DSD/USP), no conflict occurs and link training proceeds without need for crosslink training.

This option works as follows.

- Pin strap PE_NTBXL as defined above
- PPD, Port definition field is set to “01”b (NTB/NTB) on both platforms
- BIOS/SW enables the port to start training. (Order of release does not matter)

Option 2:

The second option is to use BIOS/SW to force the polarity of the ports prior to releasing the port.

This option works as follows.

- PPD, Port definition field is set to “01”b (NTB/NTB) on both platforms
- PPD, Crosslink Control Override is set to “11”b (USD/DSP) on one platform
- PPD, Crosslink Control Override is set to “10”b (DSD/USP) on the other platform
- BIOS/SW enables the port to start training. (Order of release does not matter)

After one of the platforms is forced to be an (USD/DSP) and the other platforms NTB port is forced to be a (DSD/USP), no conflict occurs and link training proceeds without need for crosslink training.

Note: When the PPD, Port definition field is set to “01”b (NTB/NTB) and the PPD, Crosslink control Override field is set to a value of “11”b or “10”b, the functionality of the pin

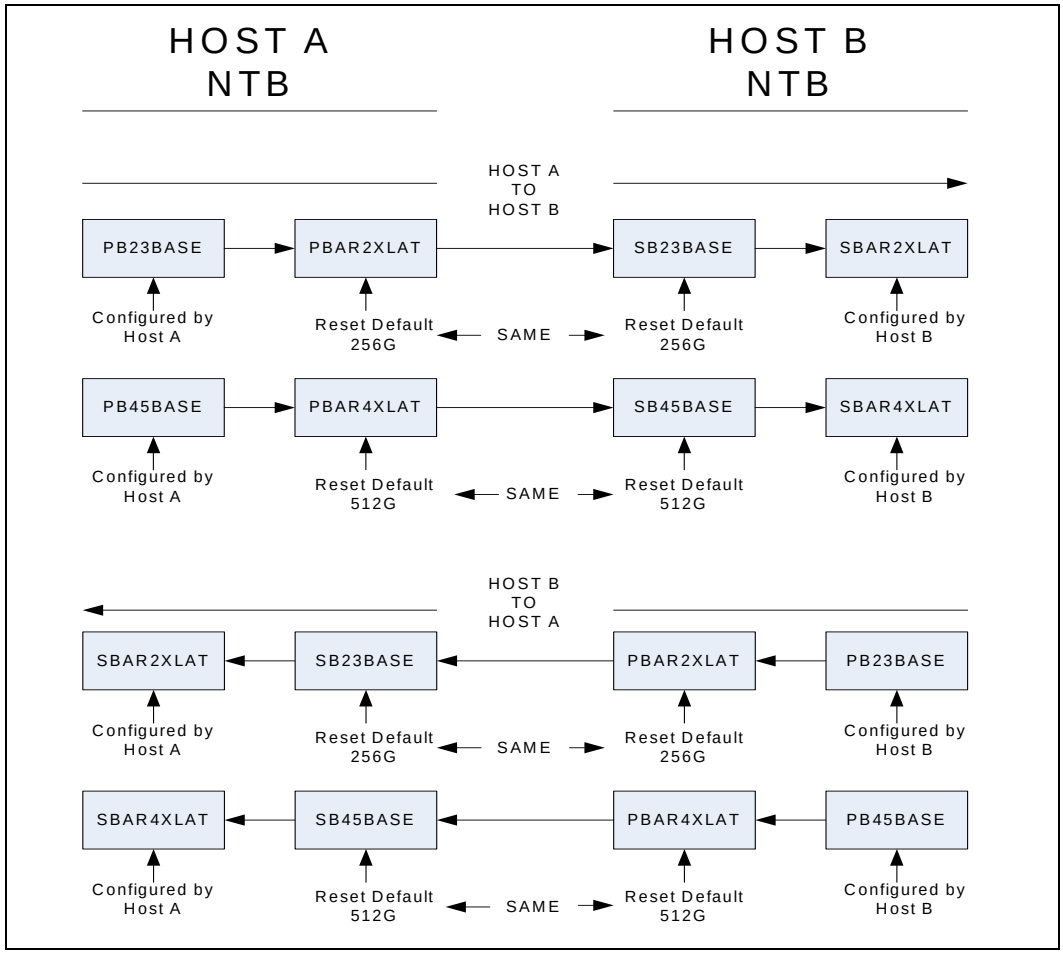
strap input PE_NTBXL is disabled and has no meaning. User should leave the PE_NTBXL pin strap unconnected in this configuration to save board space.

Note: PPD, Crosslink Configuration Status field has been provided as a means to visually see the polarity of the final result between the pin strap and the BIOS option.

3.6.4 B2B BAR and Translate Setup

When connecting two memory systems via B2B NTBs there is a requirement to match memory windows on the secondary side of the NTBs between the two systems. The registers that accomplish this are the primary bar translate registers and the secondary bar base registers on both of the connected NTBs as shown in Figure 54.

Figure 54. B2B BAR and Translate Setup



The following text explains the steps that go along with Figure 54 and assumes that we have already pre-configured the platforms for B2B operation and two memory windows in each direction. See Section 3.6.1, ""A Priori" Configuration Knowledge" for how to accomplish pre-boot configuration.

1. Host A and Host B power up independently (no required order).
2. Once each system has powered up and released control to the NTB to train the link will proceed to the L0 state (Link up).

3. Enumeration SW running independently on each host will discover and set the base address pointer for both primary BAR2/3 and primary BAR4/5 registers (PB23BASE, PB45BASE) of the NTB associated with that same host. At this point all that is known is the size and location of the memory window. E.g. 4KB to 512GB prefetchable memory window placed on a size multiple base address.

In the B2B case the memory map region that is common to the secondary side of both of the NTBs does not map to either system address map. It is only used as a mechanism to pass transactions from one NTB to the other. The requirements for this no mans land between the endpoints, is that both sides of the link must be set to the same memory window (size multiple) and must be aligned on the same base address for the associated bar.

Note: The reset default values for SB23BASE Section 3.20.2.12, and PBAR2XLAT Section 3.21.1.3, have been set to a default value of 256 GB, SB45BASE Section 3.20.2.13 and PBAR4XLAT Section 3.21.1.4 have been set to a default value of 512 GB. This provides ability to support sizes up to 256 GB window for SB23BASE and sizes up to 512 GB window for SB45BASE.

4. As a final configuration setup during run time operation the translate registers are setup by the local host associated with the physical NTB to map the transactions into the local system memory associated the respective NTB receiving the transactions. These are the SBAR2XLAT Section 3.21.1.7 and SBAR4XLAT Section 3.21.1.8 registers.

3.6.5 Enumeration and Power Sequence

Having a PCIE port that is configurable as a RP or NTB opens up additional possibilities for system level layout. For instance the second system could be on another blade in the same rack or in a separate rack all together. This design is flexible in how the system comes up regarding power cycling of the individual systems but the effects must be stated so that the end user understands what steps must be taken in order to get the two systems to communicate.

Case 1: Intel® Xeon® processor C5500/C3500 series Root Port (RP) connected to remote Endpoint (EP)

- Powered on at same time:
  Since Intel® Xeon® processor C5500/C3500 series and the attached EP are powered on at the same time enumeration will complete as expected.
- EP powered on after Intel® Xeon® processor C5500/C3500 series RP enumerates:
  When the EP is installed a hot plug event is issued in order to bring the EP on line.

Case 2: Intel® Xeon® processor C5500/C3500 series NTB connected to remote RP

- Powered on at same time:
  - Intel® Xeon® processor C5500/C3500 series NTB will enumerate and see the primary side of the NTB. The device will be seen as a RCiEP.
  - The remote host connected through the remote RP will enumerate and see the secondary side of the NTB. The device will be seen as a PCIE EP.
- Remote host connected through the remote RP is powered and enumerated before the Intel® Xeon® processor C5500/C3500 series NTB is powered on.
  - When the remote host goes through enumeration it will probe the RP connected to the NTB and find no device. (NTB is still powered off)
  - Sometime later, the Intel® Xeon® processor C5500/C3500 series NTB is powered on.

- It is the responsibility of the remote host system to introduce the Intel® Xeon® processor C5500/C3500 series NTB into its hierarchy and is outside the scope of this document to describe that procedure.
- If the attached system is another Intel® Xeon® processor C5500/C3500 series RP the EP is brought into the system as described in Case 1 above.
- When Intel® Xeon® processor C5500/C3500 series NTB is powered on and gets to enumeration, it finds its internal RCiEP and stop with respect to that port.
- When the link trains, Intel® Xeon® processor C5500/C3500 series NTB logic generates a host link up event.
- Next, a software configured and hardware generated “heartbeat” communication is setup between the two systems. The heartbeat is a periodic doorbell sent in both directions as an indication to software that each side sending the heartbeat is alive and ready for sending and receiving transactions.

Note: When the link goes up/down a “link up/down” event is issued to the local Intel® Xeon® processor C5500/C3500 series host in the same silicon as the NTB. When the link goes down the heartbeat will also be lost and all communications will be halted. Before communications are started again software must receive notification of both link up event and heartbeat from the remote link partner.

- Intel® Xeon® processor C5500/C3500 series NTB powered and enumerated before remote RP is powered on.
- The local host containing the Intel® Xeon® processor C5500/C3500 series NTB will power on and enumerate its devices. The NTB it will be discovered as RCiEP.
  - At this point the Intel® Xeon® processor C5500/C3500 series is waiting for a link up event and heartbeat before sending any transactions to the NTB port.
  - Sometime later the remote host connected through the remote RP is powered on and enumerated. When enumeration software gets to the NTB it will discover a PCIE EP.
  - The remote system will then setup and send a periodic heartbeat message. Once heartbeat and linkup are valid on each side communications can then be sent between the systems.

Case 3: Intel® Xeon® processor C5500/C3500 series NTB connected to Intel® Xeon® processor C5500/C3500 series NTB

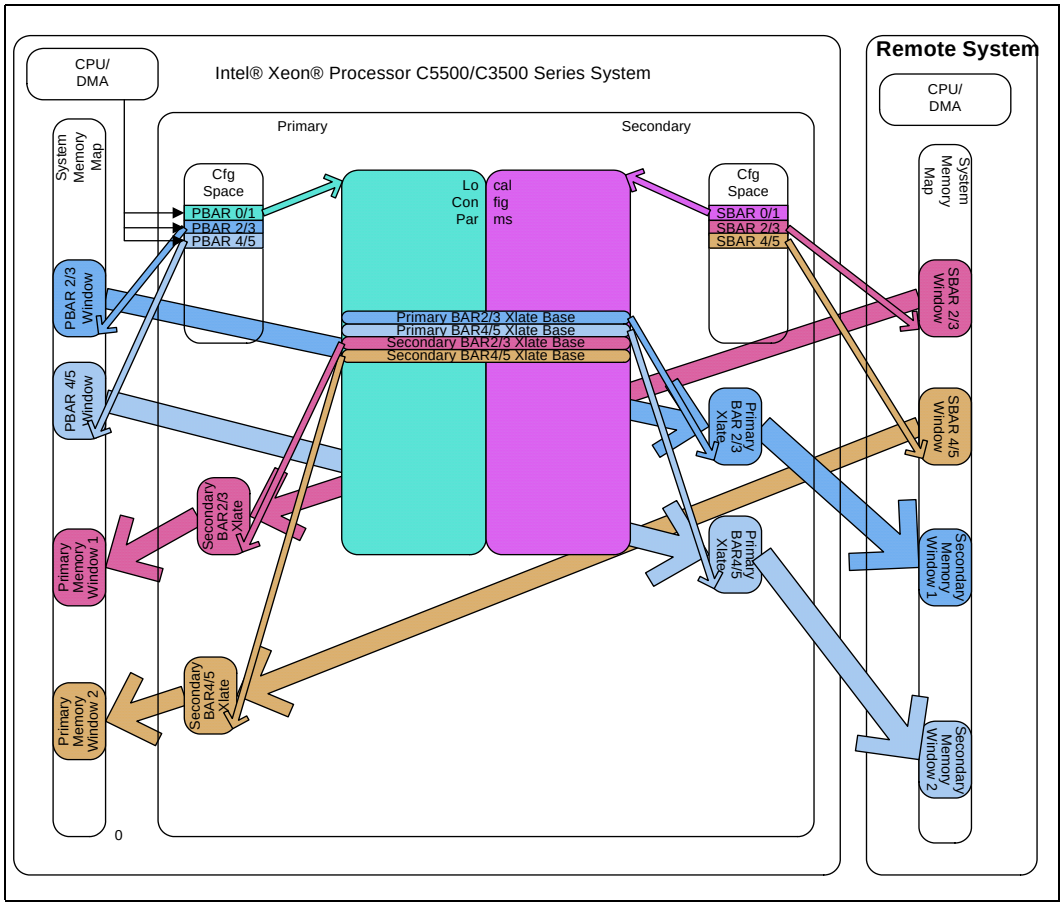
- It does not matter which side is powered on first. One side will power on, enumerate and find the internal RCiEP and then wait for link up event and a heartbeat message.
- Sometime later the system on the other side of the link is powered on and enumerated. Since it is also a NTB, it will find the internal RCiEP, and then wait for link up event and a heartbeat message.
- Now both systems are powered on and link training is started.
- Upon detection of the link up event, both sides will send a link up interrupt to their respective host.
- Both sides independently, will then setup and start sending a periodic heartbeat messages across the link.
- Once periodic heartbeat is detected by each system, it is ready for communications.

3.6.6 Address Translation

The NTB uses the BARs in the Type 0 configuration header specified above to define apertures into the memory space on the other side of the NTB. The NTB supports two sets of BARs, one on the local host interface and the other on the remote host interface.

Each BAR has control and setup registers that are writable from the other side of the bridge. The address translation register defines the address translation scheme. The limit register is used to restrict the aperture size. These registers must be programmed prior to allowing access from the remote subsystem.

Figure 55. Intel® Xeon® Processor C5500/C3500 Series NTB Port - BARs



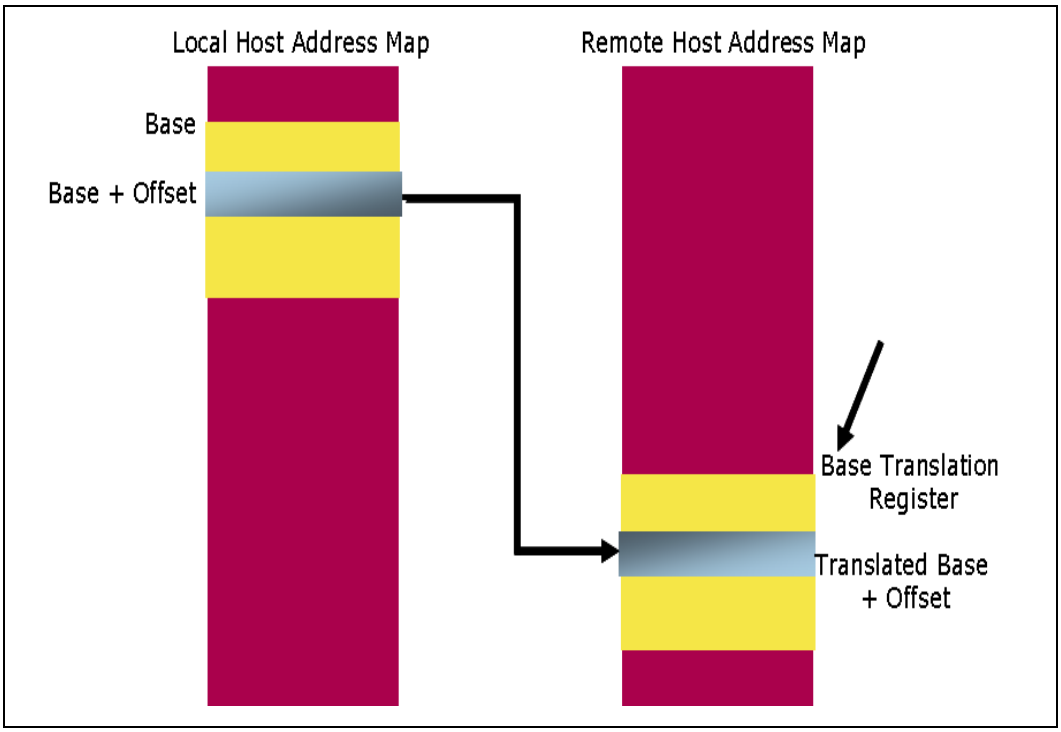
3.6.6.1 Direct Address Translation

The Intel® Xeon® processor C5500/C3500 series NTB supports two Direct Address Translation windows both inbound and outbound. These are BAR 2/3 and BAR 4/5.

Direct address translation is used to map one host address space into another host address space. The NTB is the mechanism used to connect the two host domains and translates all transactions sent across the NTB both inbound and outbound. This means all transactions traversing from the secondary side of the NTB to the primary side of the NTB are translated and all transactions traversing from the primary side of the NTB to the secondary side of the NTB are translated.

The address forwarded from one interface to the other is translated by adding a base address to the offset within the BAR that the address belongs to as shown in Figure 56.

Figure 56. Direct Address Translation



PCI Express utilizes both 32-bit and 64-bit address schemes via the 3DW and 4DW headers. To prevent address aliasing, all devices must decode the entire address range. All discussions in this section refer to 64-bit addressing. If the 3DW header is used the upper 32-bits of address are assumed to be 0000_0000h.

The NTB allows external PCI Express requesters to access memory space via address routed TLPs. The PCI Express requesters can read or write NTB memory-mapped registers or Intel® Xeon® processor C5500/C3500 series local memory space. The process of inbound/outbound address translation involves two steps:

- Address Detection Inbound/Outbound
  - Test to see if the PCI address is within the base and limit registers defined for BAR 2/3, 4/5.
  - If the address is outside of the window defined by the base and limit registers, the transaction will be terminated as an unsupported request (UR).
- Address Translation
  - Inbound with VT-d2 turned off.
    - Translate a remote address to a local physical address.
  - Inbound with VT-d2 turned on.
    - Translate a remote address to a local guest physical address that is then forwarded to the VT-d2 logic. The VT-d2 logic then converts the guest physical address to a host physical address.
  - Outbound:
    - Translate a local physical address to a remote guest address.

The following registers are used to translate the local physical address to the remote guest address in the remote host system map (Transactions going across NTB from primary side to secondary side)

Section 3.19.2.12, “PB23BASE: Primary BAR 2/3 Base Address”

Section 3.19.2.13, “PB45BASE: Primary BAR 4/5 Base Address”

Section 3.21.1.1, “PBAR2LMT: Primary BAR 2/3 Limit”

Section 3.21.1.2, “PBAR4LMT: Primary BAR 4/5 Limit”

Section 3.21.1.3, “PBAR2XLAT: Primary BAR 2/3 Translate”

Section 3.21.1.4, “PBAR4XLAT: Primary BAR 4/5 Translate”

The following registers are used to translate the remote guest address map to the local guest address or local physical address map depending on VT-d2 enabled/disabled respectively. (Transactions going across NTB from secondary side to primary side)

Section 3.20.2.12, “SB23BASE: Secondary BAR 2/3 Base Address (PCIE NTB Mode)” ,

Section 3.20.2.13, “SB45BASE: Secondary BAR 4/5 Base Address”

Section 3.21.1.5, “SBAR2LMT: Secondary BAR 2/3 Limit”

Section 3.21.1.6, “SBAR4LMT: Secondary BAR 4/5 Limit”

Section 3.21.1.7, “SBAR2XLAT: Secondary BAR 2/3 Translate”

Section 3.21.1.8, “SBAR4XLAT: Secondary BAR 4/5 Translate”

As an example direct address translation for a packet that is transmitted from the remote guest address into the local address map using BAR 2/3 registers.

Address detection equation:

Valid Address = ((Limit > Received Address[63:0] >= Base))

Register Values:

SB23BASE = 0000 003A 0000 0000H -- BAR 2/3 base address, placed on 4GB alignment by OS

SBAR2LMT = 0000 003A C000 0000H -- Reduce window to 3GB

Received Address = 0000 003A 00A0 0000H -- Valid address proceeds to translation equation

Received Address = 0000 003A C000 0001H -- Invalid address returned as UR

Translation equation: (Used after valid Address detection)

Translated Address = ((Received Address[63:0] & ~Sign_Extend(2^SBAR23SZ) | XLAT Register[63:0])).

For example, to translate an incoming address claimed by a 4 GB window based at 0000 003A 0000 0000H to a 4 GB window based at 0000 0040 0000 0000H.

Calculation:
Received Address[63:0] = 0000 003A 00A0 0000H

SBAR23SZ = 32 -- Sets the size of Secondary BAR 2/3 = 4GB
~Sign_Extend(2^SBAR23SZ) = ~Sign_Extend(0000 0001 0000 0000H) = ~(FFFF FFFF 0000 0000H) = 0000 0000 FFFF FFFFH)

SBAR2XLAT = 0000 0040 0000 0000H -- Base address into the primary side memory (size multiple aligned)
Translated Address = 0000 003A 00A0 0000H & 0000 0000 FFFF FFFFH | 0000 0040 0000 0000H = 0000 0040 00A0 0000H

The offset to the base of the 4 GB window on the incoming address is preserved in the translated address.

3.6.7 Requester ID Translation

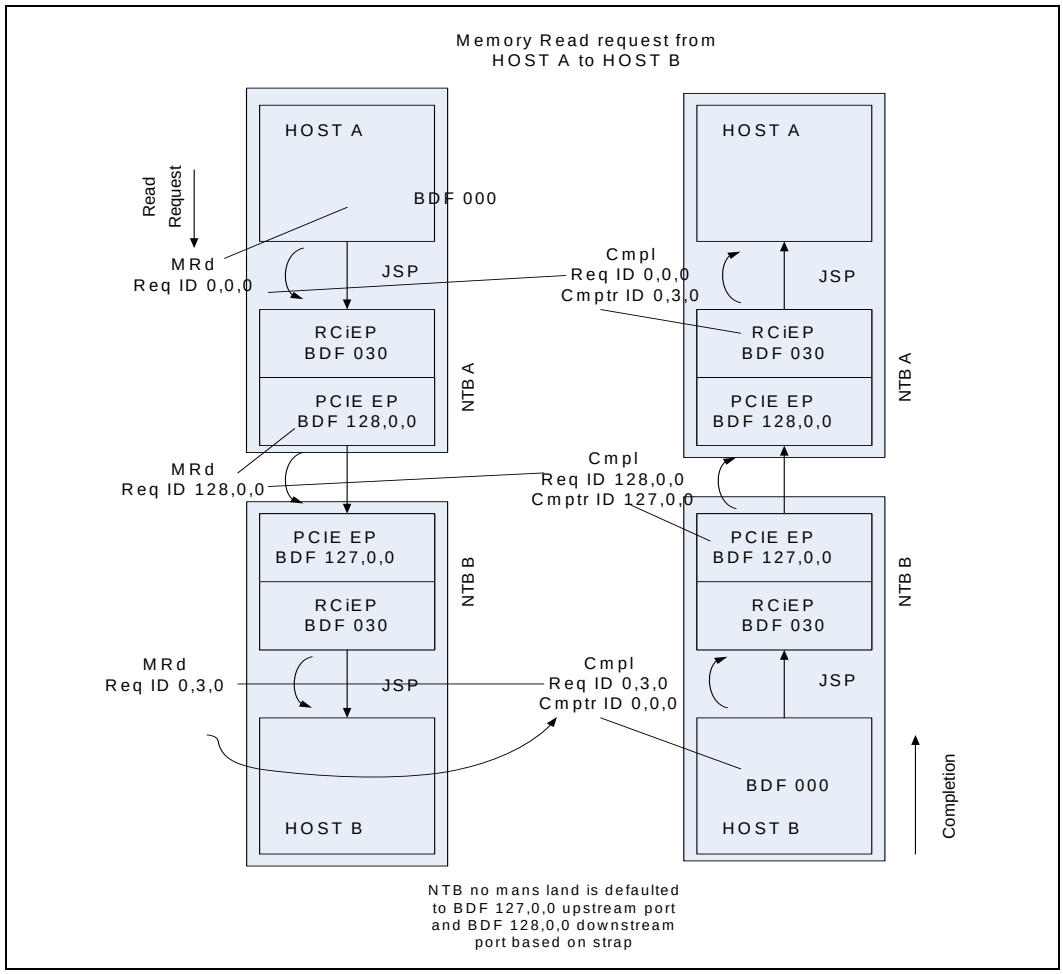
Completions for non-posted transactions are routed using Requester ID instead of the address. The NTB provides a mechanism to translate the Requester ID and the Completer ID from one domain to the other.

The Requester ID consists of the Requester's PCI bus number, device number and function number. The completer ID consists of completer's PCI bus number, device number and function number

For Intel® Xeon® processor C5500/C3500 series NTB, the primary side of the NTB will have a fixed Bus Device Function (BDF) which is BDF = 0,3,0. The BDF of the secondary side of the NTB depends on the configuration selected.

If the configuration is NTB/NTB then the BDF of the secondary side of the NTB will be defined by the Section 3.0, "PCI Express Non-Transparent Bridge" . This is because in the NTB/NTB case no configuration transactions are sent across the link and the local host associated with the NTB must setup both sides of the NTB. See Figure 57 for an example of how Requester and Completer ID translation are handled by hardware.

Figure 57. NTB to NTB Read Request, ID translation Example



If the configuration is NTB/RP then the secondary side of the NTB will be per the *PCI Express Base Specification,* Revision 2.0. For this configuration the secondary side of the NTB must capture the Bus and Device numbers supplied with all Type 0 Configuration Write requests sent across the link to the NTB.

For inbound reads received from the remote host, the NTB performs the address translation and launches the memory read on the local processor. The completions returned from memory are translated and returned back to the remote host using the correct Completer ID (the secondary side of the NTB).

For outbound reads, the NTB performs the address translation and uses the captured BDF as the Requester ID for the transaction sent across the link.

See Figure 58 and Figure 59 for examples of how Requester and Completer ID translation are handled by hardware for the NTB/RP configuration.

Figure 58. NTB to RP Read Request, ID translation Example

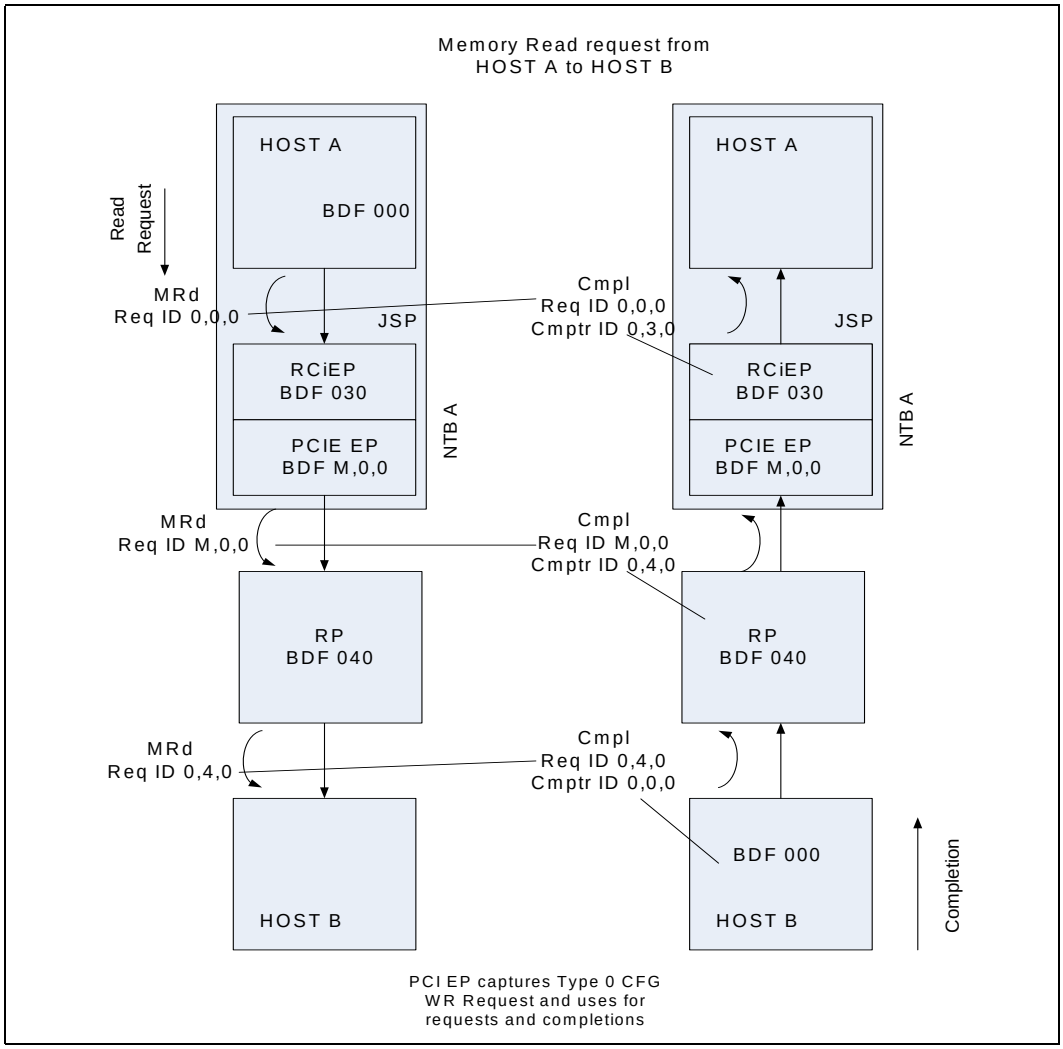
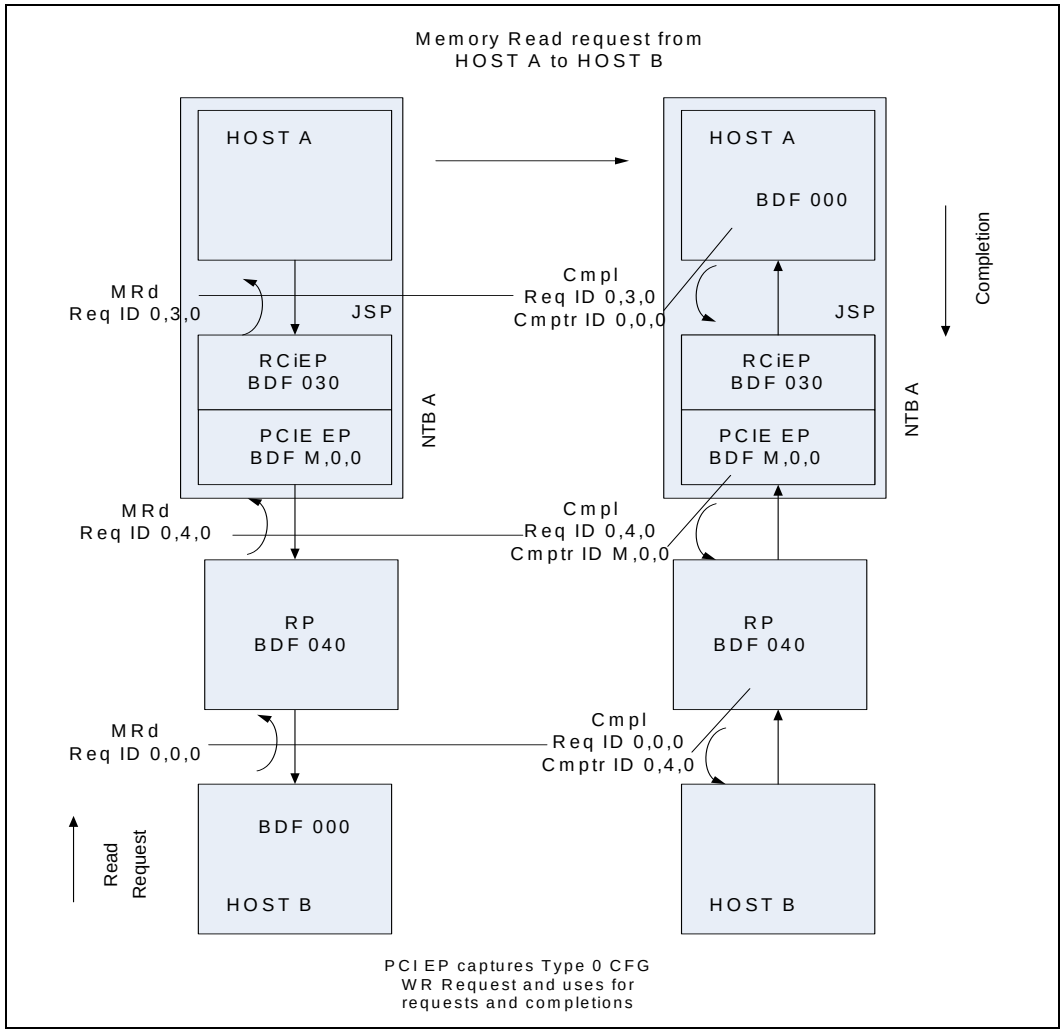


Figure 59. RP to NTB Read Request, ID translation Example



3.6.8 Peer-to-Peer Across NTB Bridge

Inbound transactions (both posted writes and non-posted reads) on the Intel® Xeon® processor C5500/C3500 series NTB can be targeted to either the local memory or to a peer PCIE port on the Intel® Xeon® processor C5500/C3500 series. This allows usage models where systems can access peer PCIE devices across the NTB port.

The NTB controller will provide a mechanism to steer transactions to either local memory or to a peer port. For non-posted reads, the NTB port will provide a mechanism to translate the Requester ID across the NTB port while the peer port will provide the mechanism to translate the Requester ID for the peer traffic.

3.7 NTB Inbound Transactions

This section talks about the NTB behavior for transactions that originate from an external agent on the PCIE link towards the PCI Express NTB port. Throughout this chapter, inbound refers to the direction towards the CPU from I/O.

3.7.1 Memory, I/O and Configuration Transactions

Table 81 lists the memory and configuration transactions supported by the Intel® Xeon® processor C5500/C3500 series which are expected to be received from the PCI Express NTB port.

The PCI Express NTB port does not support IO transactions.

For more specific information relating to how these transactions are decoded and forwarded to other interfaces, see Section 6.0, "System Address Map" .

Table 81. Incoming PCI Express NTB Memory, I/O and Configuration Request/ Completion Cycles

PCI Express Transaction	Address Space or Message	IIO Response
Inbound Write Requests	Memory	After address translation, packets are accepted by the NTB if targeting NTB MMIO space, or forwarded to Main Memory, PCI Express port (local or remote) or DMI (local or remote) depending on address.
	I/O	The NTB does not claim any IO space resources and as such should never be the recipient of an inbound IO request. If this occurs it will be returned to the requester with completion status of UR.
	Type 0 Configuration	Accepted by the NTB if targeted to the secondary side of the NTB. All other configuration cycles are unsupported and are returned with completion status of UR. ***Note:*** This will only be seen in case of NTB/RP. In NTB/NTB case configuration transaction will not be seen on the wire.
	Type 1 Configuration	Type 1 configurations are not supported and are returned with completion status of UR
Inbound Completions from Outbound Write Request	I/O	CPU will never generate an IO request to the NTB so this will never occur.
	Configuration	Configuration transactions will never be sent on the wire from the NTB perspective so this will never occur. ***Note:*** The NTB can be the target of CPU generated configuration requests to the primary side configuration registers.
Inbound Read Requests	Memory	After address translation packets are accepted by the NTB if targeting NTB MMIO space or forwarded to Main Memory, PCI Express port (local or remote), DMI (local or remote).
	I/O	The NTB does not claim any IO space resources and as such should never be the recipient of an inbound IO request. If this occurs it will be returned to the requester with completion status of UR.
	Type 0 Configuration	Accepted by the NTB if targeted to the secondary side of the bridge all other configuration cycles are unsupported and are returned with completion status of UR. ***Note:*** This will only be seen in case of NTB/RP. In NTB/NTB case configuration transaction will not be seen on the wire.
	Type 1 Configuration	Type 1 configurations are not supported and are returned with completion status of UR
Inbound Completions from Outbound Read Requests	Memory	Forward to CPU, PCI Express port (local or remote) or DMI (local or remote).
	I/O	CPU will never generate an IO request to the NTB so this will never occur.
	Configuration	Configuration transactions will never be sent on the wire from the NTB perspective so this will never occur. ***Note:*** The NTB can be the target of CPU generated configuration requests to the primary side configuration registers.

3.7.2 Inbound PCI Express Messages Supported

Table 82 lists all inbound messages that Intel® Xeon® processor C5500/C3500 series supports receiving on a PCI Express NTB secondary side. 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 82. Incoming PCI Express Message Cycles

PCI Express Transaction	Address Space or Message	IIO Response
Inbound Message	Unlock	Silently dropped by NTB. ***Note:*** PCI Express-compliant software drivers and applications must be written to prevent the use of lock semantics when accessing NTB. Because the unlock message could still be received by NTB because the RP or NTB on other side could be 'broadcasting' unlock to all ports when a lock sequence to a device (that is NOT connected to JSP) in the remote system completes.
	EOI (Intel® VDM)	Silently dropped by NTB. ***Note:*** This message could be received from remote RP or NTB that is broadcasting this message and all receivers are supposed to ignore it.
	PME_Turn_Off	The PME_turn_Off message is initiated by the remote host that is connected to the secondary side of the NTB in preparation for removing power on the remote host. ***Note:*** This only applies to NTB/RP case. The NTB/NTB case is defined by PME_Turn_Off defined in Table 84. NTB will receive and acknowledge this message with PME_TO_ACK
	PM_REQUEST_ACK (DLLP)	After the NTB sends a PM_Enter_L1 to the remote host, the remote host then blocks subsequent TLP issue and wait for all pending TLPs to Ack. The remote host will then send a PM_REQUEST_ACK back to the NTB. This message is continuously issued until the receiver link is idle. See the *PCI Express Base Specification,* Revision 2.0 for details. ***Note:*** PM_REQUEST_ACK DLLP is an inbound packet in the case of NTB/RP. For NTB/NTB this message will be seen as an outbound message from the USD NTB and an inbound message on the DSD NTB.
	PM_Active_State_Nak	When secondary side of the NTB receives a PM_Active_State_Request_L1 from the link partner and due to a temporary condition, it cannot transition to L1, it responds with PM_Active_State_Nak.
	Set_Slot_Power_Limit	Message that is sent to PCI Express device when software wrote to the Slot Capabilities Register or the PCI Express link transitions to DL_Up state. See the *PCI Express Base Specification,* Revision 2.0 for more details.
	All Other Messages	Silently discard if message type is type 1 and drop and log error if message type is type 0

3.7.2.1 Error Reporting

PCI Express NTB reports many error conditions on the primary side of the NTB through explicit error messages: ERR_COR, ERR_NONFATAL, ERR_FATAL. Intel® Xeon® processor C5500/C3500 series can be programmed to do one of the following when it receives one of these error messages:

- Generate MSI,MSI-X
- Forward the messages to PCH

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

The NTB does not report any error message towards the link partner on the secondary side of the NTB.

3.8 Outbound Transactions

This section describes the NTB behavior towards outbound transactions to an external agent on the PCIE link. Throughout the rest of the chapter, outbound refers to the direction from CPU towards I/O.

3.8.1 Memory, I/O and Configuration Transactions

The IIO will generate outbound memory transactions to NTB MMIO space and to memory on an external agent connected to the secondary side of the NTB across the PCI Express link.

The IIO will never generate I/O and configuration cycles that are sent outbound on the PCI Express link. The IIO will generate configuration cycles to the primary side of the NTB. All transaction behavior are listed in Table 83.

Table 83. 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	Accepted by the NTB if targeting MMIO space claimed by the NTB or after address detection and translation sent from the primary side to the secondary side of the NTB and on the link partner connected to the secondary side of the NTB.
	I/O	CPU will never generate an IO requests to the NTB so this will never occur.
	Configuration	Accepted by the NTB if targeted to the primary side of the NTB. (Positive decoded) Configuration transactions will never be sent outbound on the wire as the NTB is an endpoint so this will never occur.
Outbound Completions for Inbound Write Requests	I/O	The NTB does not claim any IO space resources and as such should never be the recipient of an inbound IO request. If this occurs it will be returned to the requester with completion status of UR.
	Type 0 Configuration	Response from inbound Type 0 configuration requests targeted to the secondary side configuration space of the NTB. All other configuration cycles are unsupported and are returned with completion status of UR. ***Note:*** This will only be seen in case of NTB/RP. In NTB/NTB case configuration transaction will not be seen on the wire.
	Type 1 Configuration	Type 1 configurations are not supported and are returned with completion status of UR
Outbound Read Requests	Memory	Accepted by the NTB if targeting MMIO space claimed by the NTB or after address detection and translation sent from the primary side to the secondary side of the NTB and on the link partner connected to the secondary side of the NTB.
	I/O	CPU will never generate an IO requests to the NTB so this will never occur.
	Configuration	Accepted by the NTB if targeted to the primary side of the NTB. (Positive decoded) Configuration transactions will never be sent outbound on the wire as the NTB is an endpoint so this will never occur.
Outbound Completions for Inbound Read Requests	Memory	Response for an inbound read targeting MMIO space claimed by the NTB, or after address detection and translation sent from the secondary side to the primary side of the NTB targeting main memory or a peer I/O device.
	I/O	The NTB does not claim any IO space resources and as such should never be the recipient of an inbound IO request. If this occurs it will be returned to the requester with completion status of UR.
	Type 0 Configuration	Response from inbound Type 0 configuration requests targeted to the secondary side configuration space of the NTB. All other configuration cycles are unsupported and are returned with completion status of UR. ***Note:*** This will only be seen in case of NTB/RP. In NTB/NTB case configuration transaction will not be seen on the wire.
	Type 1 Configuration	Type 1 configurations are not supported and are returned with completion status of UR

3.8.2 Lock Support

The NTB does not support lock cycles from either side of the NTB. The local host views the NTB as a RCiEP (primary side). The remote host views the NTB as a PCIE EP (secondary side).

- Primary side: PCI Express-compliant software drivers and applications must be written to prevent the use of lock semantics when accessing a Root Complex Integrated Endpoint.

Note: If erroneous software is written and lock cycles are sent from the local Intel® Xeon® processor C5500/C3500 series host to the primary side of the NTB they will be forward across the NTB to the secondary side and then passed along to the link partner attached to the NTB. If the link partner is capable of responding to the illegal upstream MRdLk request then the link partner will respond with a completion with status UR. If the link partner cannot respond to illegal upstream MRdLk request and drops the request, the NTBs completion time-out timer will time-out and complete the MRdLk request with a master abort (MA).

- Secondary side: PCI Express-compliant software drivers and applications must be written to prevent the use of lock semantics when accessing a PCI Express Endpoint.

Note: If erroneous software is written and lock cycles are sent from the external host to the secondary side of the NTB they will be completed by the NTB and returned with a completion status of UR.

3.8.3 Outbound Messages Supported

Table 84 provides a list the behavior of the NTB to downstream messages supported and not supported and appropriate behavior to these messages.

Table 84. Outgoing PCI Express Message Cycles with Respect to NTB

PCI Express Transaction	Address Space or Message	Reason for Issue
Outbound Messages	Unlock	The NTB does not support lock cycles from either side of the NTB. Primary side: PCI Express-compliant software drivers and applications must be written to prevent the use of lock semantics when accessing a Root Complex Integrated Endpoint. If erroneous software is written and the lock sequence is sent, it will be followed by an "Unlock" message to complete the lock sequence. The NTB will pass the Unlock message from the primary side to the secondary side of the bridge and then on the wire, to the remote host where it should be dropped. There is no completion.
	ASSERT_INTA DEASSERT_INTA ASSERT_INTB DEASSERT_INTB ASSERT_INTC DEASSERT_INTC ASSERT_INTD DEASSERT_INTD	These messages are only used in NTB/RP configuration and are sent from the NTB towards the RP when the local host writes any of the bits in the SDOORBELL and the INTx interrupt mechanism is enabled. INTA-D selection is based on setting of Section 3.20.2.18, "INTPIN: Interrupt Pin Register"
	PME_Turn_Off	When the local host on the primary side of the NTB wants to initiate power removal on the local system. SW on the local host sets the PME_TURN_OFF bit 5 in Section 3.19.4.20, "MISCCTRLSTS: Misc. Control and Status Register" . HW will then clear bit 5 and set bit 48 PME_TO_ACK followed by sending an upstream PM_Enter_L23. See Section 8.0, "Power Management" for details. ***Note:*** The PME_turn_Off message is never sent on the wire to the link partner from the local host to the remote host. The message and response are faked internal to the NTB.
	PME_TO_ACK	Upon receiving the PME_Turn_Off message on the secondary side of the NTB from the remote host the NTB will return a PME_TO_ACK message to the remote host.
	PM_PME	Propagate as an interrupt/general purpose event to the system. For details, refer to Section 8.0, "Power Management" .
	PM_ENTER_L1 (DLLP)	After remote host writes a state change request to the PMCS register Section 3.20.3.27, "PMCSR: Power Management Control and Status Register" on the secondary side of the NTB, the NTB then blocks subsequent TLP issue and wait for all pending TLPs to Ack. Then, sends a PM_ENTER_L1 to the remote host. ***Note:*** PM_ENTER_L1 is an outbound packet in the case of NTB/RP. For NTB/NTB case this message will be seen as an outbound message from the DSD NTB and an inbound message on the USD NTB.
	PM_ENTER_L23 (DLLP)	After sending the PME_TO_Ack, the secondary side NTB sends the PM_ENTER_L23 to the remote host to indicate to the remote host that it can remove power to the remote host. ***Note:*** PM_ENTER_L23 is an outbound packet in the case of NTB/RP. For NTB/NTB case this message will be seen as an outbound message from the DSD NTB and an inbound message on the USD NTB.
	PM_ACTIVE_STATE_REQUEST_L1 (DLLP)	After receiving acknowledgement from the link layer for the last TLP sent it can issue a PM_ACTIVE_STATE_REQUEST_L1 to the Upstream device. ***Note:*** PM_ACTIVE_STATE_REQUEST_L1 is an outbound packet in the case of NTB/RP. For NTB/NTB case this message will be seen as an outbound message from the DSD NTB and an inbound message on the USD NTB.
	All Other Messages	Silently discard if message type is type 1 and drop and log error if message type is type 0.

3.8.3.1 EOI

NTB is a Root Complex integrated End Point (RCiEP) with respect to the local host and as such should not receive EOI messages from the host when configured as a NTB.

Note: Due to hardware simplification in the PCIE logic, the BIOS must set bit 26 Disable EOI in the Section 3.19.4.20, "MISCCTRLSTS: Misc. Control and Status Register" to prevent EOI message from being sent when configured as a NTB.

3.9 32-/64-Bit Addressing

For inbound and outbound memory reads and writes, the IIO supports the 64-bit address format. If an outbound transaction's address is less than 4 GB, the IIO will issue the transaction with a 32-bit addressing format on PCI Express. Only when the address is greater than 4 GB then IIO will initiate transaction with 64-bit addressing format. See Section 8.0, "Power Management" for details of addressing limits imposed by Intel® QuickPath Interconnect and the resultant address checks that the IIO does on PCI Express packets it receives.

3.10 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

3.10.1 Transaction ID

The Transaction ID uniquely identifies every transaction in the system. The Transaction ID comprises four sub-fields described in Table 85. This table provides details on how this field in the Express header is populated by the IIO.

Table 85. PCI Express Transaction ID Handling

Field	Definition	IIO as Requester	IIO as Completer
Bus Number	Specifies the bus number that the requester resides on.	The IIO fills this field in with its internal Bus Number that the PCI Express cluster resides on the IIOBUSNO: IIO Internal Bus Number. See Section 3.6.3.17, “IIOBUSNO: IIO Internal Bus Number” in Volume 2 of the Datasheet.	The IIO 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 IIO fills this field in with its Device Number that the PCI Express cluster owns. Device 3 in this case.	
Function Number	Specifies the function number of the requester.	The IIO fills this field in with its Function Number that the PCI Express cluster owns. Function 0 in this case.	
Tag	Contains 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.	NP tx: The IIO fills this field in with a value such that every pending request carries a unique Tag. NP Tag[7:5]=QPI Source NodeID[4:2]. Bits 7:5 can be non-zero only when 8-bit tag usage is enabled. Otherwise, IIO always zeros out 7:5. NP Tag[4:0]=Any algorithm that guarantees uniqueness across all pending NP requests from the port. P Tx: No uniqueness guaranteed. Tag[7:0]=QPI Source NodeID[7:0] for CPU requests. Bits 7:5 can be non-zero only when 8-bit tag usage is enabled. Otherwise, IIO always zeros out 7:5.	

3.10.2 Attributes

PCI Express supports two attribute hints described in Table 86.This table provides how Intel® Xeon® processor C5500/C3500 series populates these attribute fields for requests and completions it generates.

Table 86. PCI Express Attribute Handling

Attribute	Definition	IIO as Requester	IIO 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 for transactions that Intel® Xeon® processor C5500/ C3500 series generates on PCIE on behalf of an Intel® QPI request. On peer-to-peer requests, the IIO forwards this attribute as-is.	Intel® Xeon® processor C5500/C3500 series 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.		

3.10.3 Traffic Class

The IIO does not optimize based on traffic class. The IIO can receive a packet with TC!= 0 and treat the packet as if it were TC = 0 from an ordering perspective. IIO forwards the TC filed 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.

3.11 Completer ID

The Completer ID field is used in PCI Express completion packets to identify the completer of the transaction. The CompleterID comprises three sub-fields described in Table 87.

Table 87. PCI Express CompleterID Handling

Field	Definition	IIO as Completer
Bus Number	Specifies the bus number that the completer resides on.	The IIO fills this field in with its internal Bus Number that the PCI Express cluster resides on
Device Number	Specifies the device number of the completer.	Device number of the root port sending the completion back to PCIE.
Function Number	Specifies the function number of the completer.	0

3.12 Initialization

This section documents the initialization flow for the different usage models.

3.12.1 Initialization Sequence with NTB Ports Connected Back-to-Back (NTB/NTB)

This usage model is discussed in Section 3.5.1. In this configuration, the secondary side of one NTB is connected to the secondary side of another NTB.

Note: This section assumes that BAR sizes have already been defined per Section 3.12, “Initialization” and crosslink configuration has been completed (if required). See Section 3.6.3, “Crosslink Configuration” .

BIOS executing on the local host (the on-die core) on each system writes the primary and secondary side NTB BAR sizes and PPD from the FWH or CMOS.

Enumeration SW reads the BARs and then sets the BAR locations in system memory.

Run time OS will configure the primary and secondary limit registers and primary and secondary address translation registers of the NTB. See Section 3.6.4, “B2B BAR and Translate Setup” .

The PCIE links attempt to initialize and train.

Once the links are trained, higher level software or the NTB device driver configures the remote host interface of the NTB on both systems and enables the connectivity between the two systems.

The advantage of this system is that connecting the NTB ports back-to-back has the following advantages.

- BIOS on both systems can be identical. The BIOS configures both the local and remote host interface of the NTB without requiring link training to complete.

- BIOS enumerates the NTB in the local host address space. The mapping of the remote host interface to the other system is done subsequently by higher level platform software.
- This mechanism avoids the race condition and timing relationship between when the two systems initialize. Each system initializes only its internal components and does not have any dependency on the availability and timing of the second system.

3.12.2 Initialization Sequence with NTB Port Connected to Root Port

This usage model is discussed in Section 3.5.2 and Section 3.5.3. In this configuration, the downstream root port on one system is connected to the secondary side of the NTB on the second system. This configuration requires the crosslink configuration described in Section 3.6.3, “Crosslink Configuration” , in order for the PCIE links in the system to initialize and train correctly.

The root port must not be allowed to enumerate the NTB port in the remote host memory space until the local host has completed the configuration of the NTB on the Intel® Xeon® processor C5500/C3500 series. Otherwise, the remote host may detect erroneous BAR and configuration registers. To ensure the correct order of the initialization sequence in this configuration one flag bit is used, the remote host access bit. Section 3.21.1.12, “NTBCNTL: NTB Control” Bit 1. At reset, bit is cleared. When the remote host access bit is cleared, the remote host cannot access the NTB.

The BIOS executing on the local host first configures the local host interface of the NTB. While this operation is underway, the remote host access bit is cleared. As a result even if the remote host completes its initialization and tries to run a discovery cycle to discover and enumerate the NTB, it is not allowed to access the NTB resources. So, the remote host is prevented from enumerating the NTB until the local host has completed the entire configuration of the bridge.

Once the NTB resources are fully configured, the BIOS sets the remote bus access bit. Subsequently, if the remote host tries to discover and enumerate the NTB, it will succeed. The BIOS also generates a hot-plug event to the remote host to indicate that Endpoint device (bridge) is now functional. The root port can then service the hot plug event and discover/enumerate the NTB.

Connecting the NTB port on one system to a root port on another system allows the Intel® Xeon® processor C5500/C3500 series system to be connected to the root port of any system, not necessarily a Intel® Xeon® processor C5500/C3500 series system.

3.13 Reset Requirements

The NTB isolates two independent systems. As such, a system reset on one system must not cause any reset activity on the second system. When one of the systems connected through the NTB port goes down, the corresponding PCIE link goes down. The second system will eventually detect that PCIE link down status and flush all pending transactions to/from the system that went down.

3.14 Power Management

The NTB will provide the D0/D3 device on/off capability. In addition, the NTB port will also support L0s state.

3.15 Scratch Pad and Doorbell Registers

Intel® Xeon® processor C5500/C3500 series supports sixteen, 32-bit scratch pad registers, (total 64B) that are accessible through the BAR0 configuration space.

The processor supports two, 16-bit doorbell registers (PDOORBELL and SDOORBELL) that are accessible through the BAR0 configuration space.

Interrupts (INTx, MSI and MSI-X) always travel in the upstream direction, they cannot be used to send interrupts across the NTB. If allowed this would mean that INTx, MSI and MSI-X would be traveling downstream from the root which is illegal. The doorbell mechanism used to send interrupts across a NTB to overcome this specific issue and to allow for inter processor interrupt communications.

Example of a doorbell with NTB/RP configuration:

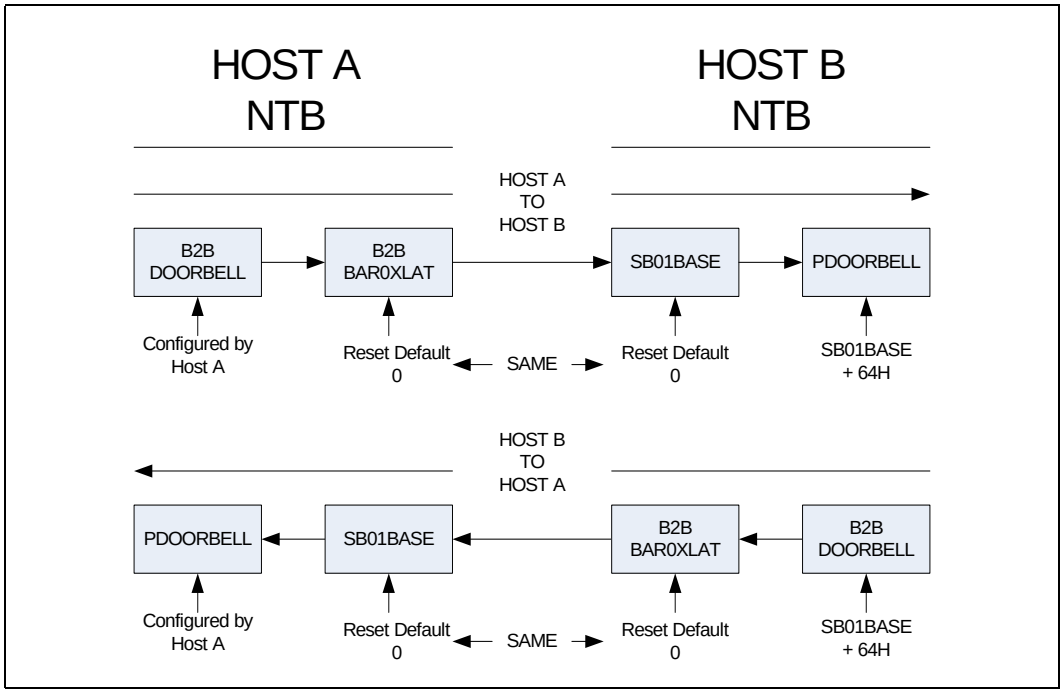
System A wishes to off load some packet processing to System B. System A writes the packets to the Primary BAR 2/3 window Section 3.19.2.12, “PB23BASE: Primary BAR 2/3 Base Address” and then into System B memory space through the corresponding Primary BAR 2/3 Translate window. Section 3.21.1.3, “PBAR2XLAT: Primary BAR 2/3 Translate”

Next a bit in the Secondary Doorbell register Section 3.21.1.17, “SDOORBELL: Secondary Doorbell” is written to start the interrupt process. Hardware on the secondary side of the NTB upon sensing that a doorbell bit was written generates an upstream interrupt. The type of the interrupt is fully programmable to be either an INTx, MSI, or MSI-X.

Upon receiving the interrupt in the local IOAPIC an ISR will do a read to the SDOORBELL to determine what the cause of the interrupt is and then do a write back to the same register to clear the bits that were set.

Example of a doorbell with NTB/NTB configuration:

Figure 60. B2B Doorbell



For NTB/NTB configuration an additional register and passing mechanism has been created to overcome the issue of back to back endpoints and will work as outlined in the example below.

Host A wishes to send heartbeat indication to Host B to notify Host B that Host A is alive and functional.

1. Host A sets a selected bit in the B2B Doorbell register Section 3.21.1.26, “B2BDOORBELL: Back-to-Back Doorbell” .
2. HW on Host A senses that the B2B doorbell has been set and creates a PMW and sends it across the link to the NTB on Host B.

Note: The default base address for B2BBAR0XLAT Section 3.21.1.26, “B2BDOORBELL: Back-to-Back Doorbell” and SB01BASE Section 3.20.2.11, “SB01BASE: Secondary BAR 0/1 Base Address (PCIE NTB Mode)” have been set to 0 so that the memory windows will align. The registers are RW and programmable from the local host associated with the physical NTB if the default values are not sufficient for the user model.

3. Transaction is received by the secondary side of the NTB on the other side of the link through the SB01BASE window.
4. HW in Host B NTB decodes the PMW as its own and sets the equivalent bits in the Primary Doorbell register Section 3.21.1.15, “PDOORBELL: Primary Doorbell” .
5. HW upon seeing the bit(s) the Primary Doorbell being set, generates an upstream interrupt based on if INTx or MSI or MSI-X is enabled and not masked.

3.16 MSI-X Vector Mapping

Intel® Xeon® processor C5500/C3500 series provides four MSI-X vectors which are mapped to groups of PDOORBELL bits per Table 96, "MSI-X Vector Handling and Processing by IIO on Primary Side". If the OS cannot support 4 MSI-X vectors but is capable of programming all of the MSI-X table and data registers, Section 3.21.2.1, "PMSIXTBL[0-3]: Primary MSI-X Table Address Register 0 - 3" , Section 3.21.2.2, "PMSIXDATA[0-3]: Primary MSI-X Message Data Register 0 - 3" then the table and data registers should be programmed according to available vectors supported. E.g. If a single vector is only supported then all of the table and data registers should be programmed to the same address and data values. If two vectors are supported it could be programmed where two table and data registers point two one address with vector 0 and the other set of two table and data registers would be programmed to a different address and vector 1.

The same mapping exists for the NTB/RP configuration for the secondary side of the NTB but uses groups of SDOORBELL bits Table 98, "MSI-X Vector Handling and Processing by IIO on Secondary Side", Section 3.21.3.1, "SMSIXTBL[0-3]: Secondary MSI-X Table Address Register 0 - 3" , Section 3.21.3.2, "SMSIXDATA[0-3]: Secondary MSI-X Message Data Register 0 - 3" .

A bit has also been added if the OS cannot support four MSI-X vectors and there is no way to program the other table and data registers. This bit can be found in Section 3.19.3.23, "PPD: PCIE Port Definition" bit 5 for primary and Section 3.20.3.23, "DEVCAP2: PCI Express Device Capabilities Register 2" bit 0 for secondary. In this case the primary side PMSIXTBL0 and PMSIXDATA0 must be programmed. Hardware will then map all PDOORBELL bit(s) to this vector. The secondary side SMSIXTBL0 and SMSIXDATA0 must be programmed. Hardware will then map all SDOORBELL bit(s) to this vector.

3.17 RAS Capability and Error Handling

The NTB RAS capabilities is a superset of the RP RAS capabilites with the one additional capability. This capability is a counter to identify misses to the inbound memory windows. See Section 3.21.1.19, "USMEMMISS: Upstream Memory Miss" for details on this register.

3.18 Registers and Register Description

The NTB port has three distinct register sets. Primary side configuration registers, Secondary side configuration registers, and MMIO registers that span both sides of the NTB.

3.18.1 Additional Registers Outside of NTB Required (Per Stepping)

This section covers any registers needed to make the NTB operational that are not directly referenced in the sections below.

3.18.2 Known Errata (Per Stepping)

This section covers NTB bugs per stepping. This is intended to provide one location user can go to that list all known bugs.

A0 stepping:

PBAR01BASE, SB01BASE, Offset 5ECH (CBDF), Bits All. This register does not capture the correct values for the BDF so should not be used for debug. The returned value for the completer ID in the Complettion packet will be incorrect. This will not impact functional operation with Intel chipsets since we do not check this field in the completion packet at the receiver. Other RPs outside of Intel is unknown.

PBAR01BASE, SB01BASE, Offset 70CH (USMEMMISS), Bits All. This register should only increment upon a memory miss to the enabled NTB BARs. Bug is that it will also increment upon receiving each CFG, IO, and message in addition to a memory BAR miss.

Bus 0, Device 3, Function 0, Offset 06H (PCISTS), Bit 3 (INTx Status). In polling mode BDF 030, Offset 04H, Bit 10 (INTxDisable: Interrupt Disable) will be set = '1' (disabled) and SW will poll the PCISTS INTx Status bit to see if an interrupt occurred. This functionality is not working on A0 stepping the PCISTS INTx Status does not get set when INTxDisable is disabled. The user will need to directly poll the PDOORBELL register to see if an interrupt occured in polling mode.

Bus 0, Device 3, Function 0, Offset C0H (SPADSEMA4), Bit 0 (Scratchpad Semaphore). User should be able to just set bit 0 = '1' in ordere to clear the semaphore register. Instead the user must write FFFFH in order to clear the scratchpad semaphore register.

Bus 0, Device 3, Function 0, Offset 188H (MISCCTRLSTS), Bit 1 (Inbound Configuration enable). This bit must be set = '1' in NTB/RP mode in order for the secondary side of the NTB to accept inbound CFG cycles. This is need for the external RP to be able to program the secondary side of the NTB.

3.18.3 Bring Up Help

This section covers commmon issues in bring up.

Bus 0, Device 3, Function 0, Offset 04H (PCICMD), Bit 2:1 (Bus Master Enable and Memory Space Enable). In order to send memory transactions across the NTB both Bits 2:1 need to be set = "11" on both sides of the NTB. Explaination. NTB is back to back EPs. MAE controls memory transactions downstream and BME controls memory transactions upstream. Here is an example. CPU side1 sends memory transaction to side 2. MAE = 1 must be set on the primary side of the NTB to get the memory transaction downstream to the secondary side of the NTB. BME = 1 must be set on the secondary side of the NTB to get the memory transaction upstream to the attached RP. The same operation occurs for transactions going towards the CPU from the wire. MAE =1 on the secondary side of the NTB and BME = 1 on the primary side of the NTB.

3.19 PCI Express Configuration Registers (NTB Primary Side)

3.19.1 Configuration Register Map (NTB Primary Side)

This section covers the NTB primary side configuration space registers.

Bus 0, Device 3, Function 0 can function in three modes: PCI Express Root Port, NTB/NTB and NTB/RP. When configured as an NTB there are two sides to discuss for configuration registers. The primary side of the NTB's configuration space is located on Bus 0, Device 3, Function 0 with respect to the Intel® Xeon® processor C5500/C3500 series and a secondary side of the NTB's configuration space is located on some enumerated bus on another system and does not exist as configuration space on the local Intel® Xeon® processor C5500/C3500 series system anywhere.

The secondary side registers are discussed in Section 3.20, "PCI Express Configuration Registers (NTB Secondary Side)"

This section discusses the primary side registers.

Figure 61. PCI Express NTB (Device 3) Type0 Configuration Space

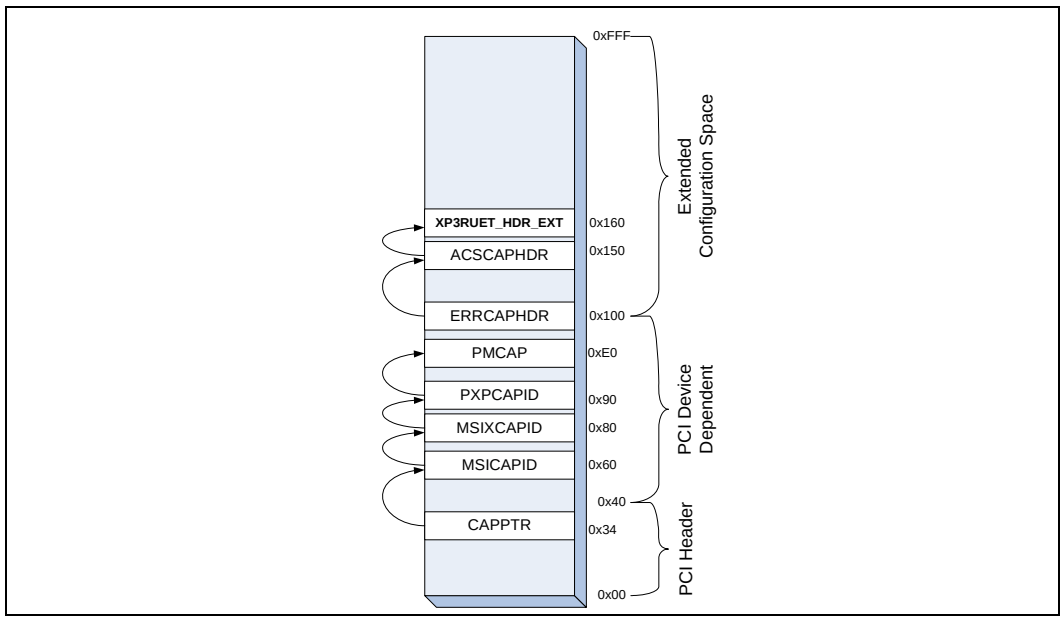


Figure 61 illustrates how each PCI Express port's configuration space appears to software. Each PCI Express configuration space has three regions:

- **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 IIO, the supported capabilities are:
  - — 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 IIO supports the Advanced Error Reporting Capability in this configuration space.

Table 88. IIO Bus 0 Device 3 Legacy Configuration Map (PCI Express Registers)

DID			VID	00h	MSIXMSGCTRL	MSIXNTPTR	MSIXCAPID	80h
PCISTS			PCICMD	04h	TABLEOFF_BIR			84h
CCR			RID	08h	PBAOFF_BIR			88h
BIST	HDR	PLAT	CLSR	0Ch				8Ch
PB01BASE				10h	PXPCAP	PXPNXTPTR	PXPCAPID	90h
				14h	DEVCAP			94h
PB23BASE				18h	DEVSTS	DEVCTRL		98h
				1Ch				9Ch
PB45BASE				20h				A0h
				24h				A4h
				28h				A8h
SID			SUBVID	2Ch				ACh
				30h				B0h
			CAPPTR	34h				B4h
				38h				B8h
MAXLAT	MINGNT	INTPIN	INTL	3Ch				BCh
				40h				C0h
				44h				C4h
				48h				C8h
				4Ch				CCh
				50h	SBAR45SZ	SBAR23SZ	PBAR45SZ	PBAR23SZ	D0h
				54h				PPD	D4h
				58h					D8h
				5Ch					DCh
MSICTRL		MSINTPTR	MSICAPID	60h	PMCAP				E0h
MSIAR				64h	PMCSR				E4h
MSIDR				68h					E8h
MSIMSK				6Ch					ECh
MSIPENDING				70h					F0h
				74h					F4h
				78h					F8h
				7Ch					FCh

Table 89. IIO Devices 3 Extended Configuration Map (PCI Express Registers) Page#0

VSECPHDR		100h	PERFCTRLSTS				180h
VSHDR		104h					184h
UNCERRSTS		108h	MISCCTRLSTS				188h
UNCERRMSK		10Ch					18Ch
UNCERRSEV		110h			PCIE_IOU_BIF_CTRL		190h
CORERRSTS		114h	NTBDEVCAP				194h
CORERRMSK		118h					198h
ERRCAP		11Ch	LNKCAP				19Ch
HDRLOG		120h	LNKSTS		LNKCON		1A0h
		124h	SLTCAP				1A4h
		128h	SLTSTS		SLTCON		1A8h
		12Ch			ROOTCON		1ACh
RPERRCMD		130h					1B0h
RPERRSTS		134h	DEVCAP2				1B4h
ERRSID		138h			DEVCTRL2		1B8h
SSMSK		13Ch					1BCh
APICLIMIT	APICBASE	140h	LNKSTS2		LNKCON2		1C0h
		144h					1C4h
		148h					1C8h
		14Ch					1CCh
ACSCAPHDR		150h					1D0h
ACSCTRL	ACSCAP	154h					1D4h
		158h					1D8h
		15Ch					1DCh
		160h	CTOCTRL				1E0h
		164h	PCIE_LER_SS_CTRLSTS				1E4h
		168h					1E8h
		16Ch					1ECh
		170h					1F0h
		174h					1F4h
		178h					1F8h
		17Ch					1FCh

Table 90. IIO Devices 3 Extended Configuration Map (PCI Express Registers) Page#1

XPCORERRSTS		200h	280h
XPCORERRMSK		204h	284h
XPUNCERRSTS		208h	288h
XPUNCERRMSK		20Ch	28Ch
XPUNCERRSEV		210h	290h
	XPUNCER RPTR	214h	294h
UNCEDMASK		218h	298h
COREDMASK		21Ch	29Ch
RPEDMASK		220h	2A0h
XPUNCEDMASK		224h	2A4h
XPCOREDMASK		228h	2A8h
		22Ch	2ACh
XPGLBERRPTR	XPGLBERRSTS	230h	2B0h
		234h	2B4h
		238h	2B8h
		23Ch	2BCh
		240h	2C0h
		244h	2C4h
		248h	2C8h
		24Ch	2CCh
		250h	2D0h
		254h	2D4h
		258h	2D8h
		25Ch	2DCh
		260h	2E0h
		264h	2E4h
		268h	2E8h
		26Ch	2ECh
		270h	2F0h
		274h	2F4h
		278h	2F8h
		27Ch	2FCh

3.19.2 Standard PCI Configuration Space (0x0 to 0x3F) - Type 0 Common Configuration Space

This section covers primary side registers in the 0x0 to 0x3F region that are common to Bus 0, Device 3. The secondary side of the NTB is discussed in the next section and is located on NTB Bus M, Device 0. 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.

Note: Several registers will be duplicated for device 3 in the three sections discussing the three modes it operates in RP, NTB/NTB, and NTB/RP primary and secondary but are repeated here for readability.

Note: Primary side configuration registers (device 3) can only be read by the local host.

3.19.2.1 VID: Vendor Identification Register

Register:VID
Bus:0
Device:3
Function:0
Offset:00h

Bit	Attr	Default	Description
15:0	**RO**	8086h	Vendor Identification Number The value is assigned by PCI-SIG to Intel.

3.19.2.2 DID: Device Identification Register (Dev#3, PCIE NTB Pri Mode)

Register:DID
Bus:0
Device:3
Function:0
Offset:02h

Bit	Attr	Default	Description
15:0	RO	3721h	**Device Identification Number** The value is assigned by Intel to each product. IIO will have a unique device id for each of its PCI Express single function devices. NTB/NTB = 3725h NTB/RP = 3726h Default value will show that of a RP until it is programmed to either NTB/NTB or NTB/RP.

3.19.2.3 PCICMD: PCI Command Register (Dev#3, PCIE NTB Pri Mode)

This register defines the PCI 3.0 compatible command register values applicable to PCI Express space.

Register:PCICMD
Bus:0
Device:3
Function:0
Offset:04h

Bit	Attr	Default	Description
15:11	RV	00h	*Reserved.* (by PCI SIG)
10	RW	0	**INTxDisable: Interrupt Disable** Controls the ability of the PCI-Express port to generate INTx messages. This bit does not affect the ability of Intel® Xeon® processor C5500/C3500 series to route interrupt messages received at the PCI-Express port. However, this bit controls the generation of legacy interrupts to the DMI for PCI-Express errors detected internally in this port (e.g. Malformed TLP, CRC error, completion time out etc.) or when receiving RP error messages or interrupts due to HP/PM events generated in legacy mode within Intel® Xeon® processor C5500/C3500 series. See the INTPIN register in Section 3.19.2.18, "INTPIN: Interrupt Pin Register" on page 186 for interrupt routing to DMI. 1: Legacy Interrupt mode is disabled 0: Legacy Interrupt mode is enabled
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/DMI 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 IIO 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 IIO 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 See the *PCI Express Base Specification*, Revision 2.0 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.
7	RO	0	**IDSEL Stepping / Wait Cycle Control** Not applicable to internal IIO devices. Hardwired to 0.
6	RW	0	**Parity Error Response** For PCI Express/DMI ports, IIO ignores this bit and always does ECC/parity checking and signaling for data/address of transactions both to and from IIO. This bit though affects the setting of bit 8 in the PCISTS register (see bit 8 in Section 3.19.2.4) .
5	RO	0	**VGA palette snoop Enable** Not applicable to PCI Express must be hardwired to 0.
4	RO	0	**Memory Write and Invalidate Enable** Not applicable to PCI Express must be hardwired to 0.

Register:PCICMD
Bus:0
Device:3
Function:0
Offset:04h

Bit	Attr	Default	Description
3	RO	0	**Special Cycle Enable** Not applicable to PCI Express must be hardwired to 0.
2	RW	0	**Bus Master Enable** When this bit is Set = 1b, the PCIE NTB will forward Memory Requests upstream from the secondary interface to the primary interface. When this bit is Cleared = 0b, the PCIE NTB will not forward Memory Requests from the secondary to the primary interface and will drop all posted memory write requests and will return Unsupported Requests UR for all non-posted memory read requests. ***Note:*** MSI/MSI-X interrupt Messages are in-band memory writes, setting the Bus Master Enable bit = 0b disables MSI/MSI-X interrupt Messages as well. Requests other than Memory or I/O Requests are not controlled by this bit. Default value of this bit is 0b.
1	RW	0	**Memory Space Enable** 1: Enables a PCI Express port's memory range registers to be decoded as valid target addresses for transactions from primary side. 0: Disables a PCI Express port's memory range registers (including the Configuration Registers range registers) to be decoded as valid target addresses for transactions from primary side.
0	RWL	0	**IO Space Enable** Controls a device's response to I/O Space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to I/O Space accesses. State after RST# is 0. NTB does not support I/O space accesses. Hardwired to 0 ***Note:*** This bit is locked and will appear as RO to SW

3.19.2.4 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 IIO bus.

Register:PCISTS
Bus:0
Device:3
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 (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.
14	RW1C	0	**Signaled System Error** 1: The device reported fatal/non-fatal (and not correctable) errors it detected on its PCI Express interface through the ERR[2:0] pins or message to PCH, with SERRE bit enabled. 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 PCH via a message. IIO 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 (IIO internal bus). 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: • Device receives a completion on the primary interface (internal bus of IIO) 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® QPI source address decoder. • Other master abort conditions detected on the IIO internal bus amongst those listed in Section 6.4.1, "Outbound Address Decoding" (IOH Platform Architecture Specification)
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 (IIO internal bus). 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: • Device receives a completion on the primary interface (internal bus of IIO) 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 the Intel® QPI that returns a failed completion status • Other completer abort conditions detected on the IIO internal bus amongst those listed in Section 6.4.2, "Inbound Address Decoding" (IOH Platform Architecture Specification).

Register:PCISTS
Bus:0
Device:3
Function:0
Offset:06h

Bit	Attr	Default	Description
11	RW1C	0	**Signaled Target Abort** This bit is set when the NTB port forwards a completer abort (CA) completion status from the secondary interface to the primary interface.
10:9	RO	0h	**DEVSEL# Timing** Not applicable to PCI Express. Hardwired to 0.
8	RW1C	0	**Master Data Parity Error** This bit is set if the **Parity Error Response** bit in the PCI Command register is set and the • Requestor receives a poisoned completion on the primary interface or • Requestor forwards a poisoned write request (including MSI/MSI-X writes) from the secondary interface to the primary interface.
7	RO	0	**Fast Back-to-Back** Not applicable to PCI Express. Hardwired to 0.
6	RO	0	*Reserved*
5	RO	0	**66MHz capable** Not applicable to PCI Express. Hardwired to 0.
4	RO	1	**Capabilities List** This bit indicates the presence of a capabilities list structure
3	RO	0	**INTx Status** When Set, indicates that an INTx emulation interrupt is pending internally in the Function.
2:0	RV	0h	*Reserved*

3.19.2.5 RID: Revision Identification Register

This register contains the revision number of the IIO. The revision number steps the same across all devices and functions i.e. individual devices do not step their RID independently.

The IIO supports the CRID feature where in this register's value can be changed by the BIOS. See Section 3.2.2, "Compatibility Revision ID" in Volume 2 of the Datasheet for details.

Register:RID
Bus:0
Device:3
Function:0
Offset:08h

Bit	Attr	Default	Description
7:4	RWO	0	**Major Revision** Steppings which require all masks to be regenerated. 0: A stepping 1: B stepping
3:0	RWO	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

3.19.2.6 CCR: Class Code Register

This register contains the Class Code for the device.

Register:CCR
Bus:0
Device:3
Function:0
Offset:09h

Bit	Attr	Default	Description
23:16	RO	06h	**Base Class** For PCI Express NTB port this field is hardwired to 06h, indicating it is a "Bridge Device".
15:8	RO	80h	**Sub-Class** For PCI Express NTB port, this field hardwired to 80h to indicate a "Other bridge type".
7:0	RO	00h	**Register-Level Programming Interface** This field is hardwired to 00h for PCI Express NTB port.

3.19.2.7 CLSR: Cacheline Size Register

Register:CLSR Bus:0 Device:3 Function:0 Offset:0Ch			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0h	**Cacheline Size** This register is set as RW for compatibility reasons only. Cacheline size for IIO is always 64B. IIO hardware ignore this setting.

3.19.2.8 PLAT: Primary Latency Timer

This register denotes the maximum time slice for a burst transaction in legacy PCI 2.3 on the primary interface. It does not affect/influence PCI-Express functionality.

Register:PLAT Bus:0 Device:3 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.

3.19.2.9 HDR: Header Type Register (Dev#3, PCIe NTB Pri Mode)

This register identifies the header layout of the configuration space.

Register:HDR Bus:0 Device:3 Function:0 Offset:0Eh PCIE_ONLY			
Bit	**Attr**	**Default**	**Description**
7	RO	0	**Multi-function Device** This bit defaults to 0 for PCI Express NTB port.
6:0	RO	00h	**Configuration Layout** This field identifies the format of the configuration header layout. It is Type0 for PCI Express NTB port. The default is 00h, indicating a “non-bridge function”.

3.19.2.10 BIST: Built-In Self Test

This register is used for reporting control and status information of BIST checks within a PCI Express port. It is not supported by Intel® Xeon® processor C5500/C3500 series.

Register:BIST
Bus:0
Device:3
Function:0
Offset:0Fh

Bit	Attr	Default	Description
7:0	RO	0h	BIST_TST: BIST Tests Not supported. Hardwired to 00h

3.19.2.11 PB01BASE: Primary BAR 0/1 Base Address

This register is used to setup the primary side NTB configuration space

Note: SW must program upper DW first and then lower DW. If lower DW is programmed first HW will clear the lower DW.

Register:PB01BASE
Bus:0
Device:3
Function:0
Offset:10h

Bit	Attr	Default	Description
63:16	RW	00h	**Primary BAR 0/1 Base** Sets the location of the BAR written by SW on a 64KB alignment
15:04	RO	00h	**Reserved** Fixed size of 64KB.
03	RO	1b	**Prefetchable** BAR points to Prefetchable memory.
02:01	RO	10b	**Type** Memory type claimed by BAR 0/1is 64-bit addressable.
00	RO	0b	**Memory Space Indicator** BAR resource is memory (as opposed to I/O).

3.19.2.12 PB23BASE: Primary BAR 2/3 Base Address

The register is used by the processor on the primary side of the NTB to setup a 64b prefetchable memory window.

Note: SW must program upper DW first and then lower DW. If lower DW is programmed first HW will clear the lower DW.

Register:PB23BASE
Bus:0
Device:3
Function:0
Offset:18h

Bit	Attr	Default	Description
63:nn	RWL	00h	**Primary BAR 2/3 Base** Sets the location of the BAR written by SW ***Notes:*** • The "nn" indicates the least significant bit that is writable. The number of bits that are writable in this register is dictated by the value loaded into the PBAR23SZ register by the BIOS at initialization time (before BIOS PCI enumeration)." • For the special case where PBAR23SZ = '0', bits 63:00 are all RO='0' resulting in the BAR being disabled. • These bits will appear to SW as RW.
(nn-1) : 12	RWL	00h	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.19.3.19, "PBAR23SZ: Primary BAR 2/3 Size" Granularity must be at least 4 KB. ***Notes:*** • For the special case where PBAR23SZ = '0', bits 63:00 are all RO='0' resulting in the BAR being disabled. • These bits will appear to SW as RO.
11:04	RO	00h	**Reserved**
03	RO	1b	**Prefetchable** BAR points to Prefetchable memory.
02:01	RO	10b	**Type** Memory type claimed by BAR 2/3 is 64-bit addressable.
00	RO	0b	**Memory Space Indicator** BAR resource is memory (as opposed to I/O).

3.19.2.13 PB45BASE: Primary BAR 4/5 Base Address

The register is used by the processor on the primary side of the NTB to setup a second 64b prefetchable memory window.

Note: SW must program upper DW first and then lower DW. If lower DW is programmed first HW will clear the lower DW.

Register:PB45BASE
Bus:0
Device:3
Function:0
Offset:20h

Bit	Attr	Default	Description
63:nn	RWL	00h	**Primary BAR 4/5 Base** Sets the location of the BAR written by SW ***Notes:*** • The "nn" indicates the least significant bit that is writable. The number of bits that are writable in this register is dictated by the value loaded into the PBAR45SZ register by the BIOS at initialization time (before BIOS PCI enumeration)." • For the special case where PBAR45SZ = '0', bits 63:00 are all RO='0' resulting in the BAR being disabled. • These bits will appear to SW as RW.
(nn-1) : 12	RWL	00h	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.19.3.20, "PBAR45SZ: Primary BAR 4/5 Size" Granularity must be at least 4 KB. ***Notes:*** • For the special case where PBAR45SZ = '0', bits 63:00 are all RO='0' resulting in the BAR being disabled. • These bits will appear to SW as RO.
11:04	RO	00h	Reserved Granularity must be at least 4 KB.
03	RO	1b	**Prefetchable** BAR points to Prefetchable memory.
02:01	RO	10b	**Type** Memory type claimed by BAR 4/5 is 64-bit addressable.
00	RO	0b	**Memory Space Indicator** BAR resource is memory (as opposed to I/O).

3.19.2.14 SUBVID: Subsystem Vendor ID (Dev#3, PCIE NTB Pri Mode)

This register identifies the vendor of the subsystem.

Register:SUBVID
Bus:0
Device:3
Function:0
Offset:2Ch

Bit	Attr	Default	Description
15:0	RWO	0000h	**Subsystem Vendor ID:** This field must be programmed during boot-up to indicate the vendor of the system board. When any byte or combination of bytes of this register is written, the register value locks and cannot be further updated.

3.19.2.15 SID: Subsystem Identity (Dev#3, PCIE NTB Pri Mode)

This register identifies a particular subsystem.

Register:SID
Bus:0
Device:3
Function:0
Offset:2Eh

Bit	Attr	Default	Description
15:0	RWO	0000h	**Subsystem ID:** This field must be programmed during BIOS initialization. When any byte or combination of bytes of this register is written, the register value locks and cannot be further updated.

3.19.2.16 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.

Register:CAPPTR
Bus:0
Device:3
Function:0
Offset:34h

Bit	Attr	Default	Description
7:0	RWO	60h	**Capability Pointer** Points to the first capability structure for the device.

3.19.2.17 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 RW for compatibility purposes.

Register:INTL
Bus:0
Device:3
Function:0
Offset:3Ch

Bit	Attr	Default	Description
7:0	RW	00h	**Interrupt Line** This bit is RW for devices that can generate a legacy INTx message and is needed only for compatibility purposes.

3.19.2.18 INTPIN: Interrupt Pin Register

The INTPIN register identifies legacy interrupt INTx support.

Register:INTPIN
Bus:0
Device:3
Function:0
Offset:3Dh

Bit	Attr	Default	Description
7:0	RWO	01h	**INTP: Interrupt Pin** This field defines the type of interrupt to generate for the PCI-Express port. 001: Generate INTA Others: Reserved BIOS/configuration Software has the ability to program this register once during boot to set up the correct interrupt for the port.

3.19.2.19 MINGNT: Minimum Grant Register

Register:MINGNT
Bus:0
Device:3
Function:0
Offset:3Eh

Bit	Attr	Default	Description
7:0	RO	00h	**Minimum Grant**: This register does not apply to PCI Express. It is hard-coded to “00”h.

3.19.2.20 MAXLAT: Maximum Latency Register

Register:MAXLAT
Bus:0
Device:3
Function:0
Offset:3Fh

Bit	Attr	Default	Description
7:0	RO	00h	**Maximum Latency**: This register does not apply to PCI Express. It is hard-coded to “00”h.

3.19.3 Device-Specific PCI Configuration Space - 0x40 to 0xFF

3.19.3.1 MSICAPID: MSI Capability ID

Register:MSICAPID
Bus:0
Device:3
Function:0
Offset:60h

Bit	Attr	Default	Description
7:0	RO	05h	**Capability ID** Assigned by PCI-SIG for MSI.

3.19.3.2 MSINXTPTR: MSI Next Pointer

Register:MSINXTPTR
Bus:0
Device:3
Function:0
Offset:61h

Bit	Attr	Default	Description
7:0	RWO	80h	**Next Ptr** This field is set to 80h for the next capability list (PCI Express capability structure) in the chain.

3.19.3.3 MSICTRL: MSI Control Register

Register:MSICTRL
Bus:0
Device:3
Function:0
Offset:62h

Bit	Attr	Default	Description
15:9	RV	00h	*Reserved.*
8	RO	1b	**Per-vector masking capable** This bit indicates that PCI Express ports support MSI per-vector masking.
7	RO	0b	**64-bit Address Capable** A PCI Express Endpoint must support the 64-bit Message Address version of the MSI Capability structure 1: Function is capable of sending 64-bit message address 0: Function is not capable of sending 64-bit message address.

Register:MSICTRL
Bus:0
Device:3
Function:0
Offset:62h

Bit	Attr	Default	Description
6:4	RW	000b	**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. See Table 91 for discussion on how the interrupts are distributed amongst the various sources of interrupt based on the number of messages allocated by software for the PCI Express NTB port. Value Number of Messages Requested 000b = 1 001b = 2 010b = 4 011b = 8 100b = 16 101b = 32 110b = Reserved 111b = Reserved
3:1	RO	001b	**Multiple Message Capable** IIO's PCI Express port supports two messages for all internal events. Value Number of Messages Requested 000b = 1 001b = 2 010b = 4 011b = 8 100b = 16 101b = 32 110b = Reserved 111b = Reserved
0	RW	0b	**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 PCI Express port to use MSI messages for RAS, provided bit 4 in Section 3.19.4.20, "MISCCTRLSTS: Misc. Control and Status Register" on page 216 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 3.19.4.20, "MISCCTRLSTS: Misc. Control and Status Register" on page 216) for details. ***Note:*** Software must disable INTx and MSI-X for this device when using MSI

3.19.3.4 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 broken into its constituent fields.

Register:MSIAR
Bus:0
Device:3
Function:0
Offset:64h

Bit	Attr	Default	Description
31:20	RW	0h	**Address MSB** This field specifies the 12 most significant bits of the 32-bit MSI address. This field is R/W.
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 IA32 processor .
3	RW	0h	**Address Redirection Hint** 0: directed 1: redirectable
2	RW	0h	**Address Destination Mode** 0: physical 1: logical
1:0	RO	0h	*Reserved.*

3.19.3.5 MSIDR: MSI Data Register

The MSI Data Register contains all the data (interrupt vector) related to MSI interrupts from the root ports.

Register:MSIDR
Bus:0
Device:3
Function:0
Offset:68h

Bit	Attr	Default	Description
31:16	RO	0000h	*Reserved.*
15	RW	0h	**Trigger Mode** 0 - Edge Triggered 1 - Level Triggered The IIO does nothing with this bit other than passing it along to the Intel® QPI
14	RW	0h	**Level** 0 - Deassert 1 - Assert The IIO does nothing with this bit other than passing it along to the Intel® QPI
13:12	RW	0h	**Don't care for IIO**
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 – *Reserved* 0111 – ExtINT - Not supported via MSI of root port 1000-1111 - *Reserved*
7:0	RW	0h	**Interrupt Vector** The interrupt vector (LSB) will be modified by the IIO to provide context sensitive interrupt information for different events that require attention from the processor. e.g Hot plug, Power Management and RAS error events. Depending on the number of Messages enabled by the processor, Table 91 illustrates how the IIO distributes these vectors.

Table 91. MSI Vector Handling and Processing by IIO on Primary Side

Number of Messages enabled by Software	Events	IV[7:0]
1	All	xxxxxxxx[1]
2	HP, PD[15:00]	xxxxxxx0
	AER	xxxxxxx1

1. The term "xxxxxx" in the Interrupt vector denotes that software initializes them and IIO will not modify any of the "x" bits except the LSB as indicated in the table as a function of MMEN

3.19.3.6 MSIMSK: MSI Mask Bit Register

The Mask Bit register enables software to disable message sending on a per-vector basis.

Register:MSIMSK
Bus:0
Device:3
Function:0
Offset:6Ch

Bit	Attr	Default	Description
31:02	RsvdP	0h	*Reserved*
01:00	RW	00b	**Mask Bits** For each Mask bit that is set, the PCI Express port is prohibited from sending the associated message. NTB supports up to 2 messages Corresponding bits are masked if set to '1'

3.19.3.7 MSIPENDING: MSI Pending Bit Register

The Mask Pending register enables software to defer message sending on a per-vector basis.

Register:MSIPENDING
Bus:0
Device:3
Function:0
Offset:70h

Bit	Attr	Default	Description
31:02	RsvdP	0h	*Reserved*
01:00	RO	0h	**Pending Bits** For each Pending bit that is set, the PCI Express port has a pending associated message. NTB supports up to 2 messages Corresponding bits are pending if set to '1'

3.19.3.8 MSIXCAPID: MSI-X Capability ID

Register:MSIXCAPID
Bus:0
Device:3
Function:0
Offset:80h

Bit	Attr	Default	Description
7:0	RO	11h	**Capability ID** Assigned by PCI-SIG for MSI-X.

3.19.3.9 MSIXNXTPTR: MSI-X Next Pointer

Register:MSIXNXTPTR
Bus:0
Device:3
Function:0
Offset:81h

Bit	Attr	Default	Description
7:0	RWO	90h	**Next Ptr** This field is set to 90h for the next capability list (PCI Express capability structure) in the chain.

3.19.3.10 MSIXMSGCTRL: MSI-X Message Control Register

Register:MSIXMSGCTRL
Bus:0
Device:3
Function:0
Offset:82h

Bit	Attr	Default	Description
15	RW	0b	**MSI-X Enable** Software uses this bit to enable MSI-X method for signaling 0: NTB is prohibited from using MSI-X to request service 1: MSI-X method is chosen for NTB interrupts ***Note:*** Software must disable INTx and MSI for this device when using MSI-X
14	RW	0b	**Function Mask** If = 1b, all the vectors associated with the NTB are masked, regardless of the per vector mask bit state. If = 0b, each vector's mask bit determines whether the vector is masked or not. Setting or clearing the MSI-X function mask bit has no effect on the state of the per-vector Mask bit.
13:11	RO	0h	*Reserved.*
10:00	RO	003h	**Table Size** System software reads this field to determine the MSI-X Table Size N, which is encoded as N-1. For example, a returned value of "00000000011" indicates a table size of 4. The value in this field depends on the setting of Section 3.19.3.23, "PPD: PCIE Port Definition" bit 5. When PPD, bit 5 = '0' (default) Table size is 4, encoded as a value of 003h When PPD, bit 5 = '1' Table size is 1, encoded as a value of 000h

3.19.3.11 TABLEOFF_BIR: MSI-X Table Offset and BAR Indicator Register (BIR)

Register default: 00002000h

Register:TABLEOFF_BIR Bus:0 Device:3 Function:0 Offset:84h			
Bit	**Attr**	**Default**	**Description**
31:03	RO	00000400h	**Table Offset** MSI-X Table Structure is at offset 8K from the PB01BASE address. See Section 3.19.3.13, "PXPCAPID: PCI Express Capability Identity Register" for the start of details relating to MSI-X registers.
02:00	RO	0h	**Table BIR** Indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X Table into Memory Space. **BIR Value Base Address register** 0 10h 1 14h 2 18h 3 1Ch 4 20h 5 24h 6 Reserved 7 Reserved For a 64-bit Base Address register, the Table BIR indicates the lower DWORD.

3.19.3.12 PBAOFF_BIR: MSI-X Pending Array Offset and BAR Indicator

Register default: 00003000h

Register:PBAOFF_BIR Bus:0 Device:3 Function:0 Offset:88h			
Bit	**Attr**	**Default**	**Description**
31:03	RO	00000600h	**Table Offset** MSI-X PBA Structure is at offset 12K from the PB01BASE BAR address. Section 3.21.2.4, "PMSIXPBA: Primary MSI-X Pending Bit Array Register" for details
02:00	RO	0h	**PBA BIR** Indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X Table into Memory Space. **BIR Value Base Address register** 0 10h 1 14h 2 18h 3 1Ch 4 20h 5 24h 6 Reserved 7 Reserved For a 64-bit Base Address register, the Table BIR indicates the lower DWORD.

3.19.3.13 PXPCAPID: PCI Express Capability Identity Register

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 3.0 configuration space.

Register:PXPCAPID
Bus:0
Device:3
Function:0
Offset:90h

Bit	Attr	Default	Description
7:0	RO	10h	**Capability ID** Provides the PCI Express capability ID assigned by PCI-SIG. Required by *PCI Express Base Specification*, Revision 2.0 to be this value.

3.19.3.14 PXPNXTPTR: PCI Express Next Pointer Register

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 3.0 configuration space.

Register:PXPNXTPTR
Bus:0
Device:3
Function:0
Offset:91h

Bit	Attr	Default	Description
7:0	RWO	E0h	**Next Ptr** This field is set to the PCI PM capability.

3.19.3.15 PXPCAP: PCI Express Capabilities Register

The PCI Express Capabilities register identifies the PCI Express device type and associated capabilities.

Register:PXPCAP
Bus:0
Device:3
Function:0
Offset:92h

Bit	Attr	Default	Description
15:14	Rsvd P	00b	*Reserved*
13:9	RO	00000b	**Interrupt Message Number** Applies only to the RPs. This field indicates the interrupt message number that is generated for PM/HP events. When there are more than one MSI/MSI-X interrupt Number, this register field is required to contain the offset between the base Message Data and the MSI/MSI-X Message that is generated when the status bits in the slot status register or RP status registers are set. IIO assigns the first vector for PM/HP events and so this field is set to 0.
8	RWO	0b	**Slot Implemented** Applies only to the RPs for NTB this value is kept at 0b. 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	0000b	**Device/Port Type** This field identifies the type of device. 0000b = PCI Express Endpoint.
3:0	RWO	2h	**Capability Version** This field identifies the version of the PCI Express capability structure. Set to 2h for PCI Express devices for compliance with the extended base registers.

3.19.3.16 DEVCAP: PCI Express Device Capabilities Register

The PCI Express Device Capabilities register identifies device specific information for the device.

Register:DEVCAP
Bus:0
Device:3
Function:0
Offset:94h

Bit	Attr	Default	Description
31:29	Rsvd P	0h	*Reserved*
28	RO	0b	**Function Level Reset Capability** A value of 1b indicates the Function supports the optional Function Level Reset mechanism. NTB does not support this functionality.
27:26	RO	0h	**Captured Slot Power Limit Scale** Does not apply to RPs or integrated devices This value is hardwired to 00h NTB is required to be able to receive the Set_Slot_Power_Limit message without error but simply discard the Message value. ***Note:*** *PCI Express Base Specification,* Revision 2.0 states Components with Endpoint, Switch, or PCI Express-PCI Bridge Functions that are targeted for integration on an adapter where total consumed power is below the lowest limit defined for the targeted form factor are permitted to ignore Set_Slot_Power_Limit Messages, and to return a value of 0 in the Captured Slot Power Limit Value and Scale fields of the Device Capabilities register
25:18	RO	00h	**Captured Slot Power Limit Value** Does not apply to RPs or integrated devices This value is hardwired to 00h NTB is required to be able to receive the Set_Slot_Power_Limit message without error but simply discard the Message value. ***Note:*** The *PCI Express Base Specification,* Revision 2.0 states that components with endpoint, switch, or PCI Express-PCI Bridge functions that are targeted for integration on an adapter where total consumed power is below the lowest limit defined for the targeted form factor are permitted to ignore Set_Slot_Power_Limit Messages, and to return a value of 0 in the Captured Slot Power Limit Value and Scale fields of the Device Capabilities register
17:16	Rsvd P	0h	*Reserved*
15	RO	1	**Role Based Error Reporting**: IIO is 1.1 compliant and so supports this feature
14	RO	0	**Power Indicator Present on Device** Does not apply to RPs or integrated devices
13	RO	0	**Attention Indicator Present** Does not apply to RPs or integrated devices
12	RO	0	**Attention Button Present** Does not apply to RPs or integrated devices
11:9	RO	000	**Endpoint L1 Acceptable Latency** Does not apply to IIO RCiEP (Link does not exist between host and RCiEP)

Register:DEVCAP
Bus:0
Device:3
Function:0
Offset:94h

Bit	Attr	Default	Description
8:6	RO	000	**Endpoint L0s Acceptable Latency** Does not apply to IIO RCiEP (Link does not exist between host and RCiEP)
5	RO	1	**Extended Tag Field Supported** IIO devices support 8-bit tag 1 = Maximum Tag field is 8 bits 0 = Maximum Tag field is 5 bits
4:3	RO	00b	**Phantom Functions Supported** IIO does not support phantom functions. 00b = No Function Number bits are used for Phantom Functions
2:0	RO	001b	**Max Payload Size Supported** IIO supports 256B payloads on PCI Express ports 001b = 256 bytes max payload size

3.19.3.17 DEVCTRL: PCI Express Device Control Register (Dev#3, PCIE NTB Pri Mode)

The PCI Express Device Control register controls PCI Express specific capabilities parameters associated with the device.

Register:DEVCTRL
Bus:0
Device:3
Function:0
Offset:98h
PCIE_ONLY

Bit	Attr	Default	Description
15	RsvdP	0h	*Reserved.*
14:12	RO	000	**Max_Read_Request_Size** This field sets maximum Read Request size generated by the Intel® Xeon® processor C5500/C3500 series as a requestor. The corresponding IOU logic in the Intel® Xeon® processor C5500/C3500 series associated with the PCIExpress port must not generate read requests with size exceeding the set value. 000: 128B max read request size 001: 256B max read request size 010: 512B max read request size 011: 1024B max read request size 100: 2048B max read request size 101: 4096B max read request size 110: Reserved 111: Reserved ***Note:*** The Intel® Xeon® processor C5500/C3500 series will not generate read requests larger than 64B on the outbound side due to the internal Micro-architecture (CPU initiated, DMA, or Peer to Peer). Hence the field is set to 000b encoding.
11	RO	0	**Enable No Snoop** Not applicable since the NTB is never the originator of a TLP. This bit has no impact on forwarding of NoSnoop attribute on peer requests.
10	RO	0	**Auxiliary Power Management Enable** Not applicable to IIO
9	RO	0	**Phantom Functions Enable** Not applicable to IIO since it never uses phantom functions as a requester.
8	RW	0h	**Extended Tag Field Enable** This bit enables the PCI Express/DMI ports to use an 8-bit Tag field as a requester.
7:5	RW	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 IIO must handle TLPs as large as the set value. As a requester (i.e. for requests where IIOs 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 DMI port aliases to 128B) others: alias to 128B This field is RW for PCI Express ports. ***Note:*** Bit 7:5 must be programmed to the same value on both primary and secondary side of the NTB

Register:DEVCTRL
Bus:0
Device:3
Function:0
Offset:98h
PCIE_ONLY

Bit	Attr	Default	Description
4	RO	0	**Enable Relaxed Ordering** Not applicable since the NTB is never the originator of a TLP. 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 RP/PCI Express NTB secondary interface/DMI ports. This bit controls the reporting of unsupported requests that IIO itself detects on requests its receives from a PCI Express/DMI port. 0: Reporting of unsupported requests is disabled 1: Reporting of unsupported requests is enabled.
2	RW	0	**Fatal Error Reporting Enable** Applies only to the PCI Express RP/PCI Express NTB secondary interface/DMI ports. Controls the reporting of fatal errors that IIO detects on the PCI Express/DMI interface. 0: Reporting of Fatal error detected by device is disabled 1: Reporting of Fatal error detected by device is enabled
1	RW	0	**Non Fatal Error Reporting Enable** Applies only to the PCI Express RP/PCI Express NTB secondary interface/DMI ports. Controls the reporting of non-fatal errors that IIO detects on the PCI Express/DMI interface. 0: Reporting of Non Fatal error detected by device is disabled 1: Reporting of Non Fatal error detected by device is enabled
0	RW	0	**Correctable Error Reporting Enable** Applies only to the PCI Express RP/PCI Express NTB secondary interface/DMI ports. Controls the reporting of correctable errors that IIO detects on the PCI Express/DMI interface 0: Reporting of link Correctable error detected by the port is disabled 1: Reporting of link Correctable error detected by port is enabled

3.19.3.18 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
Bus:0
Device:3
Function:0
Offset: 9Ah

Bit	Attr	Default	Description
15:6	RsvdZ	000h	*Reserved.*
5	RO	0h	**Transactions Pending**: Does not apply. Bit is hardwired to 0 NTB is a special case bridging device following the rule below. The *PCI Express Base Specification,* Revision 2.0 states. Root and Switch Ports implementing only the functionality required by this document do not issue Non-Posted Requests on their own behalf, and therefore are not subject to this case. Root and Switch Ports that do not issue Non-Posted Requests on their own behalf hardwire this bit to 0b.
4	RO	0	**AUX Power Detected** Does not apply to IIO.
3	RW1C	0	**Unsupported Request Detected** This bit applies only to the root/DMI ports.This bit indicates that the NTB primary 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 RP received and it detected them as unsupported requests (e.g. address decoding failures that the RP detected on a packet, receiving inbound lock reads, BME bit is clear etc.). This bit is not set on peer2peer completions with UR status that are forwarded by the RP to the PCIE link. 0: No unsupported request detected by the RP
2	RW1C	0	**Fatal Error Detected** This bit indicates that a fatal (uncorrectable) error is detected by the NTB primary 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
1	RW1C	0	**Non Fatal Error Detected** This bit gets set if a non-fatal uncorrectable error is detected by the NTB primary 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 NTB primary 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

3.19.3.19 PBAR23SZ: Primary BAR 2/3 Size

This register contains a value used to set the size of the memory window requested by the 64-bit BAR 2/3 pair for the Primary side of the NTB.

Register:PBAR23SZ
Bus:0
Device:3
Function:0
Offset:0D0h

Bit	Attr	Default	Description
7:0	RWO	00h	**Primary BAR 2/3 Size** Value indicating the size of 64-bit BAR 2/3 pair on the Primary side of the NTB. This value is loaded by BIOS prior to enumeration. The value indicates the number of bits that will be Read-Only (returning 0 when read regardless of the value written to them) during PCI enumeration. Only legal settings are 12- 39, representing BAR sizes of 2^{12} (4KB) through 2^{39} (512GB) are valid. ***Note:*** Programming a value of '0' or any other value other than (12-39) will result in the BAR being disabled.

3.19.3.20 PBAR45SZ: Primary BAR 4/5 Size

This register contains a value used to set the size of the memory window requested by the 64-bit BAR 4/5 pair for the Primary side of the NTB.

Register:PBAR45SZ
Bus:0
Device:3
Function:0
Offset:0D1h

Bit	Attr	Default	Description
7:0	RWO	00h	**Primary BAR 4/5 Size** Value indicating the size of 64-bit BAR 2/3 pair. This value is loaded by BIOS prior to enumeration. The value indicates the number of bits that will be Read-Only (returning 0 when read regardless of the value written to them) during PCI enumeration. Only legal settings are 12- 39, representing BAR sizes of 2^{12} (4KB) through 2^{39} (512GB) are valid. ***Note:*** Programming a value of '0' or any other value other than (12-39) will result in the BAR being disabled.

3.19.3.21 SBAR23SZ: Secondary BAR 2/3 Size

This register contains a value used to set the size of the memory window requested by the 64-bit BAR 2/3 pair for the Secondary side of the NTB.

Register:SBAR23SZ
Bus:0
Device:3
Function:0
Offset:0D2h

Bit	Attr	Default	Description
7:0	RWO	00h	**Secondary BAR 2/3 Size** Value indicating the size of 64-bit BAR 2/3 pair on the Secondary side of the NTB. This value is loaded by BIOS prior to enumeration. The value indicates the number of bits that will be Read-Only (returning 0 when read regardless of the value written to them) during PCI enumeration. Only legal settings are 12- 39, representing BAR sizes of 2^{12} (4 KB) through 2^{39} (512 GB) are valid. ***Note:*** Programming a value of '0' or any other value other than (12-39) will result in the BAR being disabled.

3.19.3.22 SBAR45SZ: Secondary BAR 4/5 Size

This register contains a value used to set the size of the memory window requested by the 64-bit BAR 4/5 on the secondary side of the NTB.

Register:SBAR45SZ
Bus:0
Device:3
Function:0
Offset:0D3h

Bit	Attr	Default	Description
7:0	RWO	00h	**Secondary BAR 4/5 Size** Value indicating the size of 64-bit BAR 2/3 pair on the Secondary side of the NTB. This value is loaded by BIOS prior to enumeration. The value indicates the number of bits that will be Read-Only (returning 0 when read regardless of the value written to them) during PCI enumeration. Only legal settings are 12- 39, representing BAR sizes of 2^{12} (4 KB) through 2^{39} (512 GB) are valid. ***Note:*** Programming a value of '0' or any other value other than (12-39) will result in the BAR being disabled.

3.19.3.23 PPD: PCIE Port Definition

This register defines the behavior of the PCIE port which can be either a RP, NTB connected to another NTB or an NTB connected to a Root Complex.

This register is used to set the value in the DID register on the Primary side of the NTB (located at offset 02h). This value is loaded by BIOS prior to running PCI enumeration.

Register:PPD
Bus:0
Device:3
Function:0
Offset:0D4h

Bit	Attr	Default	Description
07:06	RO	0h	**Reserved**
05	RW	0b	**NTB Primary side - MSI-X Single Message Vector:** This bit when set, causes only a single MSI-X vector to be generated if MSI-X is enabled. This bit affects the default value of the MSI-X Table Size field in the Section 3.19.3.10, "MSIXMSGCTRL: MSI-X Message Control Register"
04	RO	0h	**Crosslink Configuration Status**: This bit is written by hardware and shows the result of the PE_NTBXL strap combined with the crosslink control override settings. 0 = NTB port is configured as DSD/USP 1 = NTB port is configured as USD/DSP
03:02	RW	00b	**Crosslink Control Override**: When bit 3 of this register is set, the NTB logic ignores the setting of the external pin strap (PE_NTBXL) and directly forces the polarity of the NTB port to be either an Upstream Device (USD) or Downstream Device (DSD) based on the setting of bit 2. 11 - Force NTB port to USD/DSP; NTB ignores input from PE_NTBXL 10 - Force NTB port to DSD/USP; NTB ignores input from PE_NTBXL 01 - Reserved 00 - Use external pin (PE_NTBXL) only to determine USD or DSD (default) ***Note:*** Bits 03:02 of this register only have meaning when bits 01:00 of this same register are programmed as "01"b (NTB/NTB). When configured as NTB/RP hardware directly sets port to DSD/USP so this field is not required. ***Note:*** When using crosslink control override, the external strap PECFGSEL[2:0] must be set to "100"b (Wait-on-BIOS). The BIOS can then come and set this field and then enable the port. ***Note:*** In applications that are DP configuration, and having an external controller set up the crosslink control override through the SMBus master interface. PECFGSEL[2:0] must be set to "100"b (Wait-on-BIOS) on both chipsets. The external controller on the master can then set the crosslink control override field on both chipsets and then enable the ports on both chipsets.
01:00	RW	00b	**Port Definition** Value indicating the value to be loaded into the DID register (offset 02h). 00b - Transparent bridge 01b - 2 NTBs connected back to back 10b - NTB connected to a RP 11b - Reserved ***Note:*** When the DISNTSPB fuse is blown this field becomes RO "00"

3.19.3.24 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
Bus:0
Device:3
Function:0
Offset:E0h

Bit	Attr	Default	Description
31:27	RO	00000b	PME Support Indicates the PM states within which the function is capable of sending a PME message. NTB primary side does not forward PME messages. Bit 31 = D3cold Bit 30 = D3hot Bit 29 = D2 Bit 28 = D1 Bit 27 = D0
26	RO	0b	D2 Support IIO does not support power management state D2.
25	RO	0b	D1 Support IIO does not support power management state D1.
24:22	RO	000b	AUX Current Device does not support auxiliary current
21	RO	0b	Device Specific Initialization Device initialization is not required
20	RV	0b	*Reserved*.
19	RO	0b	PME Clock This field is hardwired to 0h as it does not apply to PCI Express.
18:16	RO	011b	Version This field is set to 3h (PM 1.2 compliant) as version number for all PCI Express ports.
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.

3.19.3.25 PMCSR: Power Management Control and Status Register

This register provides status and control information for PM events in the PCI Express port of the IIO.

Register:PMCSR
Bus:0
Device:3
Function:0
Offset:E4h

Bit	Attr	Default	Description
31:24	RO	00h	**Data** Not relevant for IIO
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	RsvdP	0h	*Reserved.*
15	RW1CS	0h	**PME Status** Applies only to RPs This bit has no meaning for NTB
14:13	RO	0h	**Data Scale** Not relevant for IIO
12:9	RO	0h	**Data Select** Not relevant for IIO
8	RWS	0h	**PME Enable** Applies only to RPs. 0: Disable ability to send PME messages when an event occurs 1: Enables ability to send PME messages when an event occurs This bit has no meaning for NTB
7:4	RsvdP	0h	*Reserved.*
3	RWO	1	No Soft Reset Indicates IIO does not reset its registers when transitioning from D3hot to D0. ***Note:*** This bit must be written by BIOS to a '1' so that this register bit cannot be cleared.
2	RsvdP	0h	*Reserved.*
1:0	RW	0h	**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. 00: D0 01: D1 (not supported by IIO) 10: D2 (not supported by IIO) 11: D3_hot 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. All devices will respond to only Type 0 configuration transactions when in D3hot state (RP 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).

3.19.4 PCI Express Enhanced Configuration Space

3.19.4.1 VSECPHDR: Vendor Specific Enhanced Capability Header

This register identifies the capability structure and points to the next structure.

Register:VSECPHDR
Bus:0
Device:3
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	000Bh	**PCI Express Extended CAP_ID** Assigned for Vendor specific Capability

3.19.4.2 VSHDR: Vender Specific Header

This register identifies the capability structure and points to the next structure.

Register:VSHDR
Bus:0
Device:3
Function:0
Offset:104h

Bit	Attr	Default	Description
31:20	RO	03Ch	**VSEC Length** This field indicates the number of bytes in the entire VSEC structure, including the PCI Express Enhanced Capability header, the Vendor-Specific header, and the Vendor-Specific Registers.
19:16	RO	1h	**VSEC Version** Set to 1h for this version of the PCI Express logic
15:0	RO	0004h	**VSEC ID** Identifies Intel Vendor Specific Capability for AER on NTB

3.19.4.3 UNCERRSTS: Uncorrectable Error Status

This register identifies uncorrectable errors detected for PCI Express/DMI port.

Register:UNCERRSTS
Bus:0
Device:3
Function:0
Offset:108h

Bit	Attr	Default	Description
31:22	RsvdZ	0h	*Reserved*
21	RW1CS	0	**ACS Violation Status**
20	RW1CS	0	**Received an Unsupported Request**
19	RsvdZ	0	*Reserved*
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	RsvdZ	0h	*Reserved*
5	RW1CS	0	**Surprise Down Error Status**
4	RW1CS	0	**Data Link Protocol Error Status**
3:1	RsvdZ	0h	*Reserved*
0	RO	0	*Reserved*

3.19.4.4 UNCERRMSK: Uncorrectable Error Mask

This register masks uncorrectable errors from being signaled.

Register:UNCERRMSK
Bus:0
Device:3
Function:0
Offset:10Ch

Bit	Attr	Default	Description
31:22	RV	0h	**Reserved**
21	RWS	0	**ACS Violation Mask**
20	RWS	0	**Unsupported Request Error Mask**
19	RV	0	*Reserved*
18	RWS	0	**Malformed TLP Status**
17	RWS	0	**Receiver Buffer Overflow Mask**
16	RWS	0	**Unexpected Completion Mask**
15	RWS	0	**Completer Abort Status**
14	RWS	0	**Completion Time-out Mask**
13	RWS	0	**Flow Control Protocol Error Mask**
12	RWS	0	**Poisoned TLP Mask**
11:6	RV	0h	*Reserved*
5	RWS	0	**Surprise Down Error Mask**
4	RWS	0	**Data Link Layer Protocol Error Mask**
3:1	RV	000	*Reserved*
0	RO	0	*Reserved*

3.19.4.5 UNCERRSEV: Uncorrectable Error Severity

This register indicates the severity of the uncorrectable errors.

Register:UNCERRSEV
Bus:0
Device:3
Function:0
Offset:110h

Bit	Attr	Default	Description
31:22	RV	0h	*Reserved*
21	RWS	0	**ACS Violation Severity**
20	RWS	0	**Unsupported Request Error Severity**
19	RV	0	*Reserved*
18	RWS	1	**Malformed TLP Severity**
17	RWS	1	**Receiver Buffer Overflow Severity**
16	RWS	0	**Unexpected Completion Severity**
15	RWS	0	**Completer Abort Status**
14	RWS	0	**Completion Time-out Severity**
13	RWS	1	**Flow Control Protocol Error Severity**
12	RWS	0	**Poisoned TLP Severity**
11:6	RV	0h	*Reserved*
5	RWS	1	**Surprise Down Error Severity**
4	RWS	1	**Data Link Protocol Error Severity**
3:1	RV	000	*Reserved*
0	RO	0	*Reserved*

3.19.4.6 CORERRSTS: Correctable Error Status

This register identifies the status of the correctable errors that have been detected by the PCI Express port.

Register:CORERRSTS
Bus:0
Device:3
Function:0
Offset:114h

Bit	Attr	Default	Description
31:14	RV	0h	*Reserved*
13	RW1CS	0	**Advisory Non-fatal Error Status**
12	RW1CS	0	**Replay Timer Time-out Status**
11:9	RV	0h	*Reserved*
8	RW1CS	0	**Replay_Num Rollover Status**
7	RW1CS	0	**Bad DLLP Status**
6	RW1CS	0	**Bad TLP Status**
5:1	RV	0h	*Reserved*
0	RW1CS	0	**Receiver Error Status**

3.19.4.7 CORERRMSK: Correctable Error Mask

This register masks correctable errors from being signaled.

Register:CORERRMSK
Bus:0
Device:3
Function:0
Offset:118h

Bit	Attr	Default	Description
31:14	RV	0h	*Reserved*
13	RWS	1	**Advisory Non-fatal Error Mask**
12	RWS	0	**Replay Timer Time-out Mask**
11:9	RV	0h	*Reserved*
8	RWS	0	**Replay_Num Rollover Mask**
7	RWS	0	**Bad DLLP Mask**
6	RWS	0	**Bad TLP Mask**
5:1	RV	0h	*Reserved*
0	RWS	0	**Receiver Error Mask**

3.19.4.8 ERRCAP: Advanced Error Capabilities and Control Register

Register:ERRCAP
Bus:0
Device:3
Function:0
Offset:11Ch

Bit	Attr	Default	Description
31:9	RV	0h	*Reserved*
8	RO	0	**ECRC Check Enable**: N/A to IIO
7	RO	0	**ECRC Check Capable**: N/A to IIO
6	RO	0	**ECRC Generation Enable**: N/A to IIO
5	RO	0	**ECRC Generation Capable**: N/A to IIO
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.

3.19.4.9 HDRLOG: Header Log

This register contains the header log when the first error occurs. Headers of the subsequent errors are not logged.

Register:HDRLOG
Bus:0
Device:3
Function:0
Offset:120h

Bit	Attr	Default	Description
127:0	ROS	0h	**Header of TLP associated with error**

3.19.4.10 RPERRCMD: Root Port Error Command Register

This register controls behavior upon detection of errors.

Register:ERRCMD
Bus:0
Device:3
Function:0
Offset:130h

Bit	Attr	Default	Description
31:3	RV	0h	*Reserved*
2	RW	0	**FATAL Error Reporting Enable** Enable MSI/MSI-X interrupt on fatal errors when set. See Section 11.6, "IIO Errors Handling Summary" (IOH Platform Architecture Specification) for details of MSI/MSI-X generation for PCI Express error events.
1	RW	0	**Non-FATAL Error Reporting Enable** Enable interrupt on a non-fatal error when set. See Section 11.6, "IIO Errors Handling Summary" (IOH Platform Architecture Specification) for details of MSI/MSI-X generation for PCI Express error events.
0	RW	0	**Correctable Error Reporting Enable** Enable interrupt on correctable errors when set. See Section 11.6, "IIO Errors Handling Summary" (IOH Platform Architecture Specification) for details of MSI/MSI-X generation for PCI Express error events.

3.19.4.11 RPERRSTS: Root Port 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 IIO, and errors detected by the RP itself (which are treated conceptually as if the RP 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
Bus:0
Device:3
Function:0
Offset:134h

Bit	Attr	Default	Description
31:27	RO	0h	**Advanced Error Interrupt Message Number** Advanced Error Interrupt Message Number offset between base message data an the MSI/MSI-X message if assigned more than one message number. IIO hardware automatically updates this register to 0x1h if the number of messages allocated to the RP is 2. See bit 6:4 in Section 3.19.3.3, "MSICTRL: MSI Control Register" on page 187 for details of the number of messages allocated to a RP.
26:7	RO	0	Reserved
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.
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, i.e 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. i.e. log the first error message. 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, i.e 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. i.e. log the first error message

3.19.4.12 ERRSID: Error Source Identification Register

Register:ERRSID
Bus:0
Device:3
Function:0
Offset:138h

Bit	Attr	Default	Description
31:16	ROS	0h	**Fatal Non Fatal Error Source ID** Requestor ID of the source when an Fatal or Non Fatal error message is received and the Error Fatal/Nonfatal Received bit is not already set. i.e log ID of the first Fatal or Non Fatal error message. When the RP itself is the cause of the received message (virtual message), then a Source ID of IIOBUSNO: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. i.e log ID of the first correctable error message. When the RP itself is the cause of the received message (virtual message), then a Source ID of IIOBUSNO:DevNo:0 is logged into this register.

3.19.4.13 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
Bus:0
Device:3
Function:0
Offset:13Ch

Bit	Attr	Default	Description
31:22	RV	0h	**Reserved**
21	RWS	0	**ACS Violation Mask**
20	RWS	0	**Unsupported Request Error Mask**
19	RV	0	*Reserved*
18	RWS	0	**Malformed TLP Status**
17	RWS	0	**Receiver Buffer Overflow Mask**
16	RWS	0	**Unexpected Completion Mask**
15	RWS	0	**Completer Abort Status**
14	RWS	0	**Completion Time-out Mask**
13	RWS	0	**Flow Control Protocol Error Mask**
12	RWS	0	**Poisoned TLP Mask**
11:6	RV	0h	*Reserved*
5	RWS	0	**Surprise Down Error Mask**
4	RWS	0	**Data Link Layer Protocol Error Mask**
3:1	RV	000	*Reserved*
0	RO	0	*Reserved*

3.19.4.14 APICBASE: APIC Base Register

BDF 030 Offset 140H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.13, “APICBASE: APIC Base Register”. See Volume 2 of the Datasheet.

3.19.4.15 APICLIMIT: APIC Limit Register

BDF 030 Offset 142H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.14, “APICLIMIT: APIC Limit Register”. See Volume 2 of the Datasheet.

3.19.4.16 ACSCAPHDR: Access Control Services Extended Capability Header

BDF 030 Offset 150H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.15, “ACSCAPHDR: Access Control Services Extended Capability Header”. See Volume 2 of the Datasheet.

3.19.4.17 ACSCAP: Access Control Services Capability Register

This register identifies the Access Control Services (ACS) capabilities.

Register:ACSCAP
Bus:0
Device:3
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	RO	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	0	ACS Upstream Forwarding (U) Indicates that the component implements ACS Upstream Forwarding.
3	RO	0	ACS P2P Completion Redirect (C) Indicates that the component implements ACS P2P Completion Redirect.
2	RO	0	ACS P2P Request Redirect (R) Indicates that the component implements ACS P2P Request Redirect.
1	RO	0	ACS Translation Blocking (B) Indicates that the component implements ACS Translation Blocking.
0	RO	0	ACS Source Validation (V) Indicates that the component implements ACS Source Validation.

3.19.4.18 ACSCTRL: Access Control Services Control Register

This register identifies the Access Control Services (ACS) control bits.

Register:ACSCTRL
Bus:0
Device:3
Function:0
Offset:156h

Bit	Attr	Default	Description
15:7	RO	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	RO	0	ACS P2P Egress Control Enable (E) This is hardwired to 0b as the component does not implement ACS P2P Egress Control.
4	RWL	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. The U bit only applies to upstream TLPs arriving at a Downstream Port, and whose normal routing targets the same Downstream Port. ***Note:*** When in NTB mode, this register bit is locked RO =0. See bits 1:0 of "PPD: PCIE Port Definition" on page 203
3	RWL	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. ***Note:*** When in NTB mode, this register bit is locked RO =0. See bits 1:0 of "PPD: PCIE Port Definition" on page 203
2	RWL	0	ACS P2P Request Redirect Enable (R) This bit determines when the component redirects peer-to-peer Requests upstream. ***Note:*** When in NTB mode, this register bit is locked RO =0. See bits 1:0 of "PPD: PCIE Port Definition" on page 203
1	RWL	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. ***Note:*** When in NTB mode, this register bit is locked RO =0. See bits 1:0 of "PPD: PCIE Port Definition" on page 203
0	RWL	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. ***Note:*** When in NTB mode, this register bit is locked RO =0. See bits 1:0 of "PPD: PCIE Port Definition" on page 203

3.19.4.19 PERFCTRLSTS: Performance Control and Status Register

BDF 030 Offset 180H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.18, "PERFCTRLSTS: Performance Control and Status Register". See Volume 2 of the Datasheet.

3.19.4.20 MISCCTRLSTS: Misc. Control and Status Register

BDF 030 Offset 188H. This register exist in both RP and NTB modes. It is documented in RP Section 22.5.6.24, "MISCCTRLSTS: Misc. Control and Status Register (Dev#0, PCIe Mode and Dev#3-6)" in Volume 2 of the Datasheet.

3.19.4.21 PCIE_IOU0_BIF_CTRL: PCIE IOU0 Bifurcation Control Register

BDF 030 Offset 190H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.21, "PCIE_IOU0_BIF_CTRL: PCIE IOU0 Bifurcation Control Register" in Volume 2 of the Datasheet.

3.19.4.22 NTBDEVCAP: PCI Express Device Capabilities Register

The PCI Express Device Capabilities register identifies device specific information for the device.

Register:NTBDEVCAP
Bus:0
Device:3
Function:0
Offset:194h

Bit	Attr	Default	Description
31:29	RsvdP	0h	*Reserved*
28	RO	0b	**Function Level Reset Capability** A value of 1b indicates the Function supports the optional Function Level Reset mechanism. NTB does not support this functionality.
27:26	RO	0h	**Captured Slot Power Limit Scale** Does not apply to RPs or integrated devices This value is hardwired to 00h NTB is required to be able to receive the Set_Slot_Power_Limit message without error but simply discard the Message value. ***Note:*** *PCI Express Base Specification,* Revision 2.0 states Components with Endpoint, Switch, or PCI Express-PCI Bridge Functions that are targeted for integration on an adapter where total consumed power is below the lowest limit defined for the targeted form factor are permitted to ignore Set_Slot_Power_Limit Messages, and to return a value of 0 in the Captured Slot Power Limit Value and Scale fields of the Device Capabilities register
25:18	RO	00h	**Captured Slot Power Limit Value** Does not apply to RPs or integrated devices This value is hardwired to 00h NTB is required to be able to receive the Set_Slot_Power_Limit message without error but simply discard the Message value. ***Note:*** *PCI Express Base Specification,* Revision 2.0 states components with Endpoint, Switch, or PCI Express-PCI Bridge Functions that are targeted for integration on an adapter where total consumed power is below the lowest limit defined for the targeted form factor are permitted to ignore Set_Slot_Power_Limit Messages, and to return a value of 0 in the Captured Slot Power Limit Value and Scale fields of the Device Capabilities register
17:16	RsvdP	0h	*Reserved*
15	RO	1	**Role Based Error Reporting**: IIO is 1.1 compliant and so supports this feature
14	RO	0	**Power Indicator Present on Device** Does not apply to RPs or integrated devices
13	RO	0	**Attention Indicator Present** Does not apply to RPs or integrated devices
12	RO	0	**Attention Button Present** Does not apply to RPs or integrated devices

Register:NTBDEVCAP
Bus:0
Device:3
Function:0
Offset:194h

Bit	Attr	Default	Description
11:9	RWO	110b	**Endpoint L1 Acceptable Latency** This field indicates the acceptable latency that an Endpoint can withstand due to the transition from L1 state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L1 Acceptable Latency number to compare against the L1 Exit Latencies reported (see below) by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L1 entry can be used with no loss of performance. Defined encodings are: 000b Maximum of 1 us 001b Maximum of 2 us 010b Maximum of 4 us 011b Maximum of 8 us 100b Maximum of 16 us 101b Maximum of 32 us 110b Maximum of 64 us 111b No limit BIOS must program this value
8:6	RWO	000b	**Endpoint L0s Acceptable Latency** This field indicates the acceptable total latency that an Endpoint can withstand due to the transition from L0s state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L0s Acceptable Latency number to compare against the L0s exit latencies reported by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L0s entry can be used with no loss of performance. Defined encodings are: 000b Maximum of 64 ns 001b Maximum of 128 ns 010b Maximum of 256 ns 011b Maximum of 512 ns 100b Maximum of 1 us 101b Maximum of 2 us 110b Maximum of 4 us 111b No limit BIOS must program this value
5	RO	1	**Extended Tag Field Supported** IIO devices support 8-bit tag 1 = Maximum Tag field is 8 bits 0 = Maximum Tag field is 5 bits
4:3	RO	00b	**Phantom Functions Supported** IIO does not support phantom functions. 00b = No Function Number bits are used for Phantom Functions
2:0	RO	001b	**Max Payload Size Supported** IIO supports 256B payloads on PCI Express ports 001b = 256 bytes max payload size

3.19.4.23 LNKCAP: PCI Express Link Capabilities Register

The Link Capabilities register identifies the PCI Express specific link capabilities

The link capabilities register needs some default values setup by the local host. This register has been created to provide a back door path to program the link capabilities from the primary side. The link capabilities register on the secondary side of the NTB is located at Section 3.20.3.20, “LNKCAP: PCI Express Link Capabilities Register” .

Register:LNKCAP
Bus:0
Device:3
Function:0
Offset:19Ch

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	RsvdP	0h	*Reserved.*
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**: IIO 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:** IIO supports reporting a surprise down error condition
18	RO	0	**Clock Power Management:** Does not apply to IIO.
17:15	RWO	010	**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 us 001: 1 us to less than 2 us 010: 2 us to less than 4 us 011: 4 us to less than 8 us 100: 8 us to less than 16 us 101: 16 us to less than 32 us 110: 32 us to 64 us 111: More than 64us

Register:LNKCAP
Bus:0
Device:3
Function:0
Offset:19Ch

Bit	Attr	Default	Description
14:12	RWO	011	**L0s Exit Latency** This field indicates the L0s exit latency (i.e L0s to L0) for the PCI-Express port. 000: Less than 64 ns 001: 64 ns to less than 128 ns 010: 128 ns to less than 256 ns 011: 256 ns to less than 512 ns 100: 512 ns to less than 1 is 101: 1 is to less than 2 is 110: 2 is to 4 is 111: More than 4 is
11:10	RWO	11	**Active State Link PM Support** This field indicates the level of active state power management supported on the given PCI-Express port. 00: Disabled 01: L0s Entry Supported 10: Reserved 11: L0s and L1 Supported
9:4	RWO	001000b	**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 - *Reserved*
3:0	RO	See description	**Link Speeds Supported** IIO supports both 2.5 Gbps and 5 Gbps speeds if Gen2_OFF fuse is OFF else it supports only Gen1 This field defaults to 0001b if Gen2_OFF fuse is ON. And when Gen2_OFF fuse is OFF this field defaults to 0010b.

1. There are restrictions with routing x2 lanes from IIO to a slot. See Section 3.3, "PCI Express Link Characteristics - Link Training, Bifurcation, Downgrading and Lane Reversal Support" (IOH Platform Architecture Specification) for details.

3.19.4.24 LNKCON: PCI Express Link Control Register

The PCI Express Link Control register controls the PCI Express Link specific parameters. The link control register needs some default values setup by the local host. This register has been created to provide a back door path to program the link control register from the primary side. The link control register on the secondary side of the NTB is located at Section 3.20.3.21, "LNKCON: PCI Express Link Control Register"

In NTB/RP mode RP will program this register. In NTB/NTB mode local host BIOS will program this register.

Register:LNKCON
Bus:0
Device:3
Function:0
Offset:1A0h

Bit	Attr	Default	Description
15:12	RsvdP	0h	**Reserved**
11	RW	0b	**Link Autonomous Bandwidth Interrupt Enable** - This bit is not applicable and is reserved for Endpoints
10	RW	0b	**Link Bandwidth Management Interrupt Enable** - This bit is not applicable and is reserved for Endpoints
09	RW	0b	**Hardware Autonomous Width Disable** - IIO never changes a configured link width for reasons other than reliability.
08	RO	0b	**Enable Clock Power Management** N/A to IIO
07	RW	0b	**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 2.0 for details.
06	RW	0b	**Common Clock Configuration** IIO does nothing with this bit
05	WO	0b	**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. ***Note:*** Hardware clears this bit on next clock after it is written.

Register:LNKCON
Bus:0
Device:3
Function:0
Offset:1A0h

Bit	Attr	Default	Description
04	RWL	0b	**Link Disable** This bit is not applicable and is reserved for Endpoints ***Note:*** Appears to SW as RO
03	RO	0b	**Read Completion Boundary** Set to zero to indicate IIO could return read completions at 64B boundaries ***Note:*** NTB is not PCIE compliant in this respect. NTB is only capable of 64B RCB. If connecting to non IA IP and the IP does the optional 128B RCB check on received packets, packets will be seen as malformed. This is not an issue with any Intel IP.
02	RsvdP	0b	**Reserved.**
01:00	RW	00b	**Active State Link PM Control**: When 01b or 11b, L0s on transmitter is enabled, otherwise it is disabled. Defined encodings are: 00b Disabled 01b L0s Entry Enabled 10b L1 Entry Enabled 11b L0s and L1 Entry Enabled ***Note:*** "L0s Entry Enabled" indicates the Transmitter entering L0s is supported. The Receiver must be capable of entering L0s even when the field is disabled (00b). ASPM L1 must be enabled by software in the Upstream component on a Link prior to enabling ASPM L1 in the Downstream component on that Link. When disabling ASPM L1, software must disable ASPM L1 in the Downstream component on a Link prior to disabling ASPM L1 in the Upstream component on that Link. ASPM L1 must only be enabled on the Downstream component if both components on a Link support ASPM L1.

3.19.4.25 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 etc. The link status register needs some default values setup by the local host. This register has been created to provide a back door path to program the link status from the primary side. The link status register on the secondary side of the NTB is located at Section 3.20.3.22, "LNKSTS: PCI Express Link Status Register" .

Register:LNKSTS
Bus:0
Device:3
Function:0
Offset:1A2h

Bit	Attr	Default	Description
15	RW1C	0	**Link Autonomous Bandwidth Status**: This bit is not applicable and is reserved for Endpoints
14	RW1C	0	**Link Bandwidth Management Status**: This bit is not applicable and is reserved for Endpoints
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	RWO	1	**Slot Clock Configuration** This bit indicates whether IIO 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 ***Note:***

Register:LNKSTS
Bus:0
Device:3
Function:0
Offset:1A2h

Bit	Attr	Default	Description
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 IIO hardware clears this bit once LTSSM has exited the recovery/ configuration state. See the *PCI Express Base Specification,* Revision 2.0 for details of which states within the LTSSM would set this bit and which states would clear this bit.
10	RO	0	*Reserved*
9:4	RO	0h	**Negotiated Link Width** This field indicates the negotiated width of the given PCI Express link after training is completed. Defined encodings are: 00 0001b: x1 00 0010b: x2 00 0100b: x4 00 1000b: x8 01 0000b: x16 All other encodings are reserved 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	1h	**Current Link Speed** This field indicates the negotiated Link speed of the given PCI Express Link. 0001- 2.5 Gbps 0010 - 5Gbps (IIO will never set this value when Gen2_OFF fuse is blown) Others - *Reserved* 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.

3.19.4.26 SLTCAP: PCI Express Slot Capabilities Register

The Slot Capabilities register identifies the PCI Express specific slot capabilities.

Register:SLTCAP
Bus:0
Device:3
Function:0
Offset:1A4h

Bit	Attr	Default	Description
31:19	RWO	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**: IIO is capable of command complete interrupt.
17	RWO	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: EMIL has been defeatured per DCN 430354. BIOS must write a 0 to this bit to lockout EMIL.
16:15	RWO	0h	**Slot Power Limit Scale** This field specifies the scale used for the Slot Power Limit Value and is initialized by BIOS. IIO 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	RWO	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. IIO uses this field when it sends a Set_Slot_Power_Limit message on PCI Express.
6	RWO	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.
5	RWO	0h	**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 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. This bit is used by IIO 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 hot pluggable slot and the hotplug surprise bit is set, then any transition to DL_Inactive is not considered an error. See the *PCI Express Base Specification,* Revision 2.0 for further details.

Register:SLTCAP
Bus:0
Device:3
Function:0
Offset:1A4h

Bit	Attr	Default	Description
4	RWO	0h	**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 BIOS programs this field with a 1 for CEM/SIOM FFs and a 0 for Express cable.
3	RWO	0h	**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 BIOS programs this field with a 1 for CEM/SIOM FFs.
2	RWO	0h	**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 BIOS programs this field with a 0 for SIOM/Express cable and with either 0 or 1 for CEM depending on system design.
1	RWO	0h	**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 BIOS programs this field with a 1 for CEM/SIOM FFs and a 0 for Express cable.
0	RWO	0h	**Attention Button Present** This bit indicates that the Attention Button event signal is routed (from slot or on-board in the chassis) to the IIOs hotplug controller. 0: indicates that an Attention Button signal is routed to IIO 1: indicates that an Attention Button is not routed to IIO BIOS programs this field with a 1 for CEM/SIOM FFs.

3.19.4.27 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
Bus:0
Device:3
Function:0
Offset:1A8h

Bit	Attr	Default	Description
15:13	RsvdP	0h	*Reserved.*
12	RWS	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	WO	0	**Electromechanical Interlock Control** When software writes either a 1 to this bit, IIO pulses the EMIL pin per *PCI Express Server/Workstation Module Electromechanical Spec*, Revision 1.0. 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	RWS	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	RW	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: *Reserved.* 01: On 10: Blink (IIO 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 IIO over a dedicated SMBus port. IIO does not generated the Power_Indicator_On/Off/Blink messages on PCI Express when this field is written to by software.
7:6	RW	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: *Reserved.* 01: On 10: Blink (The IIO drives 1.5 Hz square wave) 11: Off When this register is written, the event is signaled via the virtual pins[2] of the IIO over a dedicated SMBus port. IIO does not generated the Attention_Indicator_On/Off/Blink messages on PCI Express when this field is written to by software.

Register:SLTCON
Bus:0
Device:3
Function:0
Offset:1A8h

Bit	Attr	Default	Description
5	RW	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	RW	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
3	RW	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	RW	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.
1	RW	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.
0	RW	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.

1. More information on Virtual pins can be found in Section 11.7.2.1, “PCI Express Hot Plug Interface” (IOH Platform Architecture Specification).
2. More information on Virtual pins can be found in Section 11.7.2.1, “PCI Express Hot Plug Interface” (IOH Platform Architecture Specification).

3.19.4.28 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.

Register:SLTSTS
Bus:0
Device:3
Function:0
Offset:1AAh

Bit	Attr	Default	Description
15:9	RsvdZ	0h	*Reserved.*
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.
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. See the *PCI Express Base Specification,* Revision 2.0 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, IIO hardwires this bit to 1b. ***Note:*** OS could get confused when it sees an empty PCI Express RP i.e. “no slots + no presence”, since this is now disallowed in the spec. So BIOS must hide all unused RPs devices in IIO config space, via the DEVHIDE register in Intel® QPI Configuration Register 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 IIO 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.

Register:SLTSTS
Bus:0
Device:3
Function:0
Offset:1AAh

Bit	Attr	Default	Description
3	RW1C	0h	**Presence Detect Changed** This bit is set by the IIO 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 IIO 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 IIO 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 IIO 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. IIO silently discards the Attention_Button_Pressed message if received from PCI Express link without updating this bit.

3.19.4.29 ROOTCON: PCI Express Root Control Register

The PCI Express Root Control register specifies parameters specific to the root complex port.

Note: Since this PCI Express port can be configured as RP or NTB when configured as NTB register is moved from standard location and used to unable error reporting for upstream notification to the local host that is physically attached to the NTB.

Register:ROOTCON
Bus:0
Device:3
Function:0
Offset:1ACh

Bit	Attr	Default	Description
15:5	**Rsvd P**	0h	*Reserved.*
4	**RWL**	0b	CRS software visibility Enable ***Note:*** This bit appears as RO to SW
3	**RWL**	0b	**PME Interrupt Enable** There are no PME events for NTB ***Note:*** This bit appears as RO to SW
2	**RW**	0b	**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. The internal core error logic of IIO then decides if/how to escalate the error further (pins/ message etc.). See Section 11.5, "PCI Express* RAS" (IIO Platform Architecture Specification) for details of how/which system notification is generated for a PCI Express/DMI fatal error. 1 = Indicates that a internal core error logic notification should be generated if a fatal error (ERR_FATAL) is reported by this port. 0 = No internal core error logic notification should be generated on a fatal error (ERR_FATAL) reported by this port. Generation of system notification on a PCI Express/DMI 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. See the *PCI Express Base Specification*, Revision 1.1 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/DMI port.
1	**RW**	0b	**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. The internal core error logic of IIO then decides if/how to escalate the error further (pins/ message etc.). See Section 11.1, "IIO RAS Overview" (IIO Platform Architecture Specification) for details of how/which system notification is generated for a PCI Express/DMI 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 this port. 0 = No internal core error logic notification should be generated on a non-fatal error (ERR_NONFATAL) reported by this port. Generation of system notification on a PCI Express/DMI 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. See the *PCI Express Base Specification*, Revision 1.1 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/DMI port.

Register:ROOTCON
Bus:0
Device:3
Function:0
Offset:1ACh

Bit	Attr	Default	Description
0	**RW**	0b	**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. The internal core error logic of IIO then decides if/how to escalate the error further (pins/message etc.). See Section 11.1, "IIO RAS Overview" (IIO Platform Architecture Specification) 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 this port. 0 = No internal core error logic notification should be generated on a correctable error (ERR_COR) reported by this port. 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. See the *PCI Express Base Specification*, Revision 1.1 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/DMI port.

3.19.4.30 DEVCAP2: PCI Express Device Capabilities 2 Register

NTB Primary is a RCiEP but needs to have some capabilities associated with a RP so that transactions are guaranteed to complete. This register controls transactions sent from local CPU to an external device through the PCIE NTB port.

Register:DEVCAP2
Bus:0
Device:3
Function:0
Offset:1B4h

Bit	Attr	Default	Description
31:6	RO	0h	**Reserved**
5	RO	1	Alternative RID Interpretation (ARI) Capable - This bit is set to 1b indicating RP supports this capability.
4	RO	1	Completion Time-out Disable Supported - IIO supports disabling completion time-out
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. Four time values ranges are defined: Range A: 50us to 10ms Range B: 10ms to 250ms Range C: 250ms to 4s Range D: 4s to 64s Bits ares set according to table below to show time-out value ranges supported. 0000b: Completions Time-out 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: Ranges B, C & D 1111b: Range A, B, C & D All other values are reserved. IIO supports time-out values up to 10ms-64s. *PCI Express Base Specification,* Revision 2.0 states This field is applicable only to RPs, Endpoints that issue Requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of Requests issued on PCI Express. For all other Functions this field is reserved and must be hardwired to 0000b.

3.19.4.31 DEVCTRL2: PCI Express Device Control 2 Register

Register:DEVCTRL2
Bus:0
Device:3
Function:0
Offset:1B8h

Bit	Attr	Default	Description
15:6	RO	0h	**Reserved**
5	RW	0	**Alternative RID Interpretation (ARI) Enable** - When set to 1b, ARI is enabled for the NTB EP. ***Note:*** The BIOS must leave this bit at its default value.
4	RW	0	**Completion Time-out Disable** – When set to 1b, this bit disables the Completion Time-out mechanism for all NP tx that IIO issues on the PCIE/DMI link and in the case of Intel® QuickData Technology DMA, for all NP tx that DMA issues upstream. When 0b, completion time-out is enabled. Software can change this field while there is active traffic in the RP.
3:0	RW	0000b	**Completion Time-out Value on NP Tx that IIO issues on PCIE/DMI** – 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 = 10ms to 50ms 0001b = Reserved (IIO aliases to 0000b) 0010b = Reserved (IIO aliases to 0000b) 0101b = 16ms to 55ms 0110b = 65ms to 210ms 1001b = 260ms to 900ms 1010b = 1s to 3.5s 1101b = 4s to 13s 1110b = 17s to 64s When OS selects 17s to 64s range, BDF 030 Offset 232H. This register exists in both RP and NTB modes. It is documented in RP Section 3.4.5.34, “XPGLBERRPTR - XP Global Error Pointer Register” in Volume 2 of the Datsheet. It 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 IIO hardware. Software can change this field while there is active traffic in the RP.

3.19.4.32 LNKCON2: PCI Express Link Control Register 2

Register:LNKCON2
Bus:0
Device:3
Function: 0
Offset:1C0h

Bit	Attr	Default	Description
15:13	RO	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 sub state.**
9:7	RWS	0	**Transmit Margin - This field controls the value of the non de-emphasized voltage level at the Transmitter pins.**
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.** ***Note:*** This register is not PCIE compliant. It is reserved for endpoints but design accommodates this capability.
5	RW	0	**Hardware Autonomous Speed Disable**: IIO 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 2.5Gb/s Target Link Speed 0010b 5Gb/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, IIO 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. For PCI Express ports (Dev#1-10), this field defaults to 0001b if Gen2_OFF fuse is ON. And when Gen2_OFF fuse is OFF this field defaults to 0010b. For Device 0 this field defaults to 0001b.

3.19.4.33 LNKSTS2: PCI Express Link Status 2 Register

The PCI Express Link Status 2 register provides information on the status of the PCI Express Link current De-emphasis level and other definition is currently reserved.

Register:LNKSTS2
Bus:0
Device:3
Function:0
Offset:1C2h

Bit	Attr	Default	Description
15:01	RO	0h	**Reserved**
00	RO	0b	**Current De-emphasis Level:** When the Link is operating at 5 GT/s speed, this bit reflects the level of de-emphasis. Encodings: 1b -3.5 dB 0b -6 dB The value in this bit is undefined when the Link is operating at 2.5 GT/s speed.

3.19.4.34 CTOCTRL: Completion Time-out Control Register

BDF 030 Offset 1E0H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.35, “CTOCTRL: Completion Timeout Control Register”. See Volume 2 of the Datasheet.

3.19.4.35 PCIE_LER_SS_CTRLSTS: PCI Express Live Error Recovery/Stop and Scream Control and Status Register

BDF 030 Offset 1E4H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.35, “PCIE_LER_SS_CTRLSTS: PCI Express Live Error Recovery/Stop and Scream Control and Status Register”. See Volume 2 of the Datasheet.

3.19.4.36 XPCORERRSTS - XP Correctable Error Status Register

BDF 030 Offset 200H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.33, “XPCORERRSTS - XP Correctable Error Status Register” . See Volume 2 of the Datasheet.

3.19.4.37 XPCORERRMSK - XP Correctable Error Mask Register

BDF 030 Offset 204H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.32, “XPCORERRMSK - XP Correctable Error Mask Register”. See Volume 2 of the Datasheet.

3.19.4.38 XPUNCERRSTS - XP Uncorrectable Error Status Register

BDF 030 Offset 208H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.24, “XPUNCERRSTS - XP Uncorrectable Error Status Register”. See Volume 2 of the Datasheet.

3.19.4.39 XPUNCERRMSK - XP Uncorrectable Error Mask Register

BDF 030 Offset 20CH. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.25, “XPUNCERRMSK - XP Uncorrectable Error Mask Register” . See Volume 2 of the Datasheet.

3.19.4.40 XPUNCERRSEV - XP Uncorrectable Error Severity Register

BDF 030 Offset 210H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.26, “XPUNCERRSEV - XP Uncorrectable Error Severity Register”. See Volume 2 of the Datasheet.

3.19.4.41 XPUNCERRPTR - XP Uncorrectable Error Pointer Register

BDF 030 Offset 214H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.27, “XPUNCERRPTR - XP Uncorrectable Error Pointer Register”. See Volume 2 of the Datasheet.

3.19.4.42 UNCEDMASK: Uncorrectable Error Detect Status Mask

BDF 030 Offset 218H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.28, “UNCEDMASK: Uncorrectable Error Detect Status Mask”. See Volume 2 of the Datasheet.

3.19.4.43 COREDMASK: Correctable Error Detect Status Mask

BDF 030 Offset 21CH. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.29, “COREDMASK: Correctable Error Detect Status Mask”. See Volume 2 of the Datasheet.

3.19.4.44 RPEDMASK - Root Port Error Detect Status Mask

BDF 030 Offset 220H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.30, “RPEDMASK - Root Port Error Detect Status Mask”. See Volume 2 of the Datasheet.

3.19.4.45 XPUNCEDMASK - XP Uncorrectable Error Detect Mask Register

BDF 030 Offset 224H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.31, “XPUNCEDMASK - XP Uncorrectable Error Detect Mask Register”. See Volume 2 of the Datasheet.

3.19.4.46 XPCOREDMASK - XP Correctable Error Detect Mask Register

BDF 030 Offset 228H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.32, “XPCOREDMASK - XP Correctable Error Detect Mask Register”. See Volume 2 of the Datasheet.

3.19.4.47 XPGLBERRSTS - XP Global Error Status Register

BDF 030 Offset 230H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.33, “XPGLBERRSTS - XP Global Error Status Register”. See Volume 2 of the Datasheet.

3.19.4.48 XPGLBERRPTR - XP Global Error Pointer Register

BDF 030 Offset 232H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.34, “XPGLBERRPTR - XP Global Error Pointer Register”. See Volume 2 of the Datasheet.

3.20 PCI Express Configuration Registers (NTB Secondary Side)

3.20.1 Configuration Register Map (NTB Secondary Side)

This section covers the NTB secondary side configuration space registers.

When configured as an NTB there are two sides to discuss for configuration registers. The primary side of the NTB's configuration space is located on Bus 0, Device 3, Function 0 with respect to the Intel® Xeon® processor C5500/C3500 series and a secondary side of the NTB's configuration space is located on some enumerated bus on another system and does not exist as configuration space on the local Intel® Xeon® processor C5500/C3500 series system anywhere.

The primary side registers are discussed in Section 3.19, "PCI Express Configuration Registers (NTB Primary Side)"

This section discusses the secondary side registers.

Figure 62. PCI Express NTB Secondary Side Type0 Configuration Space

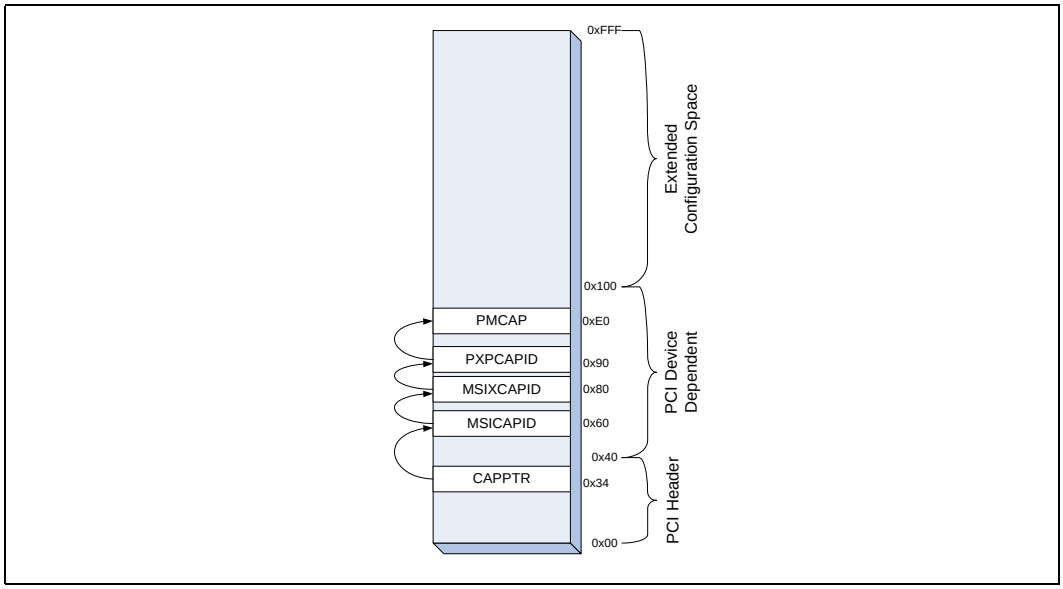
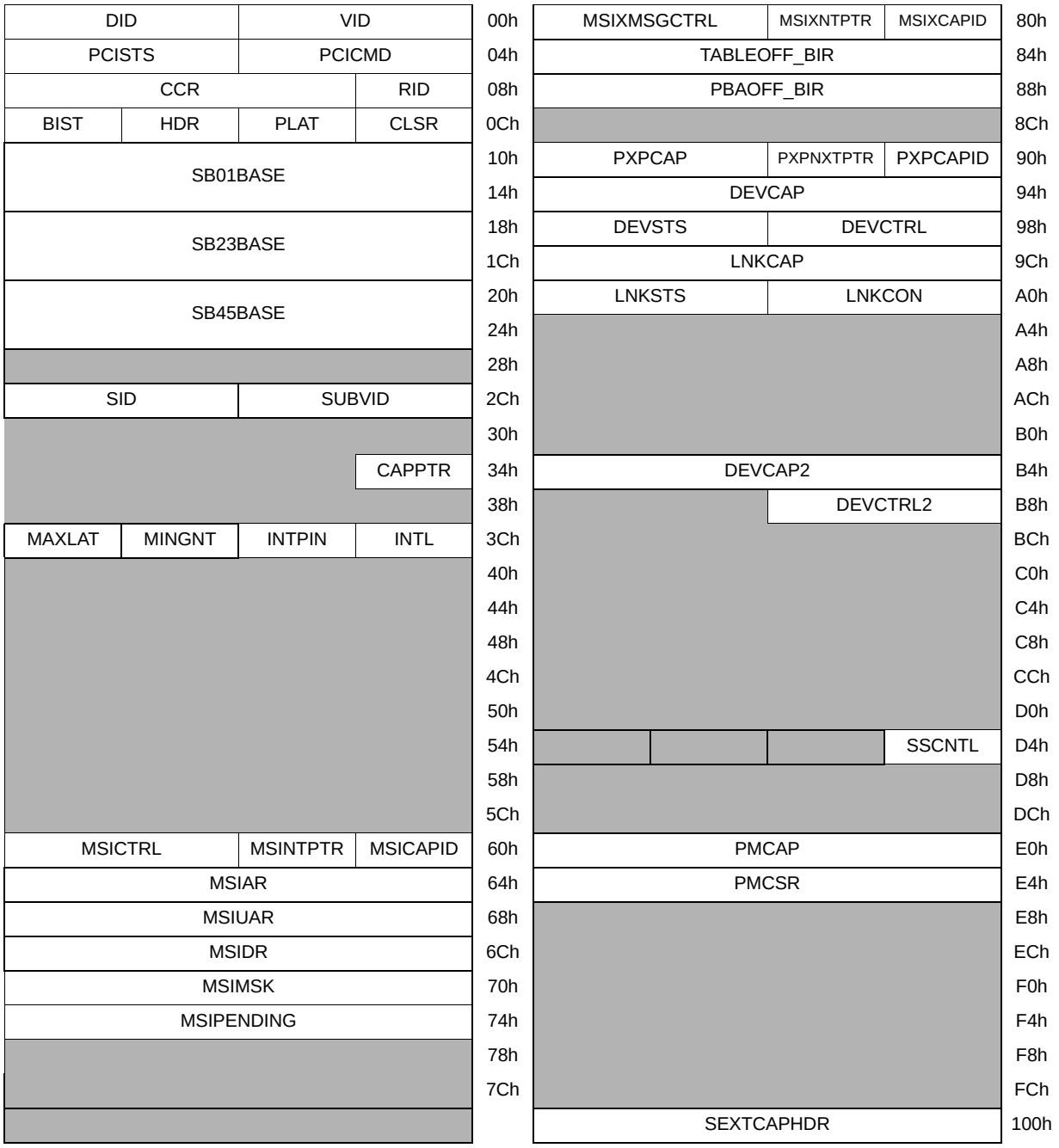


Figure 62 illustrates how each PCI Express port configuration space appears to software. Each PCI Express configuration space has three regions:

- **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 IIO, 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 IIO supports the Advanced Error Reporting Capability in this configuration space.


DID
VID
00h
PCISTS
PCICMD
04h
CCR
RID
08h
BIST
HDR
PLAT
CLSR
0Ch
SB01BASE
10h
14h
SB23BASE
18h
1Ch
SB45BASE
20h
24h
28h
SID
SUBVID
2Ch
30h
CAPPTR
34h
38h
MAXLAT
MINGNT
INTPIN
INTL
3Ch
40h
44h
48h
4Ch
50h
54h
58h
5Ch
MSICTRL
MSINTPTR
MSICAPID
60h
MSIAR
64h
MSIUAR
68h
MSIDR
6Ch
MSIMSK
70h
MSIPENDING
74h
78h
7Ch
MSIXMSGCTRL
MSIXNTPTR
MSIXCAPID
80h
TABLEOFF_BIR
84h
PBAOFF_BIR
88h
8Ch
PXPCAP
PXPNXTPTR
PXPCAPID
90h
DEVCAP
94h
DEVSTS
DEVCTRL
98h
LNKCAP
9Ch
LNKSTS
LNKCON
A0h
A4h
A8h
ACh
B0h
DEVCAP2
B4h
DEVCTRL2
B8h
BCh
C0h
C4h
C8h
CCh
D0h
SSCNTL
D4h
D8h
DCh
PMCAP
E0h
PMCSR
E4h
E8h
ECh
F0h
F4h
F8h
FCh
SEXTCAPHDR
100h

3.20.2 Standard PCI Configuration Space (0x0 to 0x3F) - Type 0 Common Configuration Space

This section covers the secondary side registers in the 0x0 to 0x3F region that are common to Bus M, Device 0. The Primary side of the NTB was discussed in the previous section and is located on NTB Bus 0, Device 3. 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.

Note: Several registers will be duplicated in the three sections discussing the three modes it operates in RP, NTB/NTB, and NTB/RP primary and secondary but are repeated here for readability.

There are three access mechanisms to get to the secondary side configuration register.

- Conventional PCI BDF from the secondary side.
- MMIO from the primary side. This is needed in order to program the secondary side configuration registers in the case of NTB/NTB. The registers are reached through the primary side BAR01 memory window at an offset starting at 500h.
- MMIO from the secondary side. This is a secondary method to reach the same registers and the conventional BDf mechanism. The registers are reached through the secondary side BAR01 memory window at an offset starting at 500h.

3.20.2.1 VID: Vendor Identification Register

Register:VID
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:00h

Bit	Attr	Default	Description
15:0	**RO**	8086h	Vendor Identification Number The value is assigned by PCI-SIG to Intel.

3.20.2.2 DID: Device Identification Register (Dev#N, PCIE NTB Sec Mode)

Register:DID
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:02h

Bit	Attr	Default	Description
15:0	RO	3727h	**Device Identification Number** The value is assigned by Intel to each product. IIO will have a unique device id for each of its PCI Express single function devices.

3.20.2.3 PCICMD: PCI Command Register (Dev#N, PCIE NTB Sec Mode)

This register defines the PCI 3.0 compatible command register values applicable to PCI Express space.

Register:PCICMD
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:04h

Bit	Attr	Default	Description
15:11	RV	00h	*Reserved.* (by PCI SIG)
10	RW	0	**INTxDisable: Interrupt Disable** Controls the ability of the PCI-Express port to generate INTx messages. This bit does not affect the ability of Intel® Xeon® processor C5500/C3500 series to route interrupt messages received at the PCI-Express port. However, this bit controls the generation of legacy interrupts to the DMI for PCI-Express errors detected internally in this port (e.g. Malformed TLP, CRC error, completion time out etc.) or when receiving RP error messages or interrupts due to HP/PM events generated in legacy mode within Intel® Xeon® processor C5500/C3500 series. See the INTPIN register in Section 3.20.2.18, "INTPIN: Interrupt Pin Register" on page 251 for interrupt routing to DMI. 1: Legacy Interrupt mode is disabled 0: Legacy Interrupt mode is enabled
9	RO	0	Fast Back-to-Back Enable Not applicable to PCI Express must be hardwired to 0.
8	RO	0	**SERR Enable** For PCI Express/DMI 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 IIO 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 IIO 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 See the *PCI Express Base Specification,* Revision 2.0 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.
7	RO	0	**IDSEL Stepping / Wait Cycle Control** Not applicable to PCI Express must be hardwired to 0.
6	RW	0	**Parity Error Response** For PCI Express/DMI ports, IIO ignores this bit and always does ECC/ parity checking and signaling for data/address of transactions both to and from IIO. This bit though affects the setting of bit 8 in the PCISTS (see bit 8 in Section 3.19.2.4) register.
5	RO	0	**VGA palette snoop Enable** Not applicable to PCI Express must be hardwired to 0.
4	RO	0	**Memory Write and Invalidate Enable** Not applicable to PCI Express must be hardwired to 0.

Register:PCICMD
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:04h

Bit	Attr	Default	Description
3	RO	0	**Special Cycle Enable** Not applicable to PCI Express must be hardwired to 0.
2	RW	0	**Bus Master Enable** 1: When this bit is Set, the PCIE NTB will forward Memory Requests that it receives on its primary internal interface to its secondary external link interface. 0: When this bit is Clear, the PCIE NTB will not forward Memory Requests that it receives on its primary internal interface. Memory requests received on the primary internal interface will be returned to requester as an Unsupported Requests UR. Requests other than Memory Requests are not controlled by this bit. Default value of this bit is 0b.
1	RW	0	**Memory Space Enable** 1: Enables a PCI Express port's memory range registers to be decoded as valid target addresses for transactions from secondary side. 0: Disables a PCI Express port's memory range registers (including the Configuration Registers range registers) to be decoded as valid target addresses for transactions from secondary side.
0	RO	0	**IO Space Enable** Controls a device's response to I/O Space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to I/O Space accesses. State after RST# is 0. NTB does not support I/O space accesses. Hardwired to 0

3.20.2.4 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 IIO bus.

Register:PCISTS
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
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 (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.
14	RO	0	**Signaled System Error** 1: The device reported fatal/non-fatal (and not correctable) errors it detected on its PCI Express interface through the ERR[2:0] pins or message to PCH, with SERRE bit enabled. 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 PCH via a message. IIO 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 (IIO internal bus). 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: • Device receives a completion on the primary interface (internal bus of IIO) 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® QPI source address decoder. • Other master abort conditions detected on the IIO internal bus amongst those listed in the Section 6.4.2, "Inbound Address Decoding" (IOH Platform Architecture Specification).
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 (IIO internal bus). 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: • Device receives a completion on the primary interface (internal bus of IIO) 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 the Intel® QPI that returns a failed completion status • Other completer abort conditions detected on the IIO internal bus amongst those listed in Section 6.4.2, "Inbound Address Decoding" (IOH Platform Architecture Specification) .

Register:PCISTS
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:06h

Bit	Attr	Default	Description
11	RW1C	0	**Signaled Target Abort** This bit is set when the NTB port forwards a completer abort (CA) completion status from the primary interface to the secondary interface.
10:9	RO	0h	**DEVSEL# Timing** Not applicable to PCI Express. Hardwired to 0.
8	RW1C	0	**Master Data Parity Error** This bit is set if the Parity Error Response bit in the PCI Command register is set and the • Requestor receives a poisoned completion on the secondary interface or • Requestor forwards a poisoned write request (including MSI/MSI-X writes) from the primary interface to the secondary interface.
7	RO	0	**Fast Back-to-Back** Not applicable to PCI Express. Hardwired to 0.
6	RO	0	*Reserved*
5	RO	0	**66MHz capable** Not applicable to PCI Express. Hardwired to 0.
4	RO	1	**Capabilities List** This bit indicates the presence of a capabilities list structure
3	RO	0	**INTx Status** When Set, indicates that an INTx emulation interrupt is pending internally in the Function.
2:0	RV	0h	*Reserved*

3.20.2.5 RID: Revision Identification Register

This register contains the revision number of the IIO. The revision number steps the same across all devices and functions i.e. individual devices do not step their RID independently.

IIO supports the CRID feature where in this register's value can be changed by BIOS. See Section 3.2.2, "Compatibility Revision ID" in Volume 2 of the Datasheet for details.

Register:RID
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
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

3.20.2.6 CCR: Class Code Register

This register contains the Class Code for the device.

Register:CCR
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:09h

Bit	Attr	Default	Description
23:16	RO	06h	**Base Class** For PCI Express NTB port this field is hardwired to 06h, indicating it is a "Bridge Device".
15:8	RO	80h	**Sub-Class** For PCI Express NTB port, this field hardwired to 80h to indicate a "Other bridge type".
7:0	RO	00h	**Register-Level Programming Interface** This field is hardwired to 00h for PCI Express NTB port.

3.20.2.7 CLSR: Cacheline Size Register

Register:CLSR Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function: 0 Offset:0Ch			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0h	**Cacheline Size** This register is set as RW for compatibility reasons only. Cacheline size for IIO is always 64B. IIO hardware ignore this setting.

3.20.2.8 PLAT: Primary Latency Timer

This register denotes the maximum time slice for a burst transaction in legacy PCI 2.3 on the primary interface. It does not affect/influence PCI Express functionality.

Register:PLAT Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 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.

3.20.2.9 HDR: Header Type Register (Dev#3, PCIe NTB Sec Mode)

This register identifies the header layout of the configuration space.

Register:HDR Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function: 0 Offset:0Eh PCIE_ONLY			
Bit	**Attr**	**Default**	**Description**
7	RO	0	**Multi-function Device** This bit defaults to 0 for PCI Express NTB port.
6:0	RO	00h	**Configuration Layout** This field identifies the format of the configuration header layout. It is Type0 for PCI Express NTB port. The default is 00h, indicating a “non-bridge function”.

3.20.2.10 BIST: Built-In Self Test

This register is used for reporting control and status information of BIST checks within a PCI Express port. It is not supported in Intel® Xeon® processor C5500/C3500 series.

Register:BIST
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:0Fh

Bit	Attr	Default	Description
7:0	RO	0h	BIST_TST: BIST Tests Not supported. Hardwired to 00h

3.20.2.11 SB01BASE: Secondary BAR 0/1 Base Address (PCIE NTB Mode)

This register is BAR 0/1 for the secondary side of the NTB. This configuration register can be modified via configuration transaction from the secondary side of the NTB and can also be modified from the primary side of the NTB via MMIO transaction to Section 3.21.1.9, “SBAR0BASE: Secondary BAR 0/1 Base Address” .

Note: SW must program upper DW first and then lower DW. If lower DW is programmed first HW will clear the lower DW.

Register:SB01BASE
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:10h

Bit	Attr	Default	Description
63:15	RW	00h	**Secondary BAR 0/1 Base** This register is reflected into the BAR 0/1 register pair in the Configuration Space of the Secondary side of the NTB written by SW on a 32KB alignment.
14:04	RO	00h	**Reserved** Fixed size of 32KB.
3	RWO	1b	**Prefetchable** 1 = BAR points to Prefetchable memory (default) 0 = BAR points to Non-Prefetchable memory
2:1	RO	10b	**Type** Memory type claimed by BAR 2/3 is 64-bit addressable.
0	RO	0b	**Memory Space Indicator** BAR resource is memory (as opposed to I/O).

3.20.2.12 SB23BASE: Secondary BAR 2/3 Base Address (PCIE NTB Mode)

This register is BAR 2/3 for the secondary side of the NTB. This configuration register can be modified via configuration transaction from the secondary side of the NTB and can also be modified from the primary side of the NTB via MMIO transaction to Section 3.21.1.10, "SBAR2BASE: Secondary BAR 2/3 Base Address"

Note: SW must program upper DW first and then lower DW. If lower DW is programmed first HW will clear the lower DW.

Register default: 000000400000000CH

Register:SB23BASE
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:18h

Bit	Attr	Default	Description
63:nn	RWL	variable	**Secondary BAR 2/3 Base** Sets the location of the BAR written by SW ***Notes:*** • The "nn" indicates the least significant bit that is writable. The number of bits that are writable in this register is dictated by the value loaded into the SBAR23SZ register by the BIOS at initialization time (before BIOS PCI enumeration). • For the special case where SBAR23SZ = '0', bits 63:00 are all RO='0' resulting in the BAR being disabled.
(nn-1) : 12	RO	variable	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.19.3.21, "SBAR23SZ: Secondary BAR 2/3 Size" Granularity must be at least 4 KB. ***Note:*** For the special case where SBAR23SZ = '0', bits 63:00 are all RO='0' resulting in the BAR being disabled.
11:04	RO	00h	**Reserved** Granularity must be at least 4 KB.
03	RO	1b	**Prefetchable** BAR points to Prefetchable memory.
02:01	RO	10b	**Type** Memory type claimed by BAR 2/3 is 64-bit addressable.
00	RO	0b	**Memory Space Indicator** BAR resource is memory (as opposed to I/O).

3.20.2.13 SB45BASE: Secondary BAR 4/5 Base Address

This register is BAR 4/5 for the secondary side of the NTB. This configuration register can be modified via configuration transaction from the secondary side of the NTB and can also be modified from the primary side of the NTB via MMIO transaction to Section 3.21.1.11, "SBAR4BASE: Secondary BAR 4/5 Base Address"

Note: SW must program upper DW first and then lower DW. If lower DW is programmed first HW will clear the lower DW.

Register default: 000000800000000CH

Register:SB45BASE
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:20h

Bit	Attr	Default	Description
63:nn	RWL	variable	**Secondary BAR 4/5 Base** Sets the location of the BAR written by SW ***Notes:*** • The "nn" indicates the least significant bit that is writable. The number of bits that are writable in this register is dictated by the value loaded into the SBAR45SZ register by the BIOS at initialization time (before BIOS PCI enumeration). • For the special case where SBAR45SZ = '0', bits 63:00 are all RO='0' resulting in the BAR being disabled. • Default is set to 512 GB
(nn-1) : 12	RO	variable	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.19.3.22, "SBAR45SZ: Secondary BAR 4/5 Size" Granularity must be at least 4 KB. ***Note:*** For the special case where SBAR45SZ = '0', bits 63:00 are all RO='0' resulting in the BAR being disabled.
11:04	RO	00h	**Reserved** Granularity must be at least 4 KB.
03	RO	1b	**Prefetchable** BAR points to Prefetchable memory.
02:01	RO	10b	**Type** Memory type claimed by BAR 4/5 is 64-bit addressable.
00	RO	0b	**Memory Space Indicator** BAR resource is memory (as opposed to I/O).

3.20.2.14 SUBVID: Subsystem Vendor ID (Dev#3, PCIE NTB Sec Mode)

This register identifies the vendor of the subsystem.

Register:SUBVID
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:2Ch

Bit	Attr	Default	Description
15:0	RWO	0000h	**Subsystem Vendor ID:** This field must be programmed during boot-up to indicate the vendor of the system board. When any byte or combination of bytes of this register is written, the register value locks and cannot be further updated.

3.20.2.15 SID: Subsystem Identity (Dev#3, PCIE NTB Sec Mode)

This register identifies a particular subsystem.

Register:SID
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:2Eh

Bit	Attr	Default	Description
15:0	RWO	0000h	**Subsystem ID:** This field must be programmed during BIOS initialization. When any byte or combination of bytes of this register is written, the register value locks and cannot be further updated.

3.20.2.16 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.

Register:CAPPTR
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:34h

Bit	Attr	Default	Description
7:0	RWO	60h	**Capability Pointer** Points to the first capability structure for the device.

3.20.2.17 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 RW for compatibility purposes only.

Register:INTL
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:3Ch

Bit	Attr	Default	Description
7:0	RW	00h	**Interrupt Line** This bit is RW for devices that can generate a legacy INTx message and is needed only for compatibility purposes.

3.20.2.18 INTPIN: Interrupt Pin Register

The INTP register identifies legacy interrupts for INTA, INTB, INTC and INTD as determined by BIOS/firmware. These are emulated over the DMI port using the appropriate Assert_Intx commands.

Register:INTPIN
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:3Dh

Bit	Attr	Default	Description
7:0	RWO	01h	**INTP: 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:*** While the PCI spec. defines only one interrupt line (INTA#) for a single function device, the logic for the NTB has been modified to meet customer requests for programmability of the interrupt pin. BIOS should always set this to INTA# for standard OS's.

3.20.2.19 MINGNT: Minimum Grant Register

Register:INTPIN
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:3Eh

Bit	Attr	Default	Description
7:0	RO	00h	**Minimum Grant**: This register does not apply to PCI Express. It is hard-coded to “00”h.

3.20.2.20 MAXLAT: Maximum Latency Register

Register:MAXLAT
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:3Fh

Bit	Attr	Default	Description
7:0	RO	00h	**Maximum Latency**: This register does not apply to PCI Express. It is hard-coded to “00”h.

3.20.3 Device-Specific PCI Configuration Space - 0x40 to 0xFF

3.20.3.1 MSICAPID: MSI Capability ID

Register:MSICAPID
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:60h

Bit	Attr	Default	Description
7:0	RO	05h	**Capability ID** Assigned by PCI-SIG for MSI.

3.20.3.2 MSINXTPTR: MSI Next Pointer

Register:MSINXTPTR
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:61h

Bit	Attr	Default	Description
7:0	RWO	80h	**Next Ptr** This field is set to 80h for the next capability list (PCI Express capability structure) in the chain.

3.20.3.3 MSICTRL: MSI Control Register

Register:MSICTRL
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:62h

Bit	Attr	Default	Description
15:9	RV	00h	*Reserved.*
8	RO	1b	**Per-vector masking capable** This bit indicates that PCI Express ports support MSI per-vector masking.
7	RO	1b	**64-bit Address Capable** A PCI Express Endpoint must support the 64-bit Message Address version of the MSI Capability structure 1: Function is capable of sending 64-bit message address 0: Function is not capable of sending 64-bit message address. ***Notes:*** • For B0 stepping this field is RO = 1 • For A0 stepping this field is RO = 0 so can only be connected to CPU requiring 32b MSI address.
6:4	RW	000b	**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. See Table 91 for a discussion on how the interrupts are distributed amongst the various sources of interrupt based on the number of messages allocated by software for the PCI Express NTB port. Value Number of Messages Requested 000b = 1 001b = 2 010b = 4 011b = 8 100b = 16 101b = 32 110b = Reserved 111b = Reserved
3:1	RO	000b	**Multiple Message Capable** IIO's PCI Express NTB port supports one message for all internal events. Value Number of Messages Requested 000b = 1 001b = 2 010b = 4 011b = 8 100b = 16 101b = 32 110b = Reserved 111b = Reserved

Register:MSICTRL Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function:0 Offset:62h			
Bit	**Attr**	**Default**	**Description**
0	RW	0b	**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 PCI Express port to use MSI messages for RAS, provided bit 4 in Section 3.19.4.20, "MISCCTRLSTS: Misc. Control and Status Register" on page 216 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 3.19.4.20, "MISCCTRLSTS: Misc. Control and Status Register" on page 216) for details. ***Note:*** Software must disable INTx and MSI-X for this device when using MSI

3.20.3.4 MSIAR: MSI Lower Address Register

The MSI Lower Address Register (MSIAR) contains the lower 32b system specific address information to route MSI interrupts.

Register:MSIAR Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function:0 Offset:64h			
Bit	**Attr**	**Default**	**Description**
31:20	RW	0h	**Address MSB** This field specifies the 12 most significant bits of the 32-bit MSI address. This field is R/W.
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 IA32 processor
3	RW	0h	**Address Redirection Hint** 0: directed 1: redirectable
2	RW	0h	Address Destination Mode 0: physical 1: logical
1:0	RO	0h	*Reserved.*

3.20.3.5 MSIUAR: MSI Upper Address Register

The optional MSI Upper Address Register (MSIAR) contains the upper 32b system specific address information to route MSI interrupts.

Register:MSIUAR Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function:0 Offset:68h			
Bit	**Attr**	**Default**	**Description**
31:00	RW	00000000h	**Upper Address MSB** If the MSI Enable bit (bit 0 of the MSICTRL) is set, the contents of this register (if non-zero) specify the upper 32-bits of a 64-bit message address (AD[63::32]). If the contents of this register are zero, the function uses the 32 bit address specified by the message address register.

3.20.3.6 MSIDR: MSI Data Register

The MSI Data Register contains all the data (interrupt vector) related to MSI interrupts from the root ports.

Register:MSIDR Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function:0 Offset:6Ch			
Bit	**Attr**	**Default**	**Description**
31:16	RO	0000h	*Reserved.*
15	RW	0h	**Trigger Mode** 0 - Edge Triggered 1 - Level Triggered IIO does nothing with this bit other than passing it along to the Intel® QPI
14	RW	0h	**Level** 0 - Deassert 1 - Assert IIO does nothing with this bit other than passing it along to the Intel® QPI
13:12	RW	0h	**Don't care for IIO**
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 – *Reserved* 0111 – ExtINT - Not supported via MSI of root port 1000-1111 - *Reserved*
7:0	RW	0h	**Interrupt Vector** The interrupt vector (LSB) will be modified by the IIO to provide context sensitive interrupt information for different events that require attention from the processor. Depending on the number of Messages enabled by the processor, Table 91 illustrates how the IIO distributes these vectors

Table 92. MSI Vector Handling and Processing by IIO on Secondary Side

Number of Messages enabled by Software	Events	IV[7:0]
1	PD[15:00]	xxxxxxxx[1]

1. The term "xxxxxx" in the Interrupt vector denotes that software initializes them and IIO will not modify any of the "x" bits except the LSB as indicated in the table as a function of MMEN

3.20.3.7 MSIMSK: MSI Mask Bit Register

The Mask Bit register enables software to disable message sending on a per-vector basis.

Register:MSIMSK
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:70h

Bit	Attr	Default	Description
31:01	Rsvd P	0h	*Reserved*
00	RW	0h	**Mask Bit** For each Mask bit that is set, the PCI Express port is prohibited from sending the associated message. NTB supports up to 1 messages Corresponding bits are masked if set to '1'

3.20.3.8 MSIPENDING: MSI Pending Bit Register

The Mask Pending register enables software to defer message sending on a per-vector basis.

Register:MSIPENDING
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:74h

Bit	Attr	Default	Description
31:01	Rsvd P	0h	*Reserved*
00	RO	0h	**Pending Bits** For each Pending bit that is set, the PCI Express port has a pending associated message. NTB supports 1 message Corresponding bits are pending if set to '1'

3.20.3.9 MSIXCAPID: MSI-X Capability ID

Register:MSIXCAPID
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:80h

Bit	Attr	Default	Description
7:0	RO	11h	**Capability ID** Assigned by PCI-SIG for MSI-X.

3.20.3.10 MSIXNXTPTR: MSI-X Next Pointer

Register:MSIXNXTPTR
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:81h

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.

3.20.3.11 MSIXMSGCTRL: MSI-X Message Control Register

Register:MSIXMSGCTRL
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:82h

Bit	Attr	Default	Description
15	RW	0b	**MSI-X Enable** Software uses this bit to enable MSI-X method for signaling 0: NTB is prohibited from using MSI-X to request service 1: MSI-X method is chosen for NTB interrupts ***Note:*** Software must disable INTx and MSI for this device when using MSI-X
14	RW	0b	**Function Mask** If = 1b, all the vectors associated with the NTB are masked, regardless of the per vector mask bit state. If = 0b, each vector's mask bit determines whether the vector is masked or not. Setting or clearing the MSI-X function mask bit has no effect on the state of the per-vector Mask bit.
13:11	RO	0h	*Reserved.*

Register:MSIXMSGCTRL Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function:0 Offset:82h			
Bit	**Attr**	**Default**	**Description**
10:00	RO	003h	**Table Size** System software reads this field to determine the MSI-X Table Size N, which is encoded as N-1. For example, a returned value of "00000000011" indicates a table size of 4. NTB table size is 4, encoded as a value of 003h The value in this field depends on the setting of Section 3.20.3.23, "DEVCAP2: PCI Express Device Capabilities Register 2" bit 0. When SSCNTL, bit 0 = '0' (default) Table size is 4, encoded as a value of 003h When SSCNTL, bit 0 ='1' Table size is 1, encoded as a value of 000h

3.20.3.12 TABLEOFF_BIR: MSI-X Table Offset and BAR Indicator Register (BIR)

Register default: 00004000h

Register:TABLEOFF_BIR Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function:0 Offset:84h			
Bit	**Attr**	**Default**	**Description**
31:03	RO	00000800h	**Table Offset** MSI-X Table Structure is at offset 16K from the SB01BASE BAR address. See Section 3.21.2.1, "PMSIXTBL[0-3]: Primary MSI-X Table Address Register 0 - 3" for the start of details relating to MSI-X registers. ***Note:*** Offset placed at 16K so that it can also be visible through the primary BAR for debug purposes.
02:00	RO	0h	**Table BIR** Indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X Table into Memory Space. **BIR Value Base Address register** 0 10h 1 14h 2 18h 3 1Ch 4 20h 5 24h 6 Reserved 7 Reserved For a 64-bit Base Address register, the Table BIR indicates the lower DWORD.

3.20.3.13 PBAOFF_BIR: MSI-X Pending Bit Array Offset and BAR Indicator

Register default: 00005000h

Register:PBAOFF_BIR Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function:0 Offset:88h			
Bit	**Attr**	**Default**	**Description**
31:03	RO	00000A00h	**Table Offset** MSI-X PBA Structure is at offset 20K from the SB01BASE BAR address. See Section 3.21.3.4, "SMSIXPBA: Secondary MSI-X Pending Bit Array Register" for details. ***Note:*** Offset placed at 20K so that it can also be visible through the primary BAR for debug purposes.
02:00	RO	0h	**PBA BIR** Indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X Table into Memory Space. **BIR Value Base Address register** 0 10h 1 14h 2 18h 3 1Ch 4 20h 5 24h 6 Reserved 7 Reserved For a 64-bit Base Address register, the Table BIR indicates the lower DWORD.

3.20.3.14 PXPCAPID: PCI Express Capability Identity Register

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 3.0 configuration space.

Register:PXPCAPID Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function: 0 Offset:90h			
Bit	**Attr**	**Default**	**Description**
7:0	RO	10h	**Capability ID** Provides the PCI Express capability ID assigned by PCI-SIG. Required by *PCI Express Base Specification,* Revision 2.0 to be this value.

3.20.3.15 PXPNXTPTR: PCI Express Next Pointer Register

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 3.0 configuration space.

Register:PXPNXTPTR
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:91h

Bit	Attr	Default	Description
7:0	RWO	E0h	**Next Ptr** This field is set to the PCI PM capability.

3.20.3.16 PXPCAP: PCI Express Capabilities Register

The PCI Express Capabilities register identifies the PCI Express device type and associated capabilities.

Register:PXPCAP
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:92h

Bit	Attr	Default	Description
15:14	Rsvd P	00b	*Reserved*
13:9	RO	00000b	**Interrupt Message Number** Applies only to the RPs. 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 RP status registers are set. IIO assigns the first vector for PM/HP events and so this field is set to 0.
8	RWO	0b	**Slot Implemented** Applies only to the RPs for NTB this value is kept at 0b. 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	0000b	**Device/Port Type** This field identifies the type of device. 0000b = PCI Express Endpoint.
3:0	RWO	2h	**Capability Version** This field identifies the version of the PCI Express capability structure. Set to 2h for PCI Express devices for compliance with the extended base registers.

3.20.3.17 DEVCAP: PCI Express Device Capabilities Register

The PCI Express Device Capabilities register identifies device specific information for the device.

Register:DEVCAP
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:94h

Bit	Attr	Default	Description
31:29	Rsvd P	0h	*Reserved*
28	RO	0b	**Function Level Reset Capability** A value of 1b indicates the Function supports the optional Function Level Reset mechanism. NTB does not support this functionality
27:26	RO	0h	**Captured Slot Power Limit Scale** Does not apply to RPs or integrated devices This value is hardwired to 00h NTB is required to be able to receive the Set_Slot_Power_Limit message without error but simply discard the Message value. ***Note:*** Components with Endpoint, Switch, or PCI Express-PCI Bridge Functions that are targeted for integration on an adapter where total consumed power is below the lowest limit defined for the targeted form factor are permitted to ignore Set_Slot_Power_Limit Messages, and to return a value of 0 in the Captured Slot Power Limit Value and Scale fields of the Device Capabilities register
25:18	RO	00h	**Captured Slot Power Limit Value** Does not apply to RPs or integrated devices This value is hardwired to 00h NTB is required to be able to receive the Set_Slot_Power_Limit message without error but simply discard the Message value. ***Note:*** Components with Endpoint, Switch, or PCI Express-PCI Bridge Functions that are targeted for integration on an adapter where total consumed power is below the lowest limit defined for the targeted form factor are permitted to ignore Set_Slot_Power_Limit Messages, and to return a value of 0 in the Captured Slot Power Limit Value and Scale fields of the Device Capabilities register
17:16	Rsvd P	0h	*Reserved*
15	RO	1	**Role Based Error Reporting**: IIO is 1.1 compliant and so supports this feature
14	RO	0	**Power Indicator Present on Device** Does not apply to RPs or integrated devices
13	RO	0	**Attention Indicator Present** Does not apply to RPs or integrated devices
12	RO	0	**Attention Button Present** Does not apply to RPs or integrated devices

Register:DEVCAP
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:94h

Bit	Attr	Default	Description
11:9	RWO	110b	**Endpoint L1 Acceptable Latency** This field indicates the acceptable latency that an Endpoint can withstand due to the transition from L1 state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L1 Acceptable Latency number to compare against the L1 Exit Latencies reported (see below) by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L1 entry can be used with no loss of performance. Defined encodings are: 000b Maximum of 1 us 001b Maximum of 2 us 010b Maximum of 4 us 011b Maximum of 8 us 100b Maximum of 16 us 101b Maximum of 32 us 110b Maximum of 64 us 111b No limit BIOS must program this value
8:6	RWO	000b	**Endpoint L0s Acceptable Latency** This field indicates the acceptable total latency that an Endpoint can withstand due to the transition from L0s state to the L0 state. It is essentially an indirect measure of the Endpoint's internal buffering. Power management software uses the reported L0s Acceptable Latency number to compare against the L0s exit latencies reported by all components comprising the data path from this Endpoint to the Root Complex Root Port to determine whether ASPM L0s entry can be used with no loss of performance. Defined encodings are: 000b Maximum of 64 ns 001b Maximum of 128 ns 010b Maximum of 256 ns 011b Maximum of 512 ns 100b Maximum of 1 us 101b Maximum of 2 us 110b Maximum of 4 us 111b No limit BIOS must program this value
5	RO	1	**Extended Tag Field Supported** IIO devices support 8-bit tag 1 = Maximum Tag field is 8 bits 0 = Maximum Tag field is 5 bits
4:3	RO	00b	**Phantom Functions Supported** IIO does not support phantom functions. 00b = No Function Number bits are used for Phantom Functions
2:0	RO	001b	**Max Payload Size Supported** IIO supports 256B payloads on PCI Express ports 001b = 256 bytes max payload size

3.20.3.18 DEVCTRL: PCI Express Device Control Register (PCIE NTB Secondary)

The PCI Express Device Control register controls PCI Express specific capabilities parameters associated with the device.

Register:DEVCTRL
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:98h
PCIE_ONLY

Bit	Attr	Default	Description
15	RsvdP	0h	*Reserved.*
14:12	RO	000	**Max_Read_Request_Size** This field sets maximum Read Request size generated by the Intel® Xeon® processor C5500/C3500 series as a requestor. The corresponding IOU logic in the Intel® Xeon® processor C5500/C3500 series associated with the PCIExpress port must not generate read requests with size exceeding the set value. 000: 128B max read request size 001: 256B max read request size 010: 512B max read request size 011: 1024B max read request size 100: 2048B max read request size 101: 4096B max read request size 110: Reserved 111: Reserved ***Note:*** The Intel® Xeon® processor C5500/C3500 series will not generate read requests larger than 64 B on the outbound side due to the internal Micro-architecture (CPU initiated, DMA, or Peer to Peer). Hence the field is set to 000b encoding.
11	RO	0	**Enable No Snoop** Not applicable since the NTB is never the originator of a TLP. This bit has no impact on forwarding of NoSnoop attribute on peer requests.
10	RO	0	**Auxiliary Power Management Enable** Not applicable to IIO
9	RO	0	**Phantom Functions Enable** Not applicable to IIO since it never uses phantom functions as a requester.
8	RW	0h	**Extended Tag Field Enable** This bit enables the PCI Express/DMI ports to use an 8-bit Tag field as a requester.
7:5	RW	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 IIO must handle TLPs as large as the set value. As a requester (i.e. for requests where IIOs 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 DMI port aliases to 128B) others: alias to 128B This field is RW for PCI Express ports. ***Note:*** Bit 7:5 must be programmed to the same value on both primary and secondary side of the NTB

Register:DEVCTRL
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset:98h
PCIE_ONLY

Bit	Attr	Default	Description
4	RO	0	**Enable Relaxed Ordering** Not applicable since the NTB is never the originator of a TLP. 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/DMI ports. This bit controls the reporting of unsupported requests that IIO itself detects on requests its receives from a PCI Express/DMI port. 0: Reporting of unsupported requests is disabled 1: Reporting of unsupported requests is enabled. ***Note:*** This register provides no functionality on the secondary side of the NTB. The NTB never reports errors outbound. All errors are sent towards local host that are detected on the link.
2	RO	0	**Fatal Error Reporting Enable** Applies only to the PCI Express RP/PCI Express NTB Secondary interface/DMI ports. Controls the reporting of fatal errors that IIO detects on the PCI Express/DMI interface. 0: Reporting of Fatal error detected by device is disabled 1: Reporting of Fatal error detected by device is enabled ***Note:*** This register provides no functionality on the secondary side of the NTB. The NTB never reports errors outbound. All errors are sent towards local host that are detected on the link.
1	RO	0	**Non Fatal Error Reporting Enable** Applies only to the PCI Express RP/PCI Express NTB Secondary interface/DMI ports. Controls the reporting of non-fatal errors that IIO detects on the PCI Express/DMI interface. 0: Reporting of Non Fatal error detected by device is disabled 1: Reporting of Non Fatal error detected by device is enabled ***Note:*** This register provides no functionality on the secondary side of the NTB. The NTB never reports errors outbound. All errors are sent towards local host that are detected on the link.
0	RO	0	**Correctable Error Reporting Enable** Applies only to the PCI Express RP/PCI Express NTB Secondary interface/DMI ports. Controls the reporting of correctable errors that IIO detects on the PCI Express/DMI interface 0: Reporting of link Correctable error detected by the port is disabled 1: Reporting of link Correctable error detected by port is enabled ***Note:*** This register provides no functionality on the secondary side of the NTB. The NTB never reports errors outbound. All errors are sent towards local host that are detected on the link.

3.20.3.19 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
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function: 0
Offset: 9Ah

Bit	Attr	Default	Description
15:6	RsvdZ	000h	*Reserved.*
5	RO	0h	**Transactions Pending**: Does not apply. Bit is hardwired to 0 NTB is a special case bridging device following the rule below. *PCI Express Base Specification,* Revision 2.0 states. Root and Switch Ports implementing only the functionality required by this document do not issue Non-Posted Requests on their own behalf, and therefore are not subject to this case. Root and Switch Ports that do not issue Non-Posted Requests on their own behalf hardwire this bit to 0b.
4	RO	0	**AUX Power Detected** Does not apply to IIO
3	RW1C	0	**Unsupported Request Detected** This bit applies only to the root/DMI ports.This bit indicates that the NTB secondary 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 RP received and it detected them as unsupported requests (e.g. address decoding failures that the RP detected on a packet, receiving inbound lock reads, BME bit is clear etc.). This bit is not set on peer2peer completions with UR status that are forwarded by the RP to the PCIE link. 0: No unsupported request detected by the RP
2	RW1C	0	**Fatal Error Detected** This bit indicates that a fatal (uncorrectable) error is detected by the NTB secondary 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
1	RW1C	0	**Non Fatal Error Detected** This bit gets set if a non-fatal uncorrectable error is detected by the NTB secondary 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 NTB secondary 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

3.20.3.20 LNKCAP: PCI Express Link Capabilities Register

The Link Capabilities register identifies the PCI Express specific link capabilities

Note: This register is a secondary view into the LNKCAP register. BIOS must set some RWO configuration bits prior to use. See Section 3.19.4.23, "LNKCAP: PCI Express Link Capabilities Register" .

Register:LNKCAP
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:9Ch

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	RsvdP	0h	*Reserved.*
21	RO	0	**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**: IIO 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**: IIO supports reporting a surprise down error condition
18	RO	0	**Clock Power Management**: Does not apply to IIO.
17:15	RO	010	**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 us 001: 1 us to less than 2 us 010: 2 us to less than 4 us 011: 4 us to less than 8 us 100: 8 us to less than 16 us 101: 16 us to less than 32 us 110: 32 us to 64 us 111: More than 64us

Register:LNKCAP
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:9Ch

Bit	Attr	Default	Description
14:12	RO	011	**L0s Exit Latency** This field indicates the L0s exit latency (i.e L0s to L0) for the PCI-Express port. 000: Less than 64 ns 001: 64 ns to less than 128 ns 010: 128 ns to less than 256 ns 011: 256 ns to less than 512 ns 100: 512 ns to less than 1 is 101: 1 is to less than 2 is 110: 2 is to 4 is 111: More than 4 is
11:10	RO	11	**Active State Link PM Support** This field indicates the level of active state power management supported on the given PCI-Express port. 00: Disabled 01: L0s Entry Supported 10: Reserved 11: L0s and L1 Supported
9:4	RO	001000b	**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 - Reserved
3:0	RO	0010b	**Link Speeds Supported** IIO supports both 2.5Gbps and 5Gbps speeds if Gen2_OFF fuse is OFF else it supports only Gen1 0001b = 2.5 GT/s Link speed supported 0010b = 5.0 GT/s and 2.5 GT/s link speed supported This field defaults to 0010b if Gen2_OFF fuse is OFF This field defaults to 0001b if Gen2_OFF fuse is ON

1. There are restrictions with routing x2 lanes from IIO to a slot. See Section 3.3, "PCI Express Link Characteristics - Link Training, Bifurcation, Downgrading and Lane Reversal Support" (IOH Platform Architecture Specification) for details.

3.20.3.21 LNKCON: PCI Express Link Control Register

The PCI Express Link Control register controls the PCI Express Link specific parameters

Note: This register is a secondary view into the LNKCAP register. Some additional controllability is available through the primary side equivalent register. See Section 3.19.4.24, "LNKCON: PCI Express Link Control Register"

Note: In NTB/RP mode RP will program this register. In NTB/NTB mode local host BIOS will program this register.

Register:LNKCON
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:A0h

Bit	Attr	Default	Description
15:12	RsvdP	0h	**Reserved**
11	RsvdP	0b	**Link Autonomous Bandwidth Interrupt Enable -** This bit is not applicable and is reserved for Endpoints
10	RsvdP	0b	**Link Bandwidth Management Interrupt Enable -** This bit is not applicable and is reserved for Endpoints
09	RO	0b	**Hardware Autonomous Width Disable:** IIO never changes a configured link width for reasons other than reliability.
08	RO	0b	**Enable Clock Power Management**: N/A to IIO
07	RW	0b	**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 2.0 for details.
06	RW	0b	**Common Clock Configuration** IIO does nothing with this bit
05	RsvdP	0b	**Retrain Link** This bit is not applicable and is reserved for Endpoints

Register:LNKCON
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:A0h

Bit	Attr	Default	Description
04	RsvdP	0b	**Link Disable** This bit is not applicable and is reserved for Endpoints
03	RO	0b	**Read Completion Boundary** Set to zero to indicate IIO could return read completions at 64B boundaries ***Note:*** NTB is not PCIE compliant in this respect. NTB is only capable of 64B RCB. If connecting to non IA IP and the IP does the optional 128B RCB check on received packets, packets will be seen as malformed. This is not an issue with any Intel IP.
02	RsvdP	0b	**Reserved**.
01:00	RW	00b	**Active State Link PM Control**: When 01b or 11b, L0s on transmitter is enabled, otherwise it is disabled. Defined encodings are: 00b Disabled 01b L0s Entry Enabled 10b L1 Entry Enabled 11b L0s and L1 Entry Enabled ***Note:*** "L0s Entry Enabled" indicates the Transmitter entering L0s is supported. The Receiver must be capable of entering L0s even when the field is disabled (00b). ASPM L1 must be enabled by software in the Upstream component on a Link prior to enabling ASPM L1 in the Downstream component on that Link. When disabling ASPM L1, software must disable ASPM L1 in the Downstream component on a Link prior to disabling ASPM L1 in the Upstream component on that Link. ASPM L1 must only be enabled on the Downstream component if both components on a Link support ASPM L1.

3.20.3.22 LNKSTS: PCI Express Link Status Register

Note: This register is a secondary view into the LNKSTS register. BIOS must set some registers prior to use. See Section 3.19.4.25, "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 etc.

Register:LNKSTS
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:A2h

Bit	Attr	Default	Description
15	Rsvd P	0	**Link Autonomous Bandwidth Status** This bit is not applicable and is reserved for Endpoints
14	Rsvd P	0	**Link Bandwidth Management Status** This bit is not applicable and is reserved for Endpoints
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 IIO 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

Register:LNKSTS
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:A2h

Bit	Attr	Default	Description
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 IIO hardware clears this bit once LTSSM has exited the recovery/ configuration state. See the *PCI Express Base Specification,* Revision 2.0 for details of which states within the LTSSM would set this bit and which states would clear this bit.
10	RO	0	*Reserved*
9:4	RO	0h	**Negotiated Link Width** 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 IIO. 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 0x16 for a link width of x16. 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	1h	**Current Link Speed** This field indicates the negotiated Link speed of the given PCI Express Link. 0001- 2.5 Gbps 0010 - 5Gbps (IIO will never set this value when Gen2_OFF fuse is blown) Others - *Reserved* 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.

3.20.3.23 DEVCAP2: PCI Express Device Capabilities Register 2

Register:DEVCAP2
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:B4h

Bit	Attr	Default	Description
31:6	RO	0h	Reserved
5	RO	0	Alternative RID Interpretation (ARI) Capable - This bit is set to 1b indicating Root Port supports this capability. NOTE: This bit is reserved and not applicable for Endpoints
4	RO	1	Completion Timeout Disable Supported - IIO supports disabling completion timeout
3:0	RO	1110b	Completion Timeout Values Supported – This field indicates device support for the optional Completion Timeout programmability mechanism. This mechanism allows system software to modify the Completion Timeout range. Bits are one-hot encoded and set according to the table below to show timeout value ranges supported. A device that supports the optional capability of Completion Timeout Programmability must set at least two bits. Four time values ranges are defined: Range A: 50us to 10ms Range B: 10ms to 250ms Range C: 250ms to 4s Range D: 4s to 64s Bits ares 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 & D 1111b: Range A, B, C & D All other values are reserved. IIO supports timeout values up to 10ms-64s.

3.20.3.24 DEVCTRL2: PCI Express Device Control Register 2

This register is intended to be controlled from the primary side of the NTB at the mirror location of BDF 030, Offset 1B8h. This register provides visibility from the secondary side of the NTB.

Register:DEVCTRL2
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:B8h

Bit	Attr	Default	Description
15:5	RO	0h	Reserved
4	RW	0	**Completion Timeout Disable** – When set to 1b, this bit disables the Completion Timeout mechanism for all NP tx that IIO issues on the PCIE/DMI link and in the case of Intel® QuickData Technology DMA, for all NP tx that DMA issues upstream. When 0b, completion timeout is enabled. Software can change this field while there is active traffic in the root port.
3:0	RW	0000b	**Completion Timeout Value on NP Tx that IIO 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 = 10ms to 50ms 0001b = Reserved (IIO aliases to 0000b) 0010b = Reserved (IIO aliases to 0000b) 0101b = 16ms to 55ms 0110b = 65ms to 210ms 1001b = 260ms to 900ms 1010b = 1s to 3.5s 1101b = 4s to 13s 1110b = 17s to 64s When OS selects 17s to 64s range, Section , “BDF 030 Offset 232H. This register exist in both RP and NTB modes. It is documented in RP Section 3.4.5.34, “XPGLBERRPTR - XP Global Error Pointer Register”. See Volume 2 of the Datasheet.” on page 237 further controls the timeout value within that range. For all other ranges selected by OS, the timeout value within that range is fixed in IIO hardware. Software can change this field while there is active traffic in the root port. This value will also be used to control **PME_TO_ACK Timeout. That is this** field sets the timeout value for receiving a 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 register is set to a 1b.

3.20.3.25 SSCNTL: Secondary Side Control

This register provides secondary side control of NTB functions.

Register:SSCNTL
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:D4h

Bit	Attr	Default	Description
15:01	RO	0h	**Reserved**
00	RW	0b	NTB Secondary side - MSI-X Single Message Vector: This bit when set, causes only a single MSI-X message to be generated if MSI-X is enabled. This bit affects the default value of the MSI-X Table Size field in the Section 3.20.3.11, "MSIXMSGCTRL: MSI-X Message Control Register"

3.20.3.26 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
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:E0h

Bit	Attr	Default	Description
31:27	RO	00000b	**PME Support** Indicates the PM states within which the function is capable of sending a PME message. NTB secondary side does not forward PME messages. Bit 31 = D3cold Bit 30 = D3hot Bit 29 = D2 Bit 28 = D1 Bit 27 = D0
26	RO	0b	**D2 Support** IIO does not support power management state D2.
25	RO	0b	**D1 Support** IIO does not support power management state D1.
24:22	RO	000b	**AUX Current** Device does not support auxiliary current
21	RO	0b	**Device Specific Initialization** Device initialization is not required
20	RV	0b	*Reserved*.

Register:PMCAP Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function:0 Offset:E0h			
Bit	**Attr**	**Default**	**Description**
19	RO	0b	**PME Clock** This field is hardwired to 0h as it does not apply to PCI Express.
18:16	RO	011b	**Version** This field is set to 3h (PM 1.2 compliant) as version number for all PCI Express ports.
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.

3.20.3.27 PMCSR: Power Management Control and Status Register

This register provides status and control information for PM events in the PCI Express port of the IIO.

Register:PMCSR Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset Bus:M Device:0 Function:0 Offset:E4h			
Bit	**Attr**	**Default**	**Description**
31:24	RO	00h	**Data** Not relevant for IIO
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	RsvdP	0h	*Reserved.*
15	RO	0h	**PME Status** Applies only to RPs This bit is hard-wired to read-only 0, since this function does not support **PME#** generation from any power state.
14:13	RO	0h	**Data Scale** Not relevant for IIO
12:9	RO	0h	**Data Select** Not relevant for IIO
8	RO	0h	**PME Enable** Applies only to RPs. 0: Disable ability to send PME messages when an event occurs 1: Enables ability to send PME messages when an event occurs

Register:PMCSR
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:E4h

Bit	Attr	Default	Description
7:4	RsvdP	0h	*Reserved.*
3	RWO	1	No Soft Reset Indicates IIO does not reset its registers when transitioning from D3hot to D0. ***Note:*** This bit must be written by BIOS to a '1' so that this register bit cannot be cleared.
2	RsvdP	0h	*Reserved.*
1:0	RW	0h	**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. 00: D0 01: D1 (not supported by IIO) 10: D2 (not supported by IIO) 11: D3_hot 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. All devices will respond to only Type 0 configuration transactions when in D3hot state (RP 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).

3.20.3.28 SEXTCAPHDR: Secondary Extended Capability Header

This register identifies the capability structure and points to the next structure. There are no additional capability structures so this register has been made all zeros.

Register:SEXTCAPHDR
Bar:PB01BASE + 500h + Offset, SB01BASE + 500h + Offset
Bus:M
Device:0
Function:0
Offset:100h

Bit	Attr	Default	Description
31:20	RO	000h	**Next Capability Offset** This field points to the next Capability in extended configuration space.
19:16	RO	0h	**Capability Version** Set to 1h for this version of the PCI Express logic
15:0	RO	0000h	**PCI Express Extended CAP_ID** Assigned for Vendor specific Capability

3.21 NTB MMIO Space

NTB MMIO space consists of a shared set of MMIO registers (shadowed), primary side MMIO registers and secondary side MMIO registers.

3.21.1 NTB Shadowed MMIO Space

All shadow registers are visible from the primary side of the NTB. Only some of the shadow registers are visible from the secondary side of the NTB. See each register description for visibility.

Table 93. NTB MMIO Shadow Registers

PBAR2LMT		00h	SPAD0	80h
		04h	SPAD1	84h
PBAR4LMT		08h	SPAD2	88h
		0Ch	SPAD3	8Ch
PBAR2XLAT		10h	SPAD4	90h
		14h	SPAD5	94h
PBAR4XLAT		18h	SPAD6	98h
		1Ch	SPAD7	9Ch
SBAR2LMT		20h	SPAD8	A0h
		24h	SPAD9	A4h
SBAR4LMT		28h	SPAD10	A8h
		2Ch	SPAD11	ACh
SBAR2XLAT		30h	SPAD12	B0h
		34h	SPAD13	B4h
SBAR4XLAT		38h	SPAD14	B8h
		3Ch	SPAD15	BCh
SBAR0BASE		40h	SPADSEMA4	C0h
		44h		C4h
SBAR2BASE		48h		C8h
		4Ch		CCh
SBAR4BASE		50h	RSDBMSIXV70	D0h
		54h	RSDBMSIXV158	D4h
NTBCNTL		58h		D8h
CBDF	SBDF	5Ch		DCh
PDBMSK	PDOORBELL	60h	WCCNTRL	E0h
SDBMSK	SDOORBELL	64h		E4h
		68h		E8h
		6Ch		ECh
	USMEMMISS	70h		F0h

Table 93. NTB MMIO Shadow Registers

	74h		F4h
	78h		F8h
	7Ch		FCh

Secondary Link State - 1 bit (trained or untrained) (change generates interrupt)

Table 94. NTB MMIO Map

B2BSPAD0		100h	180h
B2BSPAD1		104h	184h
B2BSPAD2		108h	188h
B2BSPAD3		10Ch	18Ch
B2BSPAD4		110h	190h
B2BSPAD5		114h	194h
B2BSPAD6		118h	198h
B2BSPAD7		11Ch	19Ch
B2BSPAD8		120h	1A0h
B2BSPAD9		124h	1A4h
B2BSPAD10		128h	1A8h
B2BSPAD11		12Ch	1ACh
B2BSPAD12		130h	1B0h
B2BSPAD13		134h	1B4h
B2BSPAD14		138h	1B8h
B2BSPAD15		13Ch	1BCh
	B2BDOORBELL	140h	1C0h
B2BBAR0XLAT		144h	1C4h
B2BBAR0XLAT		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

3.21.1.1 PBAR2LMT: Primary BAR 2/3 Limit

This register contains a value used to limit the size of the window exposed by 64-bit BAR 2/3 to a size less than the power-of-two expressed in the Primary BAR 2/3 pair. This register is written by the NTB device driver and will contain the formulated sum of the base address plus the size of the BAR. This final value equates to the highest address that will be accepted through this port. Accesses to the memory area above this register (and below Base + Window Size) will return Unsupported Request.

Note: If the value in PBAR2LMT is set to a non-zero value less than the value in Section 3.19.2.12, "PB23BASE: Primary BAR 2/3 Base Address" hardware will force the value in PBAR2LMT to be zero and the full size of the window defined by Section 3.19.3.19, "PBAR23SZ: Primary BAR 2/3 Size" will be used.

Note: If the value in PBAR2LMT is set equal to the value in PB23BASE the memory window for PB23BASE is disabled.

Note: If the value in PBAR2LMT is set to a value greater than the value in the PB23BASE plus 2^PBAR23SZ hardware will force the value in PBAR2LMT to be zero and the full size of the window defined by PBAR23SZ will be used.

Note: If PBAR2LMT is zero the full size of the window defined by PBAR23SZ will be used.

Register:PBAR2LMT
Bar:PB01BASE, SB01BASE
Offset:00h

Bit	Attr	Default	Description
63:40	RO	00h	**Reserved** Intel® Xeon® processor C5500/C3500 series limited to 40bit addressing
39:12	Bar: Attr PB01BASE: RW else: RO	00h	**Primary BAR 2/3 Limit** Value representing the size of the memory window exposed by Primary BAR 2/3. A value of 00h will disable this register's functionality, resulting in a BAR window equal to that described by the BAR
11:00	RO	00h	**Reserved** Limit register has a granularity of 4 KB (2^{12})

3.21.1.2 PBAR4LMT: Primary BAR 4/5 Limit

This register contains a value used to limit the size of the window exposed by 64-bit BAR 4/5 to a size less than the power-of-two expressed in the Primary BAR 4/5 pair. This register is written by the NTB device driver and will contain the formulated sum of the base address plus the size of the BAR. This final value equates to the highest address that will be accepted through this port. Accesses to the memory area above this register (and below Base + Window Size) will return Unsupported Request.

Note: If the value in PBAR4LMT is set to a value less than the value in Section 3.19.2.13, "PB45BASE: Primary BAR 4/5 Base Address" hardware will force the value in PBAR4LMT to be zero and the full size of the window defined by Section 3.19.3.20, "PBAR45SZ: Primary BAR 4/5 Size" will be used.

Note: If the value in PBAR4LMT is set equal to the value in PB45BASE the memory window for PB45BASE is disabled.

Note: If the value in PBAR4LMT is set to a value greater than the value in the PB45BASE plus 2^PBAR45SZ hardware will force the value in PBAR4LMT to be zero and the full size of the window defined by PBAR45SZ will be used.

Note: If PBAR4LMT is zero the full size of the window defined by PBAR45SZ will be used.

Register:PBAR4LMT
Bar:PB01BASE, SB01BASE
Offset:08h

Bit	Attr	Default	Description
63:40	RO	00h	**Reserved** Intel® Xeon® processor C5500/C3500 series limited to 40bit addressing
39:12	Bar: Attr PB01BASE: RW else: RO	00h	**Primary BAR 4/5 Limit** Value representing the size of the memory window exposed by Primary BAR 4/5. A value of 00h will disable this register's functionality, resulting in a BAR window equal to that described by the BAR
11:0	RO	00h	**Reserved** Limit register has a granularity of 4 KB (2^{12})

3.21.1.3 PBAR2XLAT: Primary BAR 2/3 Translate

This register contains a value used to direct accesses into the memory located on the Secondary side of the NTB made from the Primary side of the NTB through the window claimed by BAR 2/3 on the primary side. The register contains the base address of the Secondary side memory window.

Note: There is no hardware enforced limit for this register, care must be taken when setting this register to stay within the addressable range of the attached system.

Register default: 0000004000000000H

Register:PBAR2XLAT
Bar:PB01BASE, SB01BASE
Offset:10h

Bit	Attr	Default	Description
63:nn	RWL	variable	**Primary BAR 2/3 Translate** The aligned base address into Secondary side memory. ***Notes:*** • Default is set to 256 GB • These bits appear as RW to SW
(nn-1) : 12	RO	00h	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.19.2.12, "PB23BASE: Primary BAR 2/3 Base Address"
11:00	RO	variable	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.19.2.12, "PB23BASE: Primary BAR 2/3 Base Address"

3.21.1.4 PBAR4XLAT: Primary BAR 4/5 Translate

This register contains a value used to direct accesses into the memory located on the Secondary side of the NTB made from the Primary side of the NTB through the window claimed by BAR 4/5 on the primary side. The register contains the base address of the Secondary side memory window.

Note: There is no hardware enforced limit for this register, care must be taken when setting this register to stay within the addressable range of the attached system.

Register default: 0000008000000000H

Register:PBAR4XLAT
Bar:PB01BASE, SB01BASE
Offset:18h

Bit	Attr	Default	Description
63:nn	RWL	variable	**Primary BAR 4/5 Translate** The aligned base address into Secondary side memory. ***Notes:*** • Default is set to 512 GB • These bits appear as RW to SW
(nn-1) : 12	RO	00h	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.19.2.13, "PB45BASE: Primary BAR 4/5 Base Address"
11:00	RO	variable	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.19.2.13, "PB45BASE: Primary BAR 4/5 Base Address"

3.21.1.5 SBAR2LMT: Secondary BAR 2/3 Limit

This register contains a value used to limit the size of the window exposed by 64-bit BAR 2/3 to a size less than the power-of-two expressed in the Secondary BAR 2/3 pair. This register is written by the NTB device driver and will contain the formulated sum of the base address plus the size of the BAR. This final value equates to the highest address that will be accepted through this port. Accesses to the memory area above this register (and below Base + Window Size) will return Unsupported Request.

Note: If the value in SBAR2LMT is set to a value less than the value in Section 3.20.2.12, "SB23BASE: Secondary BAR 2/3 Base Address (PCIE NTB Mode)" hardware will force the value in SBAR2LMT to be zero and the full size of the window defined by Section 3.19.3.21, "SBAR23SZ: Secondary BAR 2/3 Size" will be used.

Note: If the value in SBAR2LMT is set equal to the value in SB23BASE the memory window for SB23BASE is disabled.

Note: If the value in SBAR2LMT is set to a value greater than the value in the SB23BASE plus 2^SBAR23SZ hardware will force the value in SBAR2LMT to be zero and the full size of the window defined by SBAR23SZ will be used.

Note: If SBAR2LMT is zero the full size of the window defined by SBAR23SZ will be used.

Register:SBAR2LMT
Bar:PB01BASE, SB01BASE
Offset:20h

Bit	Attr	Default	Description
63:12	RW	00h	**Secondary BAR 2/3 Limit** Value representing the size of the memory window exposed by Secondary BAR 2/3. A value of 00h will disable this register's functionality, resulting in a BAR window equal to that described by the BAR In the case of NTB/NTB SAttr access type is a don't care
11:00	RO	00h	**Reserved** Limit register has a granularity of 4 KB (2^{12})

3.21.1.6 SBAR4LMT: Secondary BAR 4/5 Limit

This register contains a value used to limit the size of the window exposed by 64-bit BAR 4/5 to a size less than the power-of-two expressed in the Secondary BAR 4/5 pair. This register is written by the NTB device driver and will contain the formulated sum of the base address plus the size of the BAR. This final value equates to the highest address that will be accepted through this port. Accesses to the memory area above this register (and below Base + Window Size) will return Unsupported Request.

Note: If the value in SBAR4LMT is set to a value less than the value in Section 3.20.2.13, "SB45BASE: Secondary BAR 4/5 Base Address" hardware will force the value in SBAR4LMT to be zero and the full size of the window defined by Section 3.19.3.22, "SBAR45SZ: Secondary BAR 4/5 Size" will be used.

Note: If the value in SBAR4LMT is set equal to the value in SB45BASE the memory window for SB45BASE is disabled.

Note: If the value in SBAR4LMT is set to a value greater than the value in the SB45BASE plus 2^SBAR45SZ hardware will force the value in SBAR4LMT to be zero and the full size of the window defined by SBAR45SZ will be used.

Note: If SBAR4LMT is zero the full size of the window defined by SBAR45SZ will be used.

Register:SBAR4LMT
Bar:PB01BASE, SB01BASE
Offset:28h

Bit	Attr	Default	Description
63:12	RW	00h	**Secondary BAR 4/5 Limit** Value representing the size of the memory window exposed by Secondary BAR 4/5. A value of 00h will disable this register's functionality, resulting in a BAR window equal to that described by the BAR
11:00	RO	00h	**Reserved** Limit register has a granularity of 4 KB (2^{20})

3.21.1.7 SBAR2XLAT: Secondary BAR 2/3 Translate

This register contains a value used to direct accesses into the memory located on the Primary side of the NTB made from the Secondary side of the NTB through the window claimed by BAR 2/3 on the secondary side. The register contains the base address of the Primary side memory window.

Note: NTB will translate full 64b range. Switch logic will perform address range checks for both normal and VT-d flows.

Register:SBAR2XLAT
Bar:PB01BASE, SB01BASE
Offset:30h

Bit	Attr	Default	Description
63:nn	RWL	00h	**Secondary BAR 2/3 Translate** The aligned base address into Primary side memory. ***Note:*** Primary side access will appear as RW to SW. Secondary side access will appear as RO
(nn-1) : 12	RWL	00h	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.20.2.12, "SB23BASE: Secondary BAR 2/3 Base Address (PCIE NTB Mode)" ***Note:*** Attr will appear as RO to SW
11:00	RO	variable	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.20.2.12, "SB23BASE: Secondary BAR 2/3 Base Address (PCIE NTB Mode)" ***Note:*** Attr will appear as RO to SW

3.21.1.8 SBAR4XLAT: Secondary BAR 4/5 Translate

This register contains a value used to direct accesses into the memory located on the Primary side of the NTB made from the Secondary side of the NTB through the window claimed by BAR 4/5 on the secondary side. The register contains the base address of the Primary side memory window.

Note: NTB will translate full 64b range. Switch logic will perform address range checks for both normal and VT-d flows.

Register:SBAR4XLAT
Bar:PB01BASE, SB01BASE
Offset:38h

Bit	Attr	Default	Description
63:nn	RWL	00h	**Secondary BAR 4/5Translate** The aligned base address into Primary side memory. ***Note:*** Primary side access will appear as RW to SW. Secondary side access will appear as RO
(nn-1) : 12	RWL	00h	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.20.2.13, "SB45BASE: Secondary BAR 4/5 Base Address" ***Note:*** Attr will appear as RO to SW
11:00	RO	variable	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. Set by Section 3.20.2.13, "SB45BASE: Secondary BAR 4/5 Base Address" ***Note:*** Attr will appear as RO to SW

3.21.1.9 SBAR0BASE: Secondary BAR 0/1 Base Address

This register is mirrored from the BAR 0/1 register pair in the Configuration Space of the Secondary side of the NTB. The register is used by the processor on the primary side of the NTB to examine and load the BAR 0/1 register pair on the Secondary side of the NTB.

Register:SBAR0BASE
Bar:PB01BASE, SB01BASE
Offset:40h

Bit	Attr	Default	Description
63:15	RW	00h	**Secondary BAR 0/1 Base** This register is reflected into the BAR 0/1 register pair in the Configuration Space of the Secondary side of the NTB.
14:04	RO	00h	**Reserved** Fixed size of 32K B.
03	RWO	1b	**Prefetchable** 1 = BAR points to Prefetchable memory (default) 0 = BAR points to Non-Prefetchable memory
02:01	RO	10b	**Type** Memory type claimed by BAR 0/1 is 64-bit addressable.
00	RO	0b	**Memory Space Indicator** BAR resource is memory (as opposed to I/O).

3.21.1.10 SBAR2BASE: Secondary BAR 2/3 Base Address

This register is mirrored from the BAR 2/3 register pair in the Configuration Space of the Secondary side of the NTB. The register is used by the processor on the primary side of the NTB to examine and load the BAR 2/3 register pair on the Secondary side of the NTB.

Register:SBAR2BASE
Bar:PB01BASE, SB01BASE
Offset:48h

Bit	Attr	Default	Description
63:nn	RWL	00h	**Secondary BAR 2/3 Base** This register is reflected into the BAR 2/3 register pair in the Configuration Space of the Secondary side of the NTB. ***Note:*** These bits will appear to SW as RW.
(nn-1) : 12	RWL	00h	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. ***Note:*** These bits will appear to SW as RO.
11:04	RO	00h	**Reserved** Granularity must be at least 4KB.
03	RO	1b	**Prefetchable** BAR points to Prefetchable memory.
02:01	RO	10b	**Type** Memory type claimed by BAR 2/3 is 64-bit addressable.
00	RO	0b	**Memory Space Indicator** BAR resource is memory (as opposed to I/O).

3.21.1.11 SBAR4BASE: Secondary BAR 4/5 Base Address

This register is mirrored from the BAR 4/5 register pair in the Configuration Space of the Secondary side of the NTB. The register is used by the processor on the primary side of the NTB to examine and load the BAR 4/5 register pair on the Secondary side of the NTB.

Register:SBAR4BASE
Bar:PB01BASE, SB01BASE
Offset:50h

Bit	Attr	Default	Description
63:nn	RWL	00h	**Secondary BAR 4/5 Base** This register is reflected into the BAR 4/5 register pair in the Configuration Space of the Secondary side of the NTB. ***Note:*** These bits will appear to SW as RW.
(nn-1) : 12	RWL	00h	**Reserved** Reserved bits dictated by the size of the memory claimed by the BAR. ***Note:*** These bits will appear to SW as RO.
11:04	RO	00h	**Reserved** Granularity must be at least 4 KB.
03	RO	1b	**Prefetchable** BAR points to Prefetchable memory.
02:01	RO	10b	**Type** Memory type claimed by BAR 4/5 is 64-bit addressable.
00	RO	0b	**Memory Space Indicator** BAR resource is memory (as opposed to I/O).

3.21.1.12 NTBCNTL: NTB Control

This register contains Control bits for the Non-transparent Bridge device.

Register:NTBCNTL
Bar:PB01BASE, SB01BASE
Offset:58h

Bit	Attr	Default	Description
31:11	RO	00h	**Reserved**
10	RW	0b	**Crosslink SBDF Disable Increment** This bit is only valid in NTB/NTB mode This bit determines if SBDF value on the DSD is incremented or not. When = 0 the DSD will increment SBDF +1 When = 1 the DSD will leave the SBDF
09:08	RW	00b	**BAR 4/5 Primary to Secondary Snoop Override Control** This bit controls the ability to force all transactions within the Primary BAR 4/5 window going from the Primary side to the Secondary side to be snoop/no-snoop independent of the ATTR field in the TLP header. 00 - All TLP sent as defined by the ATTR field 01 - Force Snoop on all TLPs: ATTR field overridden to set the" No Snoop" bit = 0 independent of the setting of the ATTR field of the received TLP. 10 - Force No-Snoop on all TLPs: ATTR field overridden to set the "No Snoop" bit = 1 independent of the setting of the ATTR field of the received TLP. 11 - Reserved
07:06	Bar: Attr PB01BASE: RW else: RO	00b	**BAR 4/5 Secondary to Primary Snoop Override Control** This bit controls the ability to force all transactions within the Secondary BAR 4/5 window going from the Secondary side to the Primary side to be snoop/no-snoop independent of the ATTR field in the TLP header. 00 - All TLP sent as defined by the ATTR field 01 - Force Snoop on all TLPs: ATTR field overridden to set the" No Snoop" bit = 0 independent of the setting of the ATTR field of the received TLP. 10 - Force No-Snoop on all TLPs: ATTR field overridden to set the "No Snoop" bit = 1 independent of the setting of the ATTR field of the received TLP. 11 - Reserved
05:04	RW	00b	**BAR 2/3 Primary to Secondary Snoop Override Control** This bit controls the ability to force all transactions within the Primary BAR 2/3 window going from the Primary side to the Secondary side to be snoop/no-snoop independent of the ATTR field in the TLP header. 00 - All TLP sent as defined by the ATTR field 01 - Force Snoop on all TLPs: ATTR field overridden to set the" No Snoop" bit = 0 independent of the setting of the ATTR field of the received TLP. 10 - Force No-Snoop on all TLPs: ATTR field overridden to set the "No Snoop" bit = 1 independent of the setting of the ATTR field of the received TLP. 11 - Reserved

03:02	Bar: Attr PB01BASE: RW else: RO	00b	**BAR 2/3 Secondary to Primary Snoop Override Control** This bit controls the ability to force all transactions within the Secondary BAR 2/3 window going from the Secondary side to the Primary side to be snoop/no-snoop independent of the ATTR field in the TLP header. 00 - All TLP sent as defined by the ATTR field 01 - Force Snoop on all TLPs: ATTR field overridden to set the" No Snoop" bit = 0 independent of the setting of the ATTR field of the received TLP. 10 - Force No-Snoop on all TLPs: ATTR field overridden to set the "No Snoop" bit = 1 independent of the setting of the ATTR field of the received TLP. 11 - Reserved
01	Bar: Attr PB01BASE: RW else: RO	1b	**Secondary Link Disable Control** This bit controls the ability to train the link on the secondary side of the NTB. This bit is used to make sure the primary side is up and operational before allowing transactions from the secondary side. 0 - Link enabled 1 - Link disabled ***Note:*** This bit logically or'd with the LNKCON bit 4
00	Bar: Attr PB01BASE: RW else: RO	1b	**Secondary Configuration Space Lockout Control** This bit controls the ability to modify the Secondary side NTB configuration registers from the Secondary side link partner. ***Note:*** This does not block MMIO space. 0 - Secondary side can read and write secondary registers 1 - Secondary side modifications locked out but reads are accepted

3.21.1.13 SBDF: Secondary Bus, Device and Function

This register contains the Bus, Device and Function for the secondary side of the NTB when PPD.Port Definition is configured as NTB/NTB Section 3.19.3.23, “PPD: PCIE Port Definition” .

Note: The region between the two NTBs is in no mans land and does not matter what value BDF is set to, but the same value must be programmed in both NTBs on each side of the link. The default values have been set to unique bus values midway in the bus region to simplify validation. The SBDF has been made programmable in case end user wishes to move the SBDF for specific validation needs.

Note: This register is only valid when configured as NTB/NTB. This register has no meaning when configured as NTB/RP or RP.

Register:SBDF
Bar:PB01BASE, SB01BASE
Offset:5Ch

Bit	Attr	Default	Description
15:8	RW	7Fh	**Secondary Bus** Value to be used for the Bus number for ID-based routing. Hardware will leave the default value of 7Fh when this port is USD Hardware will increment the default value to 80h when this port is DSD
7:3	RW	00000b	**Secondary Device** Value to be used for the Device number for ID-based routing.
2:0	RW	000b	**Secondary Function** Value to be used for the Function number for ID-based routing.

3.21.1.14 CBDF: Captured Bus, Device and Function

This register contains the Bus, Device and Function for the secondary side of the NTB when PPD.Port Definition is configured as NTB/RP Section 3.19.3.23, “PPD: PCIE Port Definition” .

Note: When configured as a NTB/RP, the NTB must capture the Bus and Device Numbers supplied with all Type 0 Configuration Write Requests completed by the NTB and supply these numbers in the Bus and Device Number fields of the Requester ID for all Requests initiated by the NTB. The Bus Number and Device Number may be changed at run time, and so it is necessary to re-capture this information with each and every Configuration Write Request.

Note: When configured as a NTB/RP, if NTB must generate a Completion prior to the initial device Configuration Write Request, 0’s must be entered into the Bus Number and Device Number fields

Note: This register is only valid when configured as NTB/RP. This register has no meaning when configured as NTB/NTB or RP.

Register:CBDF Bar:PB01BASE, SB01BASE Offset:5Eh			
Bit	**Attr**	**Default**	**Description**
15:8	RO	00h	**Secondary Bus** Value to be used for the Bus number for ID-based routing.
7:3	RO	00000b	**Secondary Device** Value to be used for the Device number for ID-based routing.
2:0	RO	000b	**Secondary Function** Value to be used for the Function number for ID-based routing.

3.21.1.15 PDOORBELL: Primary Doorbell

This register contains the bits used to generate interrupts to the processor on the Primary side of the NTB.

Register:PDOORBELL Bar:PB01BASE, SB01BASE Offset:60h			
Bit	**Attr**	**Default**	**Description**
15	Bar: Attr PB01BASE: RW1C else: RO	0b	**Link State Interrupt** This bit is set when a link state change occurs on the Secondary side of the NTB (Bit 13 of the LNKSTS: PCI Express Link Status Register). This bit is cleared by writing a 1 from the Primary side of the NTB.
14	Bar: Attr PB01BASE: RW1C else: RW1S	0b	**WC_FLUSH_ACK** This bit only has meaning when in NTB/NTB configuration. This bit is set by hardware when a write cache flush was completed on the remote system. This bit is cleared by writing a 1 from the Primary side of the NTB.
13:0	Bar: Attr PB01BASE: RW1C else: RW1S	00h	**Primary Doorbell Interrupts** These bits are written by the processor on the Secondary side of the NTB to cause a doorbell interrupt to be generated to the processor on the Primary side of the NTB if the associated mask bit in the PDBMSK register is not set. A 1 is written to this register from the Secondary side of the NTB to set the bit, and to clear the bit a 1 is written from the Primary side of the NTB. ***Note:*** If both INTx and MSI (NTB PCI CMD bit 10 and NTB MSI Capability bit 0) interrupt mechanisms are disabled software must poll for status since no interrupts of either type are generated.

3.21.1.16 PDBMSK: Primary Doorbell Mask

This register is used to mask the generation of interrupts to the Primary side of the NTB.

Register:PDBMSK
Bar:PB01BASE, SB01BASE
Offset:62h

Bit	Attr	Default	Description
15:0	Bar: Attr PB01BASE: RW else: RO	FFFFh	**Primary Doorbell Mask** This register will allow software to mask the generation of interrupts to the processor on the Primary side of the NTB. 0 - Allow the interrupt 1 - Mask the interrupt

3.21.1.17 SDOORBELL: Secondary Doorbell

This register is valid when in NTB/RP configuration. This register contains the bits used to generate interrupts to the processor on the Secondary side of the NTB.

Register:SDOORBELL
Bar:PB01BASE, SB01BASE
Offset:64h

Bit	Attr	Default	Description
15:0	Bar: Attr PB01BASE: RW1S else: RW1C	00h	**Secondary Doorbell Interrupts** These bits are written by the processor on the Primary side of the NTB to cause an interrupt to be generated to the processor on the Secondary side of the NTB if the associated mask bit in the SDBMSK register is not set. A 1 is written to this register from the Primary side of the NTB to set the bit, and to clear the bit a 1 is written from the Secondary side of the NTB. ***Note:*** If both INTx and MSI (NTB PCI CMD bit 10 and NTB MSI Capability bit 0) interrupt mechanisms are disabled software must poll for status since no interrupts of either type are generated.

3.21.1.18 SDBMSK: Secondary Doorbell Mask

This register is valid when in NTB/RP configuration. This register is used to mask the generation of interrupts to the Secondary side of the NTB.

Register:SDBMSK
Bar:PB01BASE, SB01BASE
Offset:66h

Bit	Attr	Default	Description
15:0	RW	FFFFh	**Secondary Doorbell Mask** This register will allow software to mask the generation of interrupts to the processor on the Secondary side of the NTB. 0 - Allow the interrupt 1 - Mask the interrupt

3.21.1.19 USMEMMISS: Upstream Memory Miss

This register is used to keep a rolling count of misses to the memory windows on the upstream port on the secondary side of the NTB. This a rollover counter. This counter can be used as an aid in determining if there are any programming errors in mapping the memory windows in the NTB/NTB configuration.

Register:USMEMMISS Bar:PB01BASE, SB01BASE Offset:70h			
Bit	**Attr**	**Default**	**Description**
15:0	RW	00h	**Upstream Memory Miss** This register keeps a running count of misses to any of the 3 upstream memory windows on the secondary side of the NTB. The counter does not freeze at max count, it rolls over.

3.21.1.20 SPAD[0 - 15]: Scratchpad Registers 0 - 15

This set of 16 registers, SPAD0 through SPAD15, are shared to both sides of the NTB. They are used to pass information across the bridge.

Register:SPADn Bar:PB01BASE, SB01BASE Offset:80h, 84h, 88h,8Ch, 90h, 94h, 98h, 9Ch, A0h, A4h, A8h, ACh, B0h, B4h, B8h, BCh			
Bit	**Attr**	**Default**	**Description**
31:00	RW	00h	**Scratchpad Register n** This set of 16 registers is RW from both sides of the bridge. Synchronization is provided with a hardware semaphore (SPADSEMA4). Software will use these registers to pass a protocol, such as a heartbeat, from system to system across the NTB.

3.21.1.21 SPADSEMA4: Scratchpad Semaphore

This register will allow software to share the Scratchpad registers.

Register:SPADSEMA4
Bar:PB01BASE, SB01BASE
Offset:C0h

Bit	Attr	Default	Description
31:01	RO	00h	**Reserved**
00	R0TS W1TC	0b	**Scratchpad Semaphore** This bit will allow software to synchronize write ownership of the scratchpad register set. The processor will read the register. If the returned value is 0, the bit is set by hardware to 1 and the reading processor is granted ownership of the scratchpad registers. If the returned value is 1, then the processor on the opposite side of the NTB already owns the scratchpad registers and the reading processor is not allowed to modify the scratchpad registers. To relinquish ownership, the owning processor writes a 1 to this register to reset the value to 0. Ownership of the scratchpad registers is not set in hardware, i.e. the processor on each side of the NTB is still capable of writing the registers regardless of the state of this bit. ***Note:*** For A0 stepping a value of FFFFH must be written to this register to clear the semaphore. For B0 stepping only bit 0 needs to be written to 1 in order to clear the semaphore

3.21.1.22 RSDBMSIXV70: Route Secondary Doorbell MSI-X Vector 7 to 0

This register is used to allow flexibility in the SDOORBELL Section 3.21.1.17, "SDOORBELL: Secondary Doorbell" bits 7 to 0 assignments to one of 4 MSI-X vectors.

Register:RSDBMSIXV70
Bar:PB01BASE
Offset:D0h

Bit	Attr	Default	Description
31:30	RO	0h	**Reserved**
29:28	RW	1h	MSI-X Vector assignment for SDOORBELL bit 7
27:26	RO	0h	**Reserved**
25:24	RW	1h	MSI-X Vector assignment for SDOORBELL bit 6
23:22	RO	0h	**Reserved**
21:20	RW	1h	MSI-X Vector assignment for SDOORBELL bit 5
19:18	RO	0h	**Reserved**
17:16	RW	0h	MSI-X Vector assignment for SDOORBELL bit 4
15:14	RO	0h	**Reserved**
13:12	RW	0h	MSI-X Vector assignment for SDOORBELL bit 3
11:10	RO	0h	**Reserved**
09:08	RW	0h	MSI-X Vector assignment for SDOORBELL bit 2
07:06	RO	0h	**Reserved**
05:04	RW	0h	MSI-X Vector assignment for SDOORBELL bit 1
03:02	RO	0h	**Reserved**
01:00	RW	0h	MSI-X Vector assignment for SDOORBELL bit 0 11 = MSI-X vector allocation 3 10 = MSI-X vector allocation 2 01 = MSI-X vector allocation 1 00 = MSI-X vector allocation 0

3.21.1.23 RSDBMSIXV158: Route Secondary Doorbell MSI-X Vector 15 to 8

This register is used to allow flexibility in the SDOORBELL Section 3.21.1.17, "SDOORBELL: Secondary Doorbell" bits 15 to 8 assignments to one of 4 MSI-X vectors.

Register:RSDBMSIXV158
Bar:PB01BASE
Offset:D4h

Bit	Attr	Default	Description
31:30	RO	0h	**Reserved**
29:28	RW	3h	MSI-X Vector assignment for SDOORBELL bit 15
27:26	RO	0h	**Reserved**
25:24	RW	2h	MSI-X Vector assignment for SDOORBELL bit 14
23:22	RO	0h	**Reserved**
21:20	RW	2h	MSI-X Vector assignment for SDOORBELL bit 13
19:18	RO	0h	**Reserved**
17:16	RW	2h	MSI-X Vector assignment for SDOORBELL bit 12
15:14	RO	0h	**Reserved**
13:12	RW	2h	MSI-X Vector assignment for SDOORBELL bit 11
11:10	RO	0h	**Reserved**
09:08	RW	2h	MSI-X Vector assignment for SDOORBELL bit 10
07:06	RO	0h	**Reserved**
05:04	RW	1h	MSI-X Vector assignment for SDOORBELL bit 9
03:02	RO	0h	**Reserved**
01:00	RW	1h	MSI-X Vector assignment for SDOORBELL bit 8 11 = MSI-X vector allocation 3 10 = MSI-X vector allocation 2 01 = MSI-X vector allocation 1 00 = MSI-X vector allocation 0

3.21.1.24 WCCNTRL: Write Cache Control Register

This register is used for IIO write cache controlability

Register:WCCNTRL Bar:PB01BASE, SB01BASE Offset:E0h			
Bit	**Attr**	**Default**	**Description**
31:01	RO	0h	**Reserved**
00	RW1S	0b	**WCFLUSH** When set forces snap shot flush of the IIO write cache. This register can be set either by host write or inbound MMIO write. ***Note:*** This bit is cleared by hardware upon completion of write cache flush. Software cannot clear this register. 1 = Force snap shot flush of entire IIO write cache 0 = No flush requested or flush operation complete Usage model for this register is such that only a single flush can be issued at a time until acknowledge of completion is received. Writing bit to 1 while it is already set will not cause an additional flush. Flush will only occur on transition from 0 to 1. See Section 26.7.4.1, "ADR Write Cache (WC) flush acknowledge example using NTB/NTB" for details on how to utilize this register.

3.21.1.25 B2BSPAD[0 - 15]: Back-to-back Scratchpad Registers 0 - 15

These registers are valid when in NTB/NTB configuration. This set of 16 registers, B2BSPAD0 through B2BSPAD15, is used by the processor on the Primary side of the NTB to generate accesses to the Scratchpad registers on a second NTB whose Secondary side is connected to the Secondary side of this NTB. Writing to these registers will cause the NTB to generate a PCIe packet that is sent to the connected NTB's Scratchpad registers. This mechanism allows inter-system communication through the pair of NTBs. The B2BBAR0XLAT register must be properly configured to point to BAR 0/1 on the opposite NTB for this mechanism to function properly. This mechanism doesn't require a semaphore because each NTB has a set of Scratchpad registers. The system passing information will always write to the registers on the opposite NTB, and read its own Scratchpad registers to get information from the opposite system.

Register:B2BSPADn Bar:PB01BASE, SB01BASE Offset:100h, 104h, 108h, 10Ch, 110h, 114h, 118h, 11Ch, 120h, 124h, 128h, 12Ch, 130h, 134h, 138h, 13Ch			
Bit	**Attr**	**Default**	**Description**
31:0	Bar: Attr PB01BASE: RW else: RO	00h	**Back-to-back Scratchpad Register n** This set of 16 registers is written only from the Primary side of the NTB. A write to any of these registers will cause the NTB to generate a PCIe packet which is sent across the link to the opposite NTB's corresponding Scratchpad register.

3.21.1.26 B2BDOORBELL: Back-to-Back Doorbell

This register is valid when in NTB/NTB configuration. This register is used by the processor on the primary side of the NTB to generate accesses to the PDOORBELL register on a second NTB whose Secondary side is connected to the Secondary side of this NTB. Writing to this register will cause the NTB to generate a PCIe packet that is sent to the connected NTB's PDOORBELL register, causing an interrupt to be sent to the processor on the second system. This mechanism allows inter-system communication through the pair of NTBs. The B2BBAR0XLAT register must be properly configured to point to BAR 0/1 on the opposite NTB for this mechanism to function properly.

Register:B2BDOORBELL
Bar:PB01BASE, SB01BASE
Offset:140h

Bit	Attr	Default	Description
15	RV	0b	**Reserved**
14	RV	0b	**WC_FLUSH_DONE** '1' = This bit will be set by hardware when the IIO write cache has been flushed. '0' = Hardware upon sensing that bit is set to '1' will schedule a PMW to set the corresponding bit in the remote NTB (PDOORBELL, bit 14 = '1'). Hardware will then clears this bit after scheduling the PMW. ***Note:*** SW cannot read this register, reads will always return 0
13:00	Bar: Attr PB01BASE: RW1S else: RO	00h	**B2B Doorbell Interrupt** These bits are written by the processor on the Primary side of the NTB. Writing to this register will cause a PCIe packet with the same contents as the write to be sent to the PDOORBELL register on the a second NTB connected back-to-back with this NTB, which in turn will cause a doorbell interrupt to be generated to the processor on the second NTB. Hardware on the originating NTB clears this register upon scheduling the PCIE packet.

3.21.1.27 B2BBAR0XLAT: Back-to-Back BAR 0/1 Translate

This register is valid when in NTB/NTB configuration. This register is used to set the base address where the back-to-back doorbell and scratchpad packets will be sent. This register must match the base address loaded into the BAR 0/1 pair on the opposite NTB, whose Secondary side in linked to the Secondary side of this NTB.

Note: There is no hardware enforced limit for this register, care must be taken when setting this register to stay within the addressable range of the attached system.

Register:B2BBAR0XLAT
Bar:PB01BASE, SB01BASE
Offset:144h

Bit	Attr	Default	Description
63:15	Bar: Attr PB01BASE: RW else: RO	0000h	**B2B translate** Base address of Secondary BAR 0/1 on the opposite NTB
14:00	RO	00h	**Reserved** Limit register has a granularity of 32 KB (2^{15})

3.21.2 MSI-X MMIO Registers (NTB Primary side)

Primary side MSI-X MMIO registers reached via PB01BASE

Table 95. NTB MMIO Map

PMSIXTLB0				2000h	PMSIXPBA				3000h
				2004h					3004h
PMSIXDATA0				2008h					3008h
PMSIXVECCNTL0				200Ch					300Ch
PMSIXTLB1				2010h					3010h
				2014h					3014h
PMSIXDATA1				2018h					3018h
PMSIXVECCNTL1				201Ch					301Ch
PMSIXTLB2				2020h					3020h
				2024h					3024h
PMSIXDATA2				2028h					3028h
PMSIXVECCNTL2				202Ch					302Ch
PMSIXTLB3				2030h					3030h
				2034h					3034h
PMSIXDATA3				2038h					3038h
PMSIXVECCNTL3				203Ch					303Ch
				2040h					3040h
				2044h					3044h
				2048h					3048h
				204Ch					304Ch

3.21.2.1 PMSIXTBL[0-3]: Primary MSI-X Table Address Register 0 - 3

Register:PMSIXTBLn Bar:PB01BASE, SB01BASE Offset:00002000h, 00002010h, 00002020h, 00002030h			
Bit	**Attr**	**Default**	**Description**
63:32	RW	00000000h	**MSI-X Upper Address** Upper address bits used when generating an MSI-X.
31:02	RW	00000000h	**MSI-X Address** System-specified message lower address. For MSI-X messages, the contents of this field from an MSI-X Table entry specifies the lower portion of the DWORD-aligned address (AD[31:02]) for the memory write transaction.
01:00	RO	00b	**MSG_ADD10** For proper DWORD alignment, these bits need to be 0's.

3.21.2.2 PMSIXDATA[0-3]: Primary MSI-X Message Data Register 0 - 3

Register:PMSIXDATAn Bar:PB01BASE, SB01BASE Offset:00002008h, 00002018h, 00002028h, 00002038h			
Bit	**Attr**	**Default**	**Description**
31:00	RW	0000h	**Message Data** System-specified message data.

Table 96. MSI-X Vector Handling and Processing by IIO on Primary Side

Number of Messages enabled by Software	Events	IV[7:0]
1	All	xxxxxxxx[1]
4	PD[04:00]	xxxxxxxx
	PD[09:05]	xxxxxxxx
	PD[14:10]	xxxxxxxx
	HP, BW-change, AER, PD[15]	xxxxxxxx

1. The term "xxxxxx" in the Interrupt vector denotes that software initializes them and IIO will not modify any of the "x" bits

3.21.2.3 PMSIXVECCNTL[0-3]: Primary MSI-X Vector Control Register 0 - 3

Register:PMSIXVECCNTLn Bar:PB01BASE, SB01BASE Offset:0000200Ch, 0000201Ch, 0000202Ch, 0000203Ch			
Bit	**Attr**	**Default**	**Description**
31:01	RO	00000000h	Reserved
00	RW	1b	**MSI-X Mask** When this bit is set, the NTB is prohibited from sending a message using this MSI-X Table entry. However, any other MSI-X Table entries programmed with the same vector will still be capable of sending an equivalent message unless they are also masked.

3.21.2.4 PMSIXPBA: Primary MSI-X Pending Bit Array Register

Register:PMSIXPBA
Bar:PB01BASE, SB01BASE
Offset:00003000h

Bit	Attr	Default	Description
31:04	RO	0000h	Reserved
03	RO	0b	MSI-X Table Entry 03 (NTB) has a Pending Message.
02	RO	0b	MSI-X Table Entry 02 (NTB) has a Pending Message.
01	RO	0b	MSI-X Table Entry 01 (NTB) has a Pending Message.
00	RO	0b	MSI-X Table Entry 00 (NTB) has a Pending Message.

3.21.3 MSI-X MMIO registers (NTB Secondary Side)

Secondary side MSI-X MMIO registers reached via PB01BASE (debug) and SB01BASE.

These registers are valid when in NTB/RP configuration.

Table 97. NTB MMIO Map

SMSIXTLB0				4000h	SMSIXPBA				5000h
				4004h					5004h
SMSIXDATA0				4008h					5008h
SMSIXVECCNTL0				400Ch					500Ch
SMSIXTLB1				4010h					5010h
				4014h					5014h
SMSIXDATA1				4018h					5018h
SMSIXVECCNTL1				401Ch					501Ch
SMSIXTLB2				4020h					5020h
				4024h					5024h
SMSIXDATA2				4028h					5028h
SMSIXVECCNTL2				402Ch					502Ch
SMSIXTLB3				4030h					5030h
				4034h					5034h
SMSIXDATA3				4038h					5038h
SMSIXVECCNTL3				403Ch					503Ch
				4040h					5040h
				4044h					5044h
				4048h					5048h
				404Ch					504Ch

3.21.3.1 SMSIXTBL[0-3]: Secondary MSI-X Table Address Register 0 - 3

Register:SMSIXTBLn Bar:PB01BASE, SB01BASE Offset:00004000h, 00004010h, 00004020h, 00004030h			
Bit	**Attr**	**Default**	**Description**
63:32	RW	00000000h	**MSI-X Upper Address** Upper address bits used when generating an MSI-X.
31:02	RW	00000000h	**MSI-X Address** System-specified message lower address. For MSI-X messages, the contents of this field from an MSI-X Table entry specifies the lower portion of the DWORD-aligned address (AD[31:02]) for the memory write transaction.
01:00	RO	00b	**MSG_ADD10** For proper DWORD alignment, these bits need to be 0's.

3.21.3.2 SMSIXDATA[0-3]: Secondary MSI-X Message Data Register 0 - 3

SDOORBELL bits to MSI-X mapping can be reprogrammed through Section 3.21.1.22, "RSDBMSIXV70: Route Secondary Doorbell MSI-X Vector 7 to 0" and Section 3.21.1.23, "RSDBMSIXV158: Route Secondary Doorbell MSI-X Vector 15 to 8"

Register:SMSIXDATAn Bar:PB01BASE, SB01BASE Offset:00004008h, 00004018h, 00004028h, 00004038h			
Bit	**Attr**	**Default**	**Description**
31:00	RW	0000h	**Message Data** System-specified message data.

Table 98. MSI-X Vector Handling and Processing by IIO on Secondary Side

Number of Messages Enabled by Software	Events	IV[7:0]
1	All	xxxxxxxx[1]
4	PD[04:00]	xxxxxxxx
	PD[09:05]	xxxxxxxx
	PD[14:10]	xxxxxxxx
	PD[15]	xxxxxxxx

1. The term "xxxxxx" in the Interrupt vector denotes that software initializes them and IIO will not modify any of the "x" bits

3.21.3.3 SMSIXVECCNTL[0-3]: Secondary MSI-X Vector Control Register 0 - 3

Register:SMSIXVECCNTLn Bar:PB01BASE, SB01BASE Offset:0000400Ch, 0000401Ch, 0000402Ch, 0000403Ch			
Bit	**Attr**	**Default**	**Description**
31:01	RO	00000000h	Reserved
00	RW	1b	**MSI-X Mask:** When this bit is set, the NTB is prohibited from sending a message using this MSI-X Table entry. However, any other MSI-X Table entries programmed with the same vector will still be capable of sending an equivalent message unless they are also masked.

3.21.3.4 SMSIXPBA: Secondary MSI-X Pending Bit Array Register

Register:SMSIXPBA Bar:PB01BASE, SB01BASE Offset:00005000h			
Bit	**Attr**	**Default**	**Description**
31:04	RO	0000h	Reserved
03	RO	0b	MSI-X Table Entry 03 (NTB) has a Pending Message.
02	RO	0b	MSI-X Table Entry 02 (NTB) has a Pending Message.
01	RO	0b	MSI-X Table Entry 01 (NTB) has a Pending Message.
00	RO	0b	MSI-X Table Entry 00 (NTB) has a Pending Message.

§ §

4.0 Technologies

4.1 Intel® Virtualization Technology (Intel® VT)

Intel® VT comprises technology components to support virtualization of platforms based on Intel architecture microprocessors and chipsets. Intel® Virtualization Technology (Intel® VT-x) added hardware support in the processor to improve the virtualization performance and robustness. Intel® Virtualization Technology for Directed I/O (Intel® VT-d) adds chipset hardware implementation to support and improve I/O virtualization performance and robustness.

Intel® VT-x specifications and functional descriptions are included in the *Intel® 64 and IA-32 Architectures Software Developer's Manual, Volume 3B* and is available at http://www.intel.com/products/processor/manuals/index.htm.

The Intel® VT-d 2.0 spec and other VT documents are located at http://www.intel.com/technology/platform-technology/virtualization/index.htm.

4.1.1 Intel® VT-x Objectives

Intel® VT-x provides hardware acceleration for the virtualization of IA platforms. Virtual Machine Monitor (VMM) can use Intel® VT-x features to provide improved reliable virtualized platform. By using Intel® VT-x, a VMM is:

- **Robust:** VMMs no longer need to use paravirtualization or binary translation. This means that they will be able to run off-the-shelf OSs and applications without any special steps.
- **Enhanced:** Intel® VT enables VMMs to run 64-bit guest OSs on IA x86 processors.
- **More reliable:** With the available hardware support, VMMs can now be smaller, less complex, and more efficient. This improves reliability and availability, and reduces the potential for software conflicts.
- **More secure:** The use of hardware transitions in the VMM strengthens the isolation of VMs and further prevents corruption of one VM from affecting others on the same system.

4.1.2 Intel® VT-x Features

The processor core supports the following Intel® VT-x features:

- Extended Page Tables (EPT)
  - Hardware-assisted page table virtualization.
  - Eliminates VM exits from guest OS to the VMM for shadow page-table maintenance.
- Virtual Processor IDs (VPID)
  - Ability to assign a VM ID to tag processor core hardware structures (e.g. TLBs).
  - Avoids flushes on VM transitions to give a lower-cost VM transition time and an overall reduction in virtualization overhead.
- Guest Preemption Timer
  - Mechanism for a VMM to preempt the execution of a guest OS after an amount of time specified by the VMM. The VMM sets a timer value before entering a guest.
  - Aids VMM developers in flexibility and Quality of Service (QoS) guarantees.
- Descriptor-Table Exiting
  - Descriptor-table exiting allows a VMM to protect a guest OS from internal (malicious software-based) attack by preventing the relocation of key system data structures like interrupt descriptor table (IDT), global descriptor table (GDT), local descriptor table (LDT), and task segment selector (TSS).
  - A VMM using this feature can intercept (by a VM exit) attempts to relocate these data structures and prevent them from being tampered by malicious software.

4.1.3 Intel® VT-d Objectives

The key Intel® VT-d objectives are domain-based isolation and hardware-based virtualization. A domain can be abstractly defined as an isolated environment in a platform to which a subset of host physical memory is allocated. Virtualization allows for the creation of one or more partitions on a single system. This could be multiple partitions in the same OS or multiple operating system instances running on the same system, offering benefits such as system consolidation, legacy migration, activity partitioning, or security.

4.1.4 Intel® VT-d Features

The processor supports the following Intel® VT-d features:

- The Intel® Xeon® processor C5500/C3500 series also supports Intel® VT-d2, which is a superset of VT-d that provides improved performance.
- Root entry, context entry, and default context
- 48-bit max guest address width and 40-bit max host address width
- Support for 4 K page sizes only
- Support for register-based fault recording only (for single entry only) and support for MSI interrupts for faults
  - — Support for fault collapsing based on Requester ID
- 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 interrupt remapping
- Support for queue-based invalidation interface
- Support for Intel® VT-d read prefetching/snarfing e.g. translations within a cacheline are stored in an internal buffer for reuse for subsequent transactions

4.1.5 Intel® VT-d Features Not Supported

The following features are not supported by the processor with Intel® VT-d:

- No support for PCISIG endpoint caching (ATS)
- No support for advance fault reporting
- No support for super pages
- One or two level page walks are not supported for non-isoch VT-d DMA 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.

4.2 Intel® I/O Acceleration Technology (Intel® IOAT)

Intel® I/O Acceleration Technology includes optimizations of the SW Protocol Stack (a refined TCP/IP stack lowers CPU load), Packet Header Splitting, Direct Cache Access, Interrupt Modulation (several interrupts are collected and sent to the processor with concatenated packets), Asynchronous Low Cost Copy (ALCC, HW is added so the CPU issues a memory/memory copy command, vs read/write), Lightweight Threading (each new packet is handled by a new thread, one level of protocol stack optimization), DMA enhancements, and PCIe enhancement technologies.

Support of Intel® IOAT implies complete support for IOAT HW features as well as various SW application and driver components.

The Intel® Xeon® processor C5500/C3500 series does not fully support Intel® IOAT, but it supports a subset of Intel® IOAT, as described below.

4.2.1 Intel® QuickData Technology

Intel® QuickData Technology makes Intel® chipsets excel with Intel network controllers. The Intel® Xeon® processor C5500/C3500 series uses the third generation of the Intel® QuickData Technology.

The Intel® Xeon® processor C5500/C3500 series supports Intel® QuickData Technology. A NIC that is Intel® QuickData Technology capable can be plugged into any of the processor PCIe* ports, or be plugged into a PCIe ports below the PCH, and use the Intel® QuickData Technology capabilities.

4.2.1.1 Port/Stream Priority

The Intel® Xeon® processor C5500/C3500 series does not support port or stream priority.

4.2.1.2 Write Combining

The Intel® Xeon® processor C5500/C3500 series does not support the Intel® QuickData Technology write combining feature.

4.2.1.3 Marker Skipping

The DMA engine can copy a block of data from a source buffer to a destination buffer, and can be programmed to skip bytes (marker) in the source buffer, eliminating their position in the destination buffer (i.e. destination data is packed).

4.2.1.4 Buffer Hint

A bit in the Descriptor Control Register, which when set, provides guidance to the HW that some or all of the data processed by the descriptor may be referenced again in a subsequent descriptor. Software sets this bit if the source data will most likely be specified in another descriptor of this bundle. “Bundle” indicates descriptors that are in a group. When the Bundle bit is set in the Descriptor Control Register, the descriptor is associated with the next descriptor, creating a descriptor bundle. Thus each descriptor in the bundle has “Bundle=1” except for the last one, which has Bundle=0.

4.2.1.5 DCA

The Intel® Xeon® processor C5500/C3500 series supports DCA from both the DMA engine (on both payload and completion writes) and the PCIe ports.

4.2.1.6 DMA

The Intel® Xeon® processor C5500/C3500 series incorporates a high performance DMA engine optimized primarily for moving data between memory. The DMA also supports moving data between memory and MMIO (push data packet size to PCIe is limited to a maximum size of 64B).

There are eight software-visible Intel® QuickData Technology DMA engines (i.e. eight PCI functions) and each DMA engine has one channel. These channels are concurrent and conform to the Intel® QuickData Technology specification. Each DMA engine can be independently assigned to a VM in a virtualized system.

4.2.1.6.1 Supported Features

The following features are supported by the DMA engine:

- Effective move BW of 2.5 GB/s (2.5 GB/s effective read + 2.5 GB/s effective write), calculated assuming descriptor batch size of 2 and a data payload size of 1460 B.
- Raw BW of 5 GB/s read + 5 GB/s write.
- Eight independent DMA Channels where each channel is compliant with Intel® QuickData Technology versions 3 and 2, but not compatible with version 1.
- Data transfer between two system memory locations, or from system memory to MMIO.
- CRC-32 Generation and Check.
- Flow through CRC.
- Marker Skipping.
- Page Zeroing.
- 40 bits of addressing, though the Intel® QuickData Technology DMA BAR still supports the PCI compliant 64bit BAR. Software is expected to program the BAR to less than 2^40, otherwise an error is generated. Similarly for DMA accesses generated by the DMA controller.
- Maximum transfer length of 1 MB per DMA descriptor block.
- Both coherent and non-coherent memory transfer on a per descriptor basis with independent control of coherency for source and destination.
- Support for relaxed ordering for transactions to main memory.
- Support for deep pipelining in each channel independently; i.e. while a DMA channel is servicing the descriptor/data-payload for one move operation, it pipelines the descriptor and data payload for the next move (if there is one)
- Programmable mechanisms for signaling the completion of a descriptor by generating an MSI-X interrupt or legacy level-sensitive interrupt.
- Programmable mechanism for signaling the completion of a descriptor by performing an outbound write of the completion status.
- Deterministic error handling during transfer by aborting the transfer and also permitting the controlling process to abort the transfer via command register bits.
- MSI-X with 1 vector per function.
- Interrupt coalescing.
- Support for FLR independently for each DMA engine. Allows for individual Intel® QuickData Technology DMA channels to be reset and reassigned across VMs.
- Intel® QuickData Technology DMA transactions are translated via Intel® VT-d.

4.2.1.6.2 Unsupported Features

The following features are not supported by the DMA controller:

- DMA data transfer from I/O subsystem to local system memory, and I/O to I/O subsystems are not supported.
- Backward compatibility to Intel® QuickData Technology Version 1 specifications.
- Hardware model for controlling Intel® QuickData Technology DMA via NIC hardware.
- No support for CB_Query message to unlock DMA.

4.3 Simultaneous Multi Threading (SMT)

The Intel® Xeon® processor C5500/C3500 series supports SMT, which allows a single core to function as two logical processors. While some execution resources such as caches, execution units, and buses are shared, each logical processor has its own architectural state with its own set of general-purpose registers and control registers. This feature must be enabled via the BIOS and requires operating system support.

4.4 Intel® Turbo Boost Technology

Intel® Turbo Boost technology is a feature that allows the processor core to opportunistically and automatically run faster than its rated operating frequency if it is operating below power, temperature, and current limits. The result is increased performance in multi-threaded and single threaded workloads. It must be enabled in the BIOS for the processor to operate within specification.

§ §

5.0 IIO Ordering Model

5.1 Introduction

The IIO spans two different ordering domains: one that adheres to producer-consumer ordering (PCI Express*) and one that is unordered (Intel® QPI). One of the primary functions of the IIO is to ensure that the producer-consumer ordering model is maintained in the unordered, Intel® QPI domain.

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

Table 99. Ordering Term Definitions (Sheet 1 of 2)

Term	Definition
Intel® QPI Ordering Domain	Intel® QPI has a relaxed ordering model allowing reads, writes and completions to flow independent of each other. Intel® QPI implements this through the use of multiple, independent virtual channels. With the exception of the home channel, which maintains ordering to ensure coherency, the Intel®QPI ordering domain is in general 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 (e.g. completions must pass non-posted requests). The set of these rules are described in *PCI Express Base Specification,* Revision 2.0.
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® QPI 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® QPI is largely unordered, all requests are considered to be non-posted until the target responds. Through 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 PCIe device.
Outbound Read Completion	The completion for an outbound read. For example, the read data which results in a CPU read of a PCI Express device. *While the data flows inbound, the completion is still for an outbound read.*
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 PCIe device.
Outbound Write Completion	The completion for an outbound write. For example, the completion from a PCI Express device which results from a CPU-initiated I/O or configuration write. *While the completion flows inbound, the completion is still for an outbound write.*
Inbound Read	A read issued toward an Intel® QPI component. This can be a read issued by a PCI Express device. An obvious example is a PCI Express device reading main memory.

Table 99. Ordering Term Definitions (Sheet 2 of 2)

Term	Definition
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. *While the data flows outbound, the completion is still for an inbound read.*
Inbound Write	A write issued toward an Intel® QPI 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® QPI 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.

5.2 Inbound Ordering Rules

Inbound transactions originate from PCI Express, Intel® QuickData Technology DMA, or DMI and target main memory. In general, the IIO forwards inbound transactions in FIFO order, with specific exceptions. For example, PCI Express requires that read completions are allowed to pass stalled read requests. This forces read completions to bypass any reads that might be back-pressured on Intel® QPI. Sequential, non-posted requests are not required to be completed in the order in which they were requested.[1]

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

The IIO relaxes ordering between different PCI Express ports, aside from the peer-to-peer restrictions below.

5.2.1 Inbound Ordering Requirements

In general, there are no ordering requirements between transactions received on different PCI Express interfaces. 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. See the *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: 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 6: Since requests for ownership do not establish ordering, these requests can be pipelined. Write ordering is established when the line transitions to the “Modified” state.Inbound messages follow the same ordering rules as inbound posted writes (FENCE messages have their own rules).

1. The DMI interface has exceptions to this rule as specified in Section 5.2.1.

RULE 7: If an inbound read completes with multiple sub-completions (e.g. a cacheline 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 (e.g. the No Snoop attribute is set). The IIO will order all transactions regardless of its destination.

RULE 8: For PCI Express ports, different read requests should be completed without any ordering dependency. For the DMI interface, however, all read requests with the same Tag must be completed in the order that the respective requests were issued, but, as a simplification, the IIO will return all completions in original read request order (e.g., independent of whether or not the requests have the same tag).

Different read requests issued on a PCI Express interface should be completed in any order. This attribute yields lower read latency for platforms such as Intel® Xeon® processor C5500/C3500 series in which Intel® QPI is an unordered, multipath interface. However the read completion ordering restriction on DMI implies that the IIO must guarantee stronger ordering on that interface.

5.2.2 Special Ordering Relaxations

The *PCI Express Base Specification,* Revision 2.0 specifies that reads do not have any ordering constraints with other reads. An example of why a read would be blocked is the case of an Intel® QPI address conflict. Under such a blocking condition, subsequent transactions should be allowed to proceed until the blocking condition is cleared.

Implementation note: The IIO does not do any read passing read performance optimizations.

5.2.2.1 Inbound Writes Can Pass Outbound Completions

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.

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

With the exception of peer-to-peer requests, the IIO clears the relaxed ordering for outbound transactions received from the Intel® QPI Ordering Domain. For local and remote peer-to-peer transactions, the attribute is preserved for both requests and completions.

5.2.3 Inbound Ordering Rules Summary

Table 100 indicates an ordering relationship between two inbound transactions as implemented in the IIO and summarizes the inbound ordering rules described in previous sections.

Yes — The second transaction (row) allowed to pass the first (column).

No — The second transaction not be allowed to pass the first transaction. This is may be required to satisfy the Producer-Consumer strong ordering model or may be the implementation choice for the IIO. The first transaction is considered done when it is globally observed.

Relaxed Ordering (RO) Attribute bit set (1b) means that the RO bit is set in the transaction and for VC0, IIOMISCCTRL.18 (Disable inbound RO for VC0 traffic) is clear. Otherwise, relaxed ordering is not enabled.

Table 100. Inbound Data Flow Ordering Rules

Row Pass Column?	Inbound Write or Message Request	Inbound Read Request	Outbound Read Completion	Outbound Configuration or I/O Write Completion
Inbound Write or Message Request	1. No[1] 2. Yes[2]	Yes	Yes	Yes
Inbound Read Request	No	No	Yes	Yes
Outbound Read Completion	No	Yes	1. Yes[3] 2. No[4]	No
Outbound Configuration or I/O Write Completion	No	Yes	No	No

1. A Memory Write or Message Request with the Relaxed Ordering Attribute bit cleared (0b) may **not** pass any other Memory Write or Message Request. If the IIOMISCCTRL.14 (Pipeline NS writes) is set, than the IIO will pipeline writes and will rely on the platform to maintain this strict ordering.
2. A Memory Write or Message Request with the Relaxed Ordering Attribute bit set (1b) may pass any other Memory Write or Message Request.
3. Outbound read completions from PCIe that have different tags may not return in the original request order.
4. Multiple sub-completions of a given outbound read request (i.e., with the same tag) will be returned in address order. All outbound read completions from DMI are returned in the original request order.

5.3 Outbound Ordering Rules

Outbound transactions through the IIO are memory, I/O or configuration read/write transactions originating on an Intel® QPI interface destined for a PCI Express or DMI 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 Intel® QPI, non-coherent writes are not considered complete until the IIO returns a Cmp for the NcWr transaction. On PCI Express and DMI interfaces, memory writes are posted. Therefore, the IIO should return this completion as soon as possible 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 (e.g. I/O and configuration writes), the target device will post the completion.

5.3.1 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. This rule prevents deadlocks by guaranteeing forward progress. Consider the case when the outbound queues are entirely filled with read requests and likewise, the inbound queues are also filled with read requests. The only way to prevent the deadlock is if one of the queues allow completions to flow “around” the stalled read requests.Consider the example in Rule 1. If the reads are enqueued and a write transaction is also behind one or more read requests, the only way for the read completion to proceed is if the prior posted writes are also allowed to proceed.

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. See the *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 IIO receives a request for 4 KB on the PCI Express interface and this request targets the Intel® QPI port (main memory), then the IIO splits up the request into multiple 64 B requests. Since Intel® QPI is an unordered domain, it is possible that the IIO receives the second cache line of data before the first. Under such unordered situations, the IIO 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 IIO, 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. Therefore, the IIO will not respond to the configuration write until after the register is actually written and all expected side-effects have completed.

5.3.2 Outbound Ordering Rules Summary

Table 100 indicates an ordering relationship between two outbound transactions as implemented in the IIO and summarizes the outbound ordering rules described in previous sections.

Yes — The second transaction (row) must be allowed to pass the first (column) to avoid deadlock per the *PCI Express Base Specification,* Revision 2.0 or may be the implementation choice for the IIO (i.e., this entry is Y/N in the *PCI Express Base Specification,* Revision 2.0).

No — The second transaction must not be allowed to pass the first transaction. This is may be required to satisfy the Producer-Consumer strong ordering model or may be the implementation choice for the IIO (i.e., this entry is Y/N in the *PCI Express Base Specification,* Revision 2.0).

Table 101. Outbound Data Flow Ordering Rules

Row Pass Column?	Outbound Write or Message Request	Outbound Read Request	Outbound Configuration Write Request	Inbound Read Completion
Outbound Write or Message Request	No[1]	Yes	Yes	Yes
Outbound Read Request	No	No	No	Yes
Outbound Configuration or I/O Write Request	No	No	No	Yes
Inbound Read Completion	No	Yes	Yes	1. Yes[2] 2. No[3]

1. A Memory Write or Message Request may **not** pass any other Memory Write or Message Request. The IIO does not support setting the Relaxed Ordering Attribute bit for an Outbound Memory Write or Message Request.
2. Inbound read completions from PCIe that have different Tags may not return in the original request order.
3. Multiple sub-completions of a given inbound read request (i.e., with the same Tag) will be returned in address order. All inbound read completions to DMI are returned by the IIO in the original request order.

5.4 Peer-to-Peer Ordering Rules

The IIO supports peer-to-peer read and write transactions. A peer-to-peer transaction is defined as a transaction issued on one PCI Express interface destined for another PCI Express interface (Note: PCI Express to DMI is also supported). All peer-to-peer transactions are treated as non-coherent by the system. There are three types of peer-to-peer transactions supported by the IIO:

Hinted PCI Peer-to-Peer — A transaction initiated on a PCI bus destined for another PCI bus on the *same* I/O device (i.e., not visible to the IIO). For example, a PXH (dual-PCI to PCI Express bridge).

Local Peer-to-Peer — A transaction initiated on a PCI Express port destined for another PCI Express port on the *same* IIO.

Remote Peer-to-Peer — A transaction initiated on a PCI Express port of the local IIO destined for another PCI Express port on the remote IIO connected via an Intel® QPI port.

Local and remote peer-to-peer transactions adhere to the ordering rules listed in Section 5.2.1 and Section 5.3.1.

5.4.1 Hinted Peer-to-Peer

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

5.4.2 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 internal Coherent IIO interface. Local peer-to-peer write transactions complete when the outbound ordering logic for the target PCI Express port receives the transaction and ordering rules are satisfied. Local peer-to-peer read transactions are completed by the target device.

5.4.3 Remote Peer-to-Peer

In the initiating IIO, a remote peer-to-peer transaction follows the same ordering rules as inbound transactions destined to main memory. In the target IIO, a remote peer-to-peer transaction follows the same ordering rules as outbound transactions destined to an I/O device.

RULE 1: Similar to peer to peer write requests, the IIO must serialize remote peer-to-peer read *completions*.

5.5 Interrupt Ordering Rules

IOxAPIC or MSI interrupts are either directed to a single processor or broadcast to multiple processors (see Interrupt chapter for more details). The IIO treats interrupts as posted transactions with exceptions (Section 5.5.1). 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 IIO.

Since interrupts are treated as posted transactions, the ordering rule that read completions push interrupts naturally applies as well. For example:

- An interrupt generated by a PCI Express interface must be strongly ordered with read completions from configuration registers within that same PCI Express root port.
- Read completions from the integrated IOAPIC's registers (configuration and memory-mapped I/O space) must push all interrupts generated by the integrated IOAPIC.
- Read completions from the Intel® VT-d registers must push all interrupts generated by the Intel® VT-d logic (e.g. an error condition).

Similarly, MSIs generated by the IIO internal devices, such as the DMA engine, root ports, and I/OxAPIC, also need to follow the ordering rules of posted writes. For example, an interrupt generated by the DMA engine must be ordered with read completions from the DMA engine registers.

5.5.1 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. Finally, for level-triggered interrupts, the processor sends an End-of-Interrupt (EOI) special cycle to clear the interrupt in the IOxAPIC.

The EOI special cycle is treated as an outbound posted transaction with regard to ordering rules.

5.5.2 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.

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

5.6 Configuration Register Ordering Rules

The IIO implements legacy PCI configuration space registers. Legacy PCI configuration registers are accessed with NcCfgRd and NcCfgWr transactions (using PCI Bus, Device Function) received on the Intel® QPI interface.

For PCI configuration space, the ordering requirements are the same as standard, non-posted configuration cycles on PCI. See Section 5.2.1 and Section 5.3.1 for details. Furthermore, on configuration writes to the IIO the completion is returned by the IIO only after the data is actually written into the register.

5.7 Intel® VT-d Ordering Exceptions

The transaction flow to support the address remapping feature of Intel® VT-d requires that the IIO 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.

§ §

6.0 System Address Map

This chapter provides a basic overview of the system address map and describes how the IIO comprehends and decodes the various regions in the system address map. The term “IIO” in this chapter refers to the integrated IO module of Intel® Xeon® processor C5500/C3500 series. This chapter does not provide the full details of the platform system address space as viewed by software and it also does not provide the details of processor address decoding.

The Intel® Xeon® processor C5500/C3500 series supports 40 bits [39:0] of memory addressing on its Intel® QPI interface. The IIO 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® QPI (which is dependent on the Intel® QPI profile but is always equal to 2^40 for the IIO) are reported as errors by IIO. The IIO as a requester would never generate requests on PCI Express with any of address bits 63 to 40 set. For packets the IIO receives from Intel® QPI and for packets the IIO receives from PCI Express that fall below the top of Intel® QPI physical address space, the upper address bits from top of Intel® QPI physical address space up to bit 63 must be considered as 0s for target address decoding purposes. The IIO always performs full 64-bit target address decoding.

The IIO supports 16 bits of I/O addressing on its Intel® QPI interface. The IIO as a requester would never generate I/O requests on PCI Express with any of address bits 31 to 16 set.

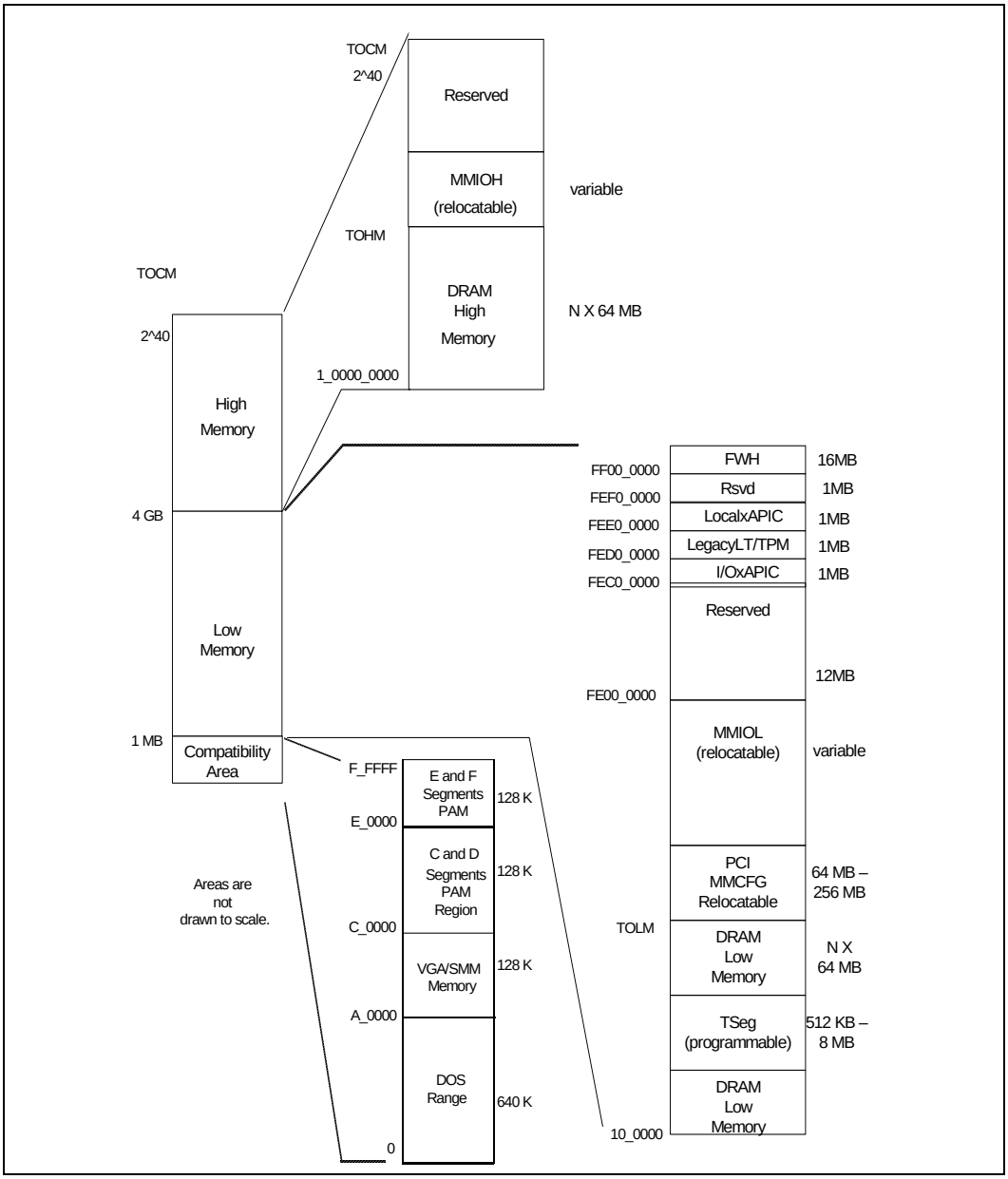
The IIO supports PCI configuration addressing up to 256 buses, 32 devices per bus and eight functions per device. A single grouping of 256 buses, 32 devices per bus and eight functions per device is referred to as a PCI *segment*. The processor source decoder supports multiple PCI segments in the system.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

6.1 Memory Address Space

Figure 63 shows the system memory address space. 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 IIO that is attached to the PCH or provides a path towards the PCH. This port is also the recipient of all addresses that are not positively decoded towards any PCI Express device or towards memory.

Figure 63. System Address Map



6.1.1 System DRAM Memory Regions

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

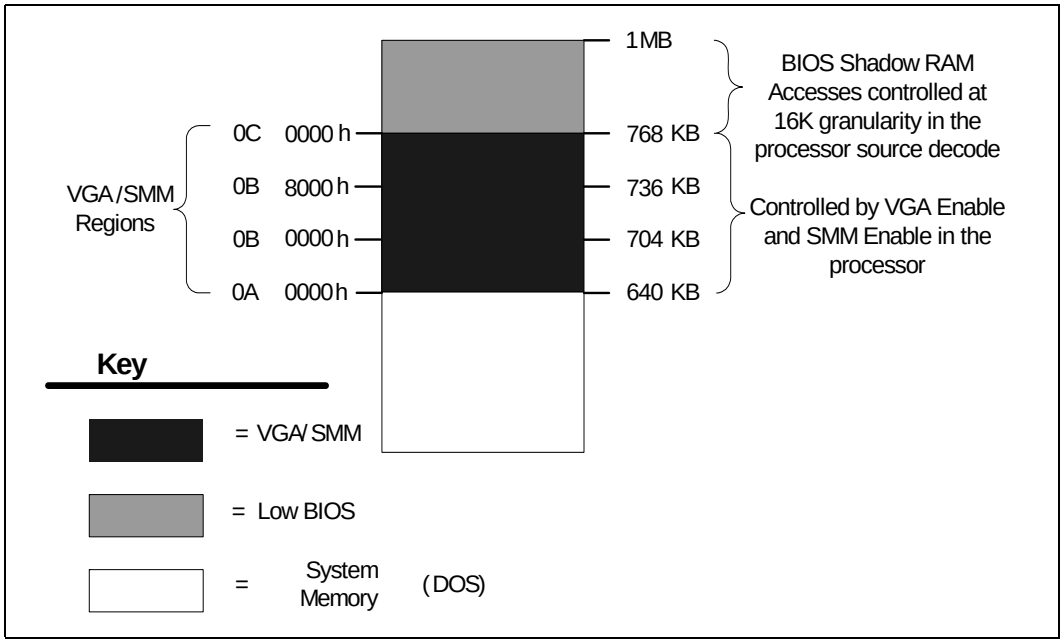
These address ranges are always mapped to system DRAM memory, regardless of the system configuration. The top of main memory below 4 G is defined by the Top of Low Memory (TOLM). Memory between 4 GB and TOHM is extended system memory. Since the platform may contain multiple processors, the memory space 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 IIO 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 IIO positively decodes these ranges via a couple of software programmable range registers. For outbound transactions, it would be an error for IIO to receive non-coherent accesses to these addresses from Intel® QPI. However, the IIO 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.

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

Figure 64 shows the memory address regions below 1 MB. These regions are legacy access ranges.

Figure 64. VGA/SMM and Legacy C/D/E/F Regions



6.1.2.1 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 IIO. If the VGAEN bit is set in the IIO PCI bridge control register (BCR) of a PCI Express port, then transactions within the VGA space 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 (per the IIO 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. See also the *PCI-PCI Bridge 1.2 Specification* for further details on the VGA decoding. 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 IIO does not support the MDA (monochrome display adapter) space independent of the VGA space.

Note: For a Intel® Xeon® processor C5500/C3500 series DP configuration, only one of the four PCIe ports in the legacy Intel® Xeon® processor C5500/C3500 series may have the VGAEN bit set.

The VGA memory address range can also be mapped to system memory in SMM. The IIO 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. See Table 106 for further details of inbound and outbound VGA decoding.

6.1.2.2 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). IIO 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 PCH. IIO does not explicitly decode inbound accesses to the E/F address region. It is expected that the DRAM low range that IIO decodes will be setup to cover the E/F address range. By virtue of that, the IIO 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 IIO does not explicitly decode these regions for inbound accesses. Software must program one of the system DRAM memory decode ranges that the IIO uses for inbound system memory decoding to include these ranges.

All outbound accesses to the C thorough 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 IIO, if one exists; else it is an error condition.

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

6.1.3 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 IIO uses for inbound system memory decoding to include this address range. The IIO will forward inbound accesses to this region to system memory (unless any of these access addresses fall within a protected dram range). It would be an error for IIO to receive outbound accesses to an address in this region, other than snoop requests from Intel® QPI links. However, the IIO 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.

6.1.3.1 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. IIO would completer abort all inbound transactions that target these address ranges. IIO should not receive transactions that target these addresses in the outbound direction, but IIO does not explicitly check for this error condition but rather subtractively forwards such transactions to the subtractive decode port of the IIO, if one exists downstream.

The location (1 MB aligned) and size (from 512 KB to 8 MB) in IIO can be programmed by software. This range check by IIO can also be disabled by the TSEG_EN control bit.

6.1.4 PAM Memory Area Details

There are 13 memory regions from 768 KB to 1 MB (0C0000h - 0FFFFFh) which comprise the PAM Memory Area. These regions can be programmed as Disabled, Read Only,Write Only and R/W from a DRAM perspective. This region can be used to shadow the BIOS region to DRAM for faster access. See the processor's SAD_PAM0123 and SAD_PAM456 registers for details.

Non-snooped accesses from PCI Express or DMI to this region are always sent to DRAM.

6.1.5 ISA Hole (15 MB –16 MB)

A hole can be created at 15 MB-16 MB as controlled by the fixed hole enable bit (HEN) in the processor's SAD_HEN register. Accesses within this hole are forwarded to the DMI Interface. The range of physical DRAM memory disabled by opening the hole is not remapped to the top of the memory – that physical DRAM space is not accessible. This 15 MB-16 MB hole is an optionally enabled ISA hole.

6.1.6 Memory Address Range TOLM – 4 GB

6.1.6.1 PCI Express Memory Mapped Configuration Space (PCI MMCFG)

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.

6.1.6.2 MMIOL

Address Region	From	To
MMIOL	GMMIOL.Base	GMMIOL.Limit

This region is used for PCIE device memory addressing below 4 GB. Each IIO in the system is allocated a portion of this address range and individual PCIe ports and other integrated devices within an IIO (e.g. Intel® QuickData Technology DMA BAR, I/OxAPIC MBAR) use sub-portions within that range. IIO-specific requirements define how software allocates this system region amongst IIOs to support of peer-to-peer between IIOs. See Section 6.4.3, “Intel® VT-d Address Map Implications” for details of these restrictions. Each IIO has a couple of MMIOL address range registers (LMMIOL and GMMIOL) to support local and remote peer-to-peer in the MMIOL address range. See Section 6.4, “IIO Address Decoding” for details of how these registers are used in the inbound and outbound MMIOL range decoding.

6.1.6.3 I/OxAPIC Memory Space

Address Region	From	To
I/OxAPIC	FEC0_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 are populated in downstream devices like the PCH and also the IIO’s integrated I/OxAPIC. The range can be further divided by various downstream ports in the IIO and the integrated I/OxAPIC. Each downstream port in IIO contains a Base/Limit register pair (APICBase/APICLimit) to decode its I/OxAPIC range. Addresses that falls within this range are forwarded to that port. Similarly, the integrated I/OxAPIC decodes its I/OxAPIC base address via the ABAR register. The range decoded via the ABAR register is a fixed size of 256 B. 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 4KB in size. It is called the MBAR and 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 are blocked by the IIO and return a completer abort response. Outbound accesses to this address range that are not positively decoded towards any one PCIe port are sent to the subtractive decode port of the IIO. See section Section 6.4.1, “Outbound Address Decoding” and Section 6.4.2, “Inbound Address Decoding” for complete 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 IIO irrespective of the setting of the MemorySpaceEnable bit in the root port P2P bridge register.

6.1.6.4 HPET/Others

Address Region	From	To
HPET/Others	FED0_0000	FEDF_FFFF

This region covers the High performance event timers in the PCH. All inbound/peer-to-peer accesses to this region are completer aborted by the IIO.

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

6.1.6.5 Local XAPIC

Address Region	From	To
Local XAPIC	FEE0_0000	FEEF_FFFF

The local XAPIC space is used to deliver interrupts to the CPU(s). Message Signaled Interrupts (MSI) from PCIe devices that target this address are forwarded as SpcInt messages to the CPU. See Chapter 7.0, “Interrupts,” for details of interrupt routing by IIO.

The processors may also use this region to send inter-processor interrupts (IPI) from one processor to another. But, the IIO is never a recipient of such an interrupt. Inbound reads to this address are considered errors and are completer aborted by IIO. Outbound accesses to this address range should never occur, but IIO does not explicitly check for this error condition but simply forwards the transaction subtractively to its subtractive decode port, if one exists downstream.

6.1.6.6 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, IIO with firmware connected south of it will communicate this on the Intel® QPI ports so that CPU hardware can configure the path to firmware. The IIO does not support accesses to this address range inbound that is, those inbound transactions are aborted and a completer abort response is sent back.

6.1.7 Address Regions above 4 GB

6.1.7.1 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 4GB boundary. The IIO forwards all inbound accesses to this region to DRAM, unless any of these access addresses are also marked protected. See GENPROTRANGE1.BASE and GENPROTRANGE2.BASE registers. 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 IIO treats them as non-coherent. All other addresses are treated as coherent (unless modified via the NS attributes on PCI Express). The IIO should not receive outbound accesses to this region, but the IIO does not explicitly check for this error condition but rather subtractively forwards these accesses to the subtractive decode port, if one exists downstream.

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

6.1.7.2 Memory Mapped IO High

Address Region	From	To
MMIIO	GMMIIO.Base	GMMIIO.Limit

The high memory mapped I/O range is located above main memory. This region is used to map I/O address requirements above 4 GB range. Each IIO in the system is allocated a portion of this system address region and within that portion each PCIe port and other integrated IIO devices (Intel® QuickData Technology DMA BAR) use up a sub-range. IIO-specific requirements define how software allocates this system region amongst IIOs to support of peer-to-peer between IIOs in this address range. See Section 6.4.3, “Intel® VT-d Address Map Implications” for details of these restrictions.

Each IIO has a couple of MMIIO address range registers (LMMIOH and GMMIOH) to support local and remote peer-to-peer in the MMIIO address range. See Section 6.4.1, “Outbound Address Decoding” and Section 6.4.2, “Inbound Address Decoding” for details of inbound and outbound decoding for accesses to this region.

6.1.8 Protected System DRAM Regions

The IIO supports two address ranges 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

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

6.2 IO Address Space

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

- I/O addresses used for VGA controllers.
- I/O addresses used for ISA aliasing.
- I/O addresses used for the PCI Configuration protocol - CFC/CF8.
- I/O addresses used by downstream PCI/PCIE IO devices, typically legacy devices. This space is divided amongst the IIOs in the system. Each IIO can be associated with an IO range. The range can be further divided by various downstream ports in the IIO. Each downstream port in the IIO contains a BAR to decode its IO range. Address that falls within this range is forwarded to its respective IIO, then subsequently to the downstream port in the IIO.

6.2.1 VGA I/O Addresses

Legacy VGA device uses up the addresses 3B0h-3BBh, 3C0h-3DFh. Any PCIe, DMI port in the IIO can be a valid target of these address ranges if the VGAEN bit in the P2P bridge control register corresponding to that port is set (besides the condition where these regions are positively decoded within the P2P I/O address range). In the outbound direction at the PCI-2-PCI bridge (part of PCIe port) direction, by default, the IIO only decodes the bottom 10 bits of the 16-bit I/O address when decoding this VGA address range with the VGAEN bit set in the P2P bridge control register. But when the VGA16DECEN bit is set in addition to VGAEN being set, the IIO performs a full 16-bit decode for that port when decoding the VGA address range outbound. .

Note: For an Intel® Xeon® processor C5500/C3500 series DP configuration, only one of the four PCIe ports in the legacy Intel® Xeon® processor C5500/C3500 series may have the VGAEN bit set.

6.2.2 ISA Addresses

The IIO supports ISA addressing per the *PCI-PCI Bridge 1.2 Specification*. ISA addressing is enabled in a PCIe port via the ISAEN bit in the bridge configuration space. When the VGAEN bit is set in a PCIe port without the VGA16DECEN bit being set, then the ISAEN bit must be set in all the peer PCIe ports in the system.

6.2.3 CFC/CF8 Addresses

These addresses are used by legacy operating systems to generate PCI configuration cycles. These have been replaced with a memory-mapped configuration access mechanism in PCI Express (which only PCI Express aware operating systems utilize). That said, the IIO does not explicitly decode these I/O addresses and 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. See also Table 106, “Inbound Memory Address Decoding” on page 337 and Table 105, “Subtractive Decoding of Outbound I/O Requests from Intel® QPI” on page 334 for further details of I/O address decoding in the IIO.

6.2.4 PCIe 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 PCIe devices. Each IIO is allocated a chunk of I/O address space and there are IIO-specific requirements on how these chunks are distributed amongst IIOs. See Section 6.4.3, “Intel® VT-d Address Map Implications” for details of these restrictions.

6.3 IIO Address Map Notes

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

6.3.2 Non-Coherent Address Space

The IIO supports one coarse main memory range which can be treated as non-coherent by the IIO, i.e. 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 IIO decodes towards system memory. Inbound accesses to the NC range are not snooped.

6.4 IIO Address Decoding

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

6.4.1 Outbound Address Decoding

This section covers address decoding that the IIO performs on a transaction from the Intel® QPI that targets one of the downstream devices/ports of the IIO. In the description in the rest of the section, PCIe refers to all of a standard PCI Express port and DMI, unless noted otherwise.

6.4.1.1 General Overview

- Before any transaction from the Intel® QPI is decoded by the IIO, the NodeID in the incoming transaction must match the NodeIDs assigned to the IIO (any exceptions are noted when required). Else it is an error. See Chapter 11.0, “IIO Errors Handling Summary,” for details of error handling.
- All target decoding toward PCIe, firmware and internal IIO devices follow address based routing. Address based routing follows the standard PCI tree hierarchy routing.
- NodeID based routing is not supported downstream of the Intel® QPI port in the IIO.
- The subtractive decode port in the IIO is the port that is a) the recipient of all addresses that are not positively decoded towards any of the valid targets in the IIO and b) the recipient of all message/special cycles that are targeted at the legacy PCH. For the legacy IIO, the DMI port is the subtractive decode port. For the

non legacy IIO, the Intel® QPI port is the subtractive decode port. Thus all subtractively decoded transactions will eventually target the PCH.

  — The SUBDECEN bit in the IIO Miscellaneous Control Register (IIOMISCCTRL) sets the subtractive port of the IIO.

  — Virtual peer-to-peer bridge decoding related registers with their associated control bits (e.g. VGAEN bit) and other miscellaneous address ranges (I/OxAPIC) of a DMI port are NOT valid (and ignored by the IIO decoder) when they are set as the subtractive decoding port. Subtractive decode transactions are forwarded to the legacy DMI port, irrespective of the setting of the MSE/IOSE bits in that port.

- Unless specified otherwise, all addresses are first positively decoded against all target address ranges. Valid targets are PCIe, DMI, Intel® QuickData Technology DMA, and I/OxAPIC . Beside the standard peer-to-peer decode ranges (refer to the *PCI-PCI Bridge 1.2 Specification* for details) for PCIe ports, 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 the IIO will forward the transaction towards that target.

  — For outbound transactions, when no target is positively decoded, the transactions are sent to the downstream DMI port if it is indicated as the subtractive decode port. In the Intel® Xeon® processor C5500/C3500 series, if DMI is not the subtractive decode port as in a non-legacy Intel® Xeon® processor C5500/C3500 series, the transaction is master aborted.

  — For inbound transactions on a legacy Intel® Xeon® processor C5500/C3500 series, when no target is positively decoded, the transactions are sent to DMI. In a non-legacy Intel® Xeon® processor C5500/C3500 series, when no target is positively decoded, the transactions are sent to the Intel® QPI and eventually to the DMI port on the legacy IIO.

- For positive decoding, the memory decode to each PCIE 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 only 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 PCIe port or the internal CSR configuration space of the IIO.
- The target decoding for internal VTdCSR space is based on whether the incoming CSR address is within the VTdCSR range (limit is 8K plus the base, VTBAR).
- Each PCIE/DMI port in the IIO has one special address range - I/OxAPIC.
- No loopback supported i.e. 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.

6.4.1.2 FWH Decoding

FWH accesses are allowed only from a CPU. Accesses from SMBus or PCIe are not supported. All FWH addresses (4 GB:4 GB-16 MB) and (1 MB:1 MB-128 K) that do not positively decode to the IIO's PCIe ports, are subtractively forwarded to its legacy decode port.

When the IIO receives a transaction from QPI 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 DMI.

6.4.1.3 I/OxAPIC Decoding

I/OxAPIC accesses are allowed only from the Intel® QPI. The IIO provides an I/OxAPIC base/limit register per PCIe port for decoding to I/OxAPIC in downstream components like PXH. The integrated I/OxAPIC in the IIO 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 IO registers on PCIe.

6.4.1.4 Other Outbound Target Decoding

Other address ranges (besides CSR, FWH, I/OxAPIC) that need to be decoded per PCIe/DMI port include the standard P2P bridge decode ranges (mmiol, mmioh, i/o, vga, config). See the *PCI-PCI Bridge 1.2 Specification* and *PCI Express Base Specification,* Revision 1.1 for details. These ranges are also summarized in Table 102, "Outbound Target Decoder Entries" below.

- Intel® QuickData Technology DMA memory BAR
  - — Remote peer-to-peer accesses from Intel® QPI that target Intel® QuickData Technology DMA BAR region are not completer aborted by the IIO. If inbound protection is needed, VTd translation table should be used to protect at the source IIO. If the VTd table is not enabled, a Generic Protected Memory Range could be used to protect. A last defense is to turn off IB P2P MMIO via new bits in the IIOMISCCTRL register.

6.4.1.5 Summary of Outbound Target Decoder Entries

Table 102, "Outbound Target Decoder Entries" provides a list of all the target decoder entries in the IIO, such as PCIe port, required by the outbound target decoder to positively decode towards a target.

Table 102. Outbound Target Decoder Entries

Address Region	Target Decoder Entry	Comments
VGA (Memory space 0xA_0000 - 0xB_FFFF and IO space 0x3B0 - 0x3BB and 0x3C0 - 0x3DF)	4+1[1]	Fixed.
TPM/LT/FW ranges (E/F segs and 4G-16M to 4G)	1	Fixed.
MMIOL	4	Variable. From P2P Bridge Configuration Register Space
I/OxAPIC	4	Variable. From P2P Bridge Configuration Register Space
MMIOH	4	Variable. From P2P Bridge Configuration Register Space (upper 32 bits PM BASE/LIMIT)
CFGBUS	1	Legacy IIO internal bus number should be set to bus 0.
	4	Variable. From P2P Bridge Configuration Register Space for PCIe bus number decode.
Intel® QuickData Technology DMA	8	Variable. Intel® QuickData Technology DMA BAR
VTBAR	1	Variable. Decodes the VT-d chipset registers.
ABAR	1	Variable. Decodes the sub-region within FEC address range for the integrated I/OxAPIC in the IIO.
MBAR	1	Variable. Decodes any 32-bit base address for the integrated I/OxAPIC in the IIO.
IO	4	Variable. From four local P2P Bridge Configuration Register Space of the PCIE port.

1. This is listed as 4+1 entries because each of the four (or five of non-legacy IIO) local P2P bridges have their own VGA decode enable bit and local IIO has to comprehend this bit individually for each port, and local IIO QPIPVGASAD.Valid bit is used to indicate the dual IIO has VGA port or not.

6.4.1.6 Summary of Outbound Memory/IO/Configuration Decoding

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

Note: Intel® Xeon® processor C5500/C3500 series will support configurations cycles that originate only from the CPU. For Intel® Xeon® processor C5500/C3500 series's NTB, inbound CFG is support for access to the Secondary configuration registers.

Table 103. Decoding of Outbound Memory Requests from Intel® QPI (from CPU or Remote Peer-to-Peer)

Address Range	Conditions[1]	IIO Behavior
Intel® QuickData Technology DMA BAR, I/OxAPIC BAR, ABAR, VTBAR	CB_BAR, ABAR, MBAR, VTBAR and remote p2p access	Completer Abort
	CB_BAR, ABAR, MBAR, VTBAR and not remote p2p access	Forward to that target
All other memory accesses	Not (CB_BAR, ABAR, MBAR, VTBAR, TPM) and one of the downstream ports[2] positively claimed the address	Forward to that port
	Not (CB_BAR, ABAR, MBAR, VTBAR, TPM) and none of the downstream ports positively claimed the address and DMI is the subtractive decode port	Forward to DMI (legacy Intel® Xeon® processor C5500/C3500 series)
	Not (CB_BAR, ABAR, MBAR, VTBAR, TPM) and none of the downstream ports positively claimed the address and DMI is not the subtractive decode port	Master Abort locally (non-legacy Intel® Xeon® processor C5500/C3500 series)

1. See description before this table for clarification of what is actually described in the table
2. For this table, NTB is considered to be one of the 'downstream ports'.

Table 104. Decoding of Outbound Configuration Requests from Intel® QPI and Decoding of Outbound Peer-to-Peer Completions from Intel® QPI

Address Range	Conditions	IIO Behavior
Bus 0	Bus 0 and legacy IIO and device number matches one of internal device numbers	Forward to that internal device.
	Bus 0 and legacy IIO and device number does NOT match one of IIO's internal device numbers	If remote peer-to-peer configuration transaction, master abort, else forward to the downstream subtractive decode port i.e. the legacy DMI port If the transaction is a configuration request, the request is forwarded as a Type 0[1] configuration transaction to the subtractive decode port
	Bus 0 and NOT legacy IIO	Master Abort
Bus 1-255	Bus 1-255 and it matches the IOHBUSNO and device number matches one of the IIO's internal device numbers	If remote peer-to-peer configuration transaction, master abort, else forward to that internal device.
	Bus 1-255 and it matches the IOHBUSNO and device number does NOT match any of the IIO's internal device numbers	Master Abort
	Bus 1-255 and it does not match the IOHBUSNO but positively decodes to one of the downstream PCIe ports	Forward to that port. Configuration requests are forwarded as a Type 0[2] (if bus number matches secondary bus number of port) or a Type 1.
	Bus 1-255 and it does not match the IOHBUSNO and does not positively decode to one of the downstream PCIe ports and DMI is the subtractive decode port	Forward to DMI[3]. Forward configuration request as Type 0/1, depending on secondary bus number register of the port.
	Bus 1-255 and it does not match the IOHBUSNO and does not positively decode to one of the downstream PCIe ports and DMI is not the subtractive decode port	Master Abort

1. When forwarding to DMI, Type 0 transaction with any device number is required to be forwarded by the IIO (unlike the standard PCI Express root ports)
2. If a downstream port is a standard PCI Express root port, then PCI Express spec requires that all non-zero-device numbered Type0 transactions are master aborted by the root port. If the downstream port is non-legacy DMI, then Type 0 transaction with any device number is allowed/forwarded.
3. When forwarding to DMI, Type 0 transaction with any device number is required to be forwarded by the IIO (unlike the standard PCI Express root ports)

Table 105, "Subtractive Decoding of Outbound I/O Requests from Intel® QPI" details IIO behavior when no target has been positively decoded for an incoming I/O transaction from Intel® QPI.

Table 105. Subtractive Decoding of Outbound I/O Requests from Intel® QPI

Address Range	Conditions[1]	IIO 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

1. See description before this table for clarification of what is actually described in the table.

Table 104, "Decoding of Outbound Configuration Requests from Intel® QPI and Decoding of Outbound Peer-to-Peer Completions from Intel® QPI" details IIO behavior for configuration requests from Intel® QPI and peer-to-peer completions from Intel® QPI.

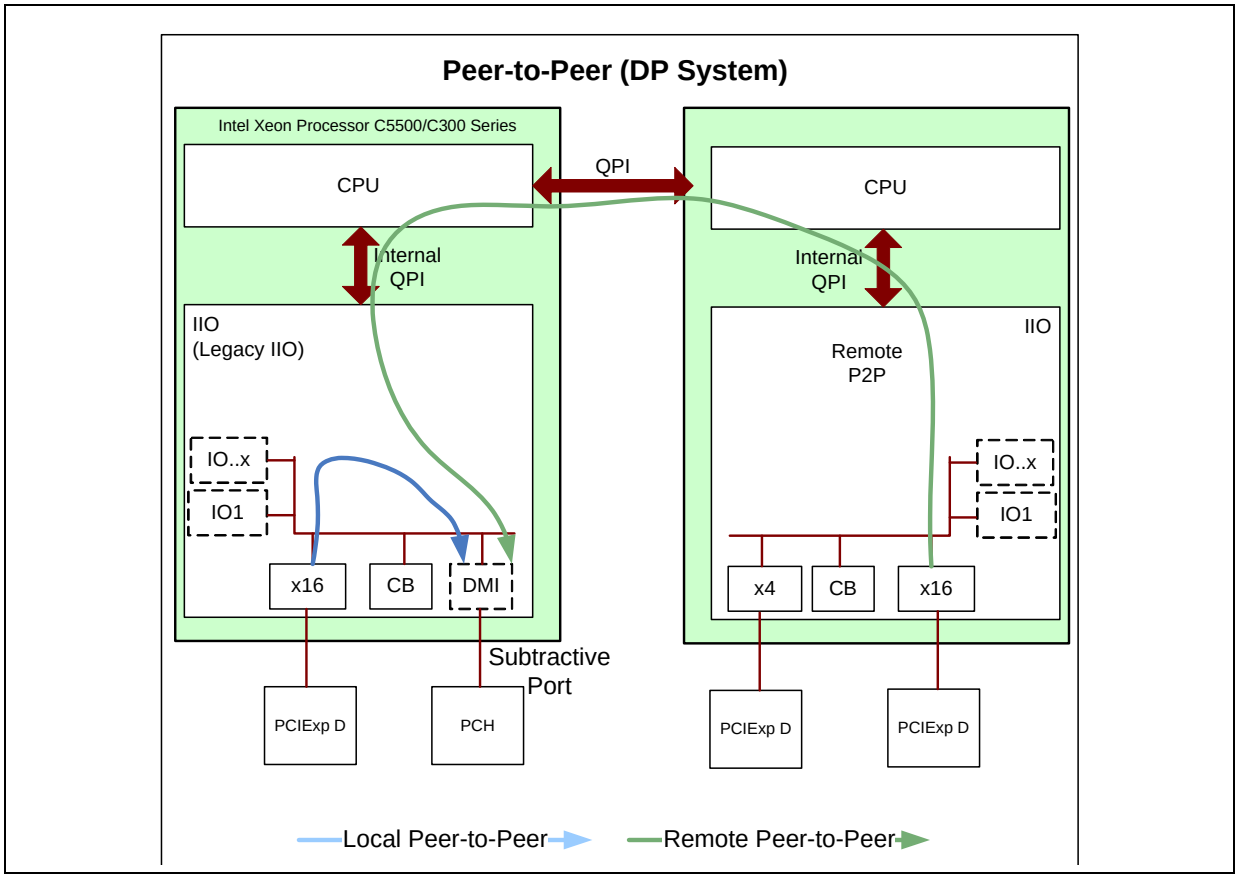
6.4.2 Inbound Address Decoding

This section covers the decoding that is done on any transaction that is received on a PCIE or DMI port or any transaction that originates from the Intel®QuickData Technology DMA port.

6.4.2.1 Overview

- All inbound addresses that fall above the top of Intel® QPI physical address limit are flagged as errors by the IIO. Top of Intel® QPI physical address limit is dependent on the Intel® QPI profile. Register IIOMISCCTRL: IIO MISC Control defines the top-of- Intel® QPI-physical memory.
- Inbound decoding towards main memory in the IIO 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 socket using the Intel® QPI 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 IIO decoder is cognizant of that and steers these addresses towards 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® QPI (remote peer-to-peer) and peer-to-peer not crossing Intel® QPI (local peer-to-peer). See Figure 65, "Intel® Xeon® Processor C5500/C3500 Series Only: Peer-to-Peer Illustration" on page 336 for illustration of remote peer-to-peer.The second step involves actual target decoding for local peer-to-peer (if transaction targets another device south of the IIO) and also involves source decoding using Intel® QPI source address decoders for remote peer-to-peer.
  - A pair of base/limit registers are provided for the IIO 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 (i.e. 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. The VGA memory addresses also overlap one of the system memory address regions, but the IIO always treats the VGA addresses as peer-to-peer addresses. VGA I/O addresses (3B0h-3BBh, 3C0h-3DFh) always are forwarded to the VGA I/O agent of the system. The IIO performs only 16-bit VGA I/O address decode inbound.
  - Subtractively decoded inbound addresses are forwarded to the subtractive decode port of the IIO.
- Inbound accesses to I/OxAPIC, FWH, and Intel® QuickData Technology DMA BAR, are blocked by the IIO (completer aborted).

Figure 65. Intel® Xeon® Processor C5500/C3500 Series Only: Peer-to-Peer Illustration



6.4.2.2 Summary of Inbound Address Decoding

Table 106, "Inbound Memory Address Decoding" summarizes the IIO behavior on inbound memory transactions from any PCIe port. This table is only intended to show the routing of transactions based on the address. It is not intended to show the details of several control bits that govern forwarding of memory requests from a given PCI Express port. See the *PCI Express Base Specification,* Revision 2.0 and the registers chapter for details of these control bits.

Table 106. Inbound Memory Address Decoding (Sheet 1 of 2)

Address Range	Conditions	IIO Behavior
DRAM	Address within 0:TOLM or 4GB:TOHM and SAD hit	Forward to Intel® QPI.
Interrupts	Address within FEE00000-FEEFFFFF and write	Forward to Intel® QPI.
	Address within FEE00000-FEEFFFFF and read	UR response
HPET, I/OxAPIC, TSeg, Relocated CSeg, FWH, VTBAR[1] (when enabled), Protected Intel® VT-d range Low and High, Generic Protected dram range, Intel® QuickData Technology DMA and I/OxAPIC BARs[2]	• FC00000-FEDFFFFF or FEF00000-FFFFFFFF • TOCM >= Address >= TOCM-64GB • VTBAR • VT-d_Prot_High • VT-d_Prot_Low • Generic_Prot_DRAM • Intel® QuickData Technology DMA BAR • I/OxAPIC ABAR and MBAR	Completer abort
VGA[3]	Address within 0A0000h-0BFFFFh and main switch SAD is programmed to forward VGA	Forward to Intel® QPI
	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 DMI port is the subtractive decoding port	Forward to DMI
	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 DMI is not the subtractive decode port	Master abort
Other peer-to-peer[4]	Address within LMMIOL.BASE/LMMIOL.LIMIT or LMMIOH.BASE/LMMIOH.LIMIT and a PCIE port positively 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 and DMI is the subtractive decoding port	Forward to DMI
	Address within LMMIOL.BASE/LMMIOL.LIMIT or LMMIOH.BASE/LMMIOH.LIMIT and no PCIE port decoded as target and DMI is not the subtractive decoding port	Master Abort Locally
	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® QPI

Table 106. Inbound Memory Address Decoding (Sheet 2 of 2)

Address Range	Conditions	IIO Behavior
DRAM Memory holes and other non-existent regions	• {4G <= Address <= TOHM (OR) 0 <= Address <= TOLM } AND address does not decode to any socket in Intel® QPI source decoder • Address > TOCM • When VT-d translation enabled, and guest address greater than 2^GPA_LIMIT	Master Abort
All Else		Forward to subtractive decode port for legacy Intel® Xeon® processor C5500/C3500 series. Aborts locally for non-legacy Intel® Xeon® processor C5500/ C3500 series.

1. The VTBAR range would be within the MMIOL range of that IIO. By that token, VTBAR range can never overlap with any dram ranges.
2. The Intel® QuickData Technology DMA BAR and I/OxAPIC MBAR regions of an IIO overlap with MMIOL/MMIOH ranges of that IIO
3. Intel® QuickData Technology DMA does not support generating memory accesses to the VGA memory range and it will abort all transactions to that address range. Also, if peer-to-peer memory read disable bit is set, VGA memory reads are aborted
4. If peer-to-peer memory read disable bit is set, then peer-to-peer memory reads are aborted

Inbound I/O and configuration transactions from any PCIe port are not supported and will be master aborted.

6.4.3 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 VT-d. Inbound I/O, configuration and message decode and forwarding happens the same whether VT-d is enabled or not. For memory transaction decode, the host address map in 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-VT-d mode. There is not a fixed guest address map that the IIO 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) i.e. it is OS dependent. IIO 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, the IIO also supports an additional memory range called the VTBAR range and this range is used to handle accesses to VT-d related chipset registers. Only aligned DWORD/QWORD accesses are allowed to this region. Only outbound and SMBus accesses are allowed to this range and also these can only be accesses outbound from Intel® QPI. *Inbound accesses to this address range are completer aborted by the IIO.*

§ §

7.0 Interrupts

7.1 Overview

This chapter describes how interrupts are handled in the IIO module. See the Software Developers Manual for details on how the CPUs process interrupts.

The IIO module 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 IIO or forwarded to the Direct Media Interface (DMI) (See the section on Legacy Interrupt Handling). When legacy interrupts are forwarded to DMI, the compatibility bridge either converts them to MSI writes via its integrated I/OxAPIC, or handles them via the legacy 8259 controller.

All root port interrupt sources within the IIO (hot plug, error, power management) support both MSI mode interrupt delivery and legacy INTx mode interrupt delivery. Intel® QuickData Technology supports MSIX and legacy INTx interrupt deliveries. Where noted, the root port interrupt sources (except the error source) also support the ACPI-based mechanism (via GPE messages) for system driver notification. IIO also supports generation of SMI/PMI/NMI interrupts directly from the IIO to the processor (bypassing PCH), in support of IIO error reporting. For Intel® QPI-defined legacy virtual message (VLW) signaling, IIO supports an inband VLW interface to the legacy bridge and an inband interface on Intel® QPI. IIO logic handles the conversion between the two.

7.2 Legacy PCI Interrupt Handling

On PCI Express, interrupts are represented with either MSI-x or inbound interrupt messages (Assert_INTx/De-assert_INTx). For legacy interrupts, the integrated I/OxAPIC in IIO converts the legacy interrupt messages from PCI Express to MSI interrupts. If the I/OxAPIC is disabled (via the mask bits in the I/OxAPIC table entries), then the messages are routed to the legacy PCH. The subsequent paragraphs describe how IIO handles this INTx message flow, from its PCI Express ports and internal devices.

The IIO (both legacy and non-legacy) tracks the assert/de-assert messages for each of the four interrupts INTA, INTB, INTC, INTD from each PCI Express port (including when configured as NTB) and Intel® QuickData Technology DMA. Each of these interrupts from each root port is routed to a specific I/OxAPIC table entry (see Table 108 for the mapping) in that IIO. 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 PCH, as controlled by the mask bit in the Redirection Table, provided the ‘Disable PCI INTx Routing to PCH’ bit is clear in the QPIPINTRC register.

There is a 1:1 correspondence between the message type received from PCI Express and the message type forwarded to the legacy PCH. For example, if a PCI Express port INTA message is masked in the integrated I/OxAPIC, then it is forwarded to legacy PCH as INTA message (subject to the ‘Disable Interrupt Routing to PCH’ bit being clear).

An IIO is not always guaranteed to have its DMI port enabled for legacy. When an IIO's DMI port is disabled for legacy in non-legacy IIOs, then it has to route the INTx messages it receives from its downstream PCI Express ports, to its coherent interface, provided they are not serviced via the integrated I/OxAPIC.

7.2.1 Integrated I/OxAPIC

The integrated I/OxAPIC in IIO converts legacy PCI Express interrupt messages into MSI interrupts. The I/OxAPIC appears as a PCI Express end-point device in the IIO configuration space. The I/OxAPIC has a 24-deep table that allows for 24 unique MSI interrupts. This table is programmed via either the MBAR memory region or the ABAR memory region.

In legacy IIO with DMI, there are potentially 25 unique legacy interrupts possible, 4 root ports * 4 (sources #1 - #4) + 4 for Intel® QuickData Technology DMA (source #6) + 1 IIO RootPorts/core (source #8) + 4 for DMI (source #5) as shown in Table 108. These are mapped to the 24 entries in the I/OxAPIC as shown in the table. The distribution is based on guaranteeing that there is at least one un-shared interrupt line (INTA) for each PCI-E port (from x16 down to x4), and two Intel® QuickData Technology DMA (INTA and INTB) as a possible source of interrupt.

Table 107. Interrupt Source in IOxAPIC Table Mapping

Interrupt Source	PCI Express Port Devices	INT[A-D] Used
1	PCIE Port 0	A,B,C,D/x16, x8, x4
2	PCIE Port 1	A, B, C, D /x4
3	PCIE Port 2	A, B, C, D/x8, x4
4	PCIE Port 3	A, B, C, D/x4
5	PCIE Port (DMI)	A, B, C, D/x4
6	Intel® QuickData Technology DMA	A, B, C, D
8	Root Port	A

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 109, Table 110, Table 111, and Table 112 provide the format of the interrupt message generated by the I/OxAPIC based on the table values.

Table 108. I/OxAPIC Table Mapping to PCI Express Interrupts (Sheet 1 of 2)

I/OxAPIC Table Entry#	Interrupt Source # in Table 107	PCI Express Virtual Wire Type[1]
0	1	INTA
1	2	INTB
2	3	INTC
3	4, <6>	INTD, <INTD>
4	5	INTA
5	6	INTB
6	1	INTD
7	8	INTA
8	2	INTA
9	2	INTC

Table 108. I/OxAPIC Table Mapping to PCI Express Interrupts (Sheet 2 of 2)

I/OxAPIC Table Entry#	Interrupt Source # in Table 107	PCI Express Virtual Wire Type[1]
10	2	INTD
11	3	INTA
12	3	INTB
13	3	INTD
14	4	INTA
15	4	INTB
16	4	INTC
17	5	INTB
18	5	INTC
19	5	INTD
20	6	INTA
21	6	INTC
22	1	INTC
23	1	INTB

1. < >, [ ], and { } in the table associate an interrupt from a given device number (as shown in the 'PCI Express Port/Intel® QuickData Technology 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 3 corresponds to the Wire-OR of INTD message from source#6 (Intel® QuickData Technology DMA INTD) and INTD message from source#4 (PCIE port #3).

7.2.1.1 Integrated I/OxAPIC EOI Flow

Software can set up each I/OxAPIC entry 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. It 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 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 and if it is still asserted, generate a new MSI message.

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

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

7.2.2 PCI Express INTx Message Ordering

INTx messages on PCI Express are posted transactions and hence must follow the posted ordering rules. For example, if the INTx message is preceded by a memory write A, then the INTx message must push the memory write to a global ordering point before it is delivered to its destination, which could be the I/OxAPIC cluster that

determines further action. This guarantees that any MSI generated from the integrated I/OxAPIC, or from the I/OxAPIC in PCH, if the integrated I/OxAPIC is disabled, will be ordered behind the memory write A, guaranteeing producer/consumer sanity.

7.2.3 INTR_Ack / INTR_Ack_Reply Messages

INTR_Ack and INTR_Ack_Reply messages on DMI 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 IIO via the IntAck cycle on Intel® QuickPath Interconnect. The IntAck transaction issued by the processor socket behaves as an IO read cycle in that the completion for the IntAck message contains the interrupt vector. The IIO converts this cycle to a posted message on the DMI port (no completions).

- IntAck: The IIO forwards the IntAck received on the Coherent Interface (as an NCS transaction) as a posted message INTR_Ack to legacy PCH over DMI. A completion for IntAck is not yet sent on Intel® QuickPath Interconnect.
- INTR_Ack_Reply: The PCH returns the 8-bit interrupt vector from the 8259 controller through this posted VDM. The INTR_Ack_Reply message pushes upstream writes in both the PCH and the IIO. This IIO 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.

7.3 MSI

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.

MSI interrupts generated from PCI Express ports or from integrated functions within the IIO are memory writes to a specific address range, 0xFEEx_xxxx. If interrupt remapping is disabled in the IIO, then the interrupt write directly provides the information regarding the interrupt destination processor and interrupt vector. Details are as shown in Table 109 and Table 110. If interrupt remapping is enabled in the IIO, then interrupt write fields are interpreted as shown in Table 111 and Table 112.

Table 109. MSI Address Format when Remapping Disabled (Sheet 1 of 2)

Bits	Description
31:20	FEEh
19:12	**Destination ID:** This will be the bits 63:56 of the I/O Redirection Table entry for the interrupt associated with this message. In IA32 mode: For physical mode interrupts, this field becomes APICID[7:0] on the QPI interrupt packet and APICID[31:8] are reserved in the QPI packet. For logical cluster mode interrupts, 19:16 of this field becomes APICID[19:16] on the QPI interrupt packet and 15:12 of this field becomes APICID[3:0] on the QPI interrupt packet. For logical flat mode interrupts, 19:12 of this field becomes APICID[7:0] on the QPI interrupt packet.
11:4	**EID:** this will be the bits 55:48 of the I/O Redirection Table entry for the interrupt associated with this message.

Table 109. MSI Address Format when Remapping Disabled (Sheet 2 of 2)

Bits	Description
3	**Redirection Hint**: This bit allows the interrupt message to be directed to one among many targets, based on chipset redirection algorithm. 0 = The message will be delivered to the agent (CPU) listed in bits 19:4 1 = The message will be delivered to an agent based on the IIO redirection algorithm and the scope the interrupt as specified in the interrupt address. The Redirection Hint bit will be a 1 if bits 10:8 in the Delivery Mode field associated with corresponding interrupt are encoded as 001 (Lowest Priority). Otherwise, the Redirection Hint bit will be 0.
2	**Destination Mode:** This 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 QPI.
1:0	00

Table 110. 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 i.e. asserted
13:12	00
11	**Destination Mode:** This is the corresponding bit from the I/O Redirection Table entry. 1=logical mode and 0=physical mode. Note that this bit is set to 0 before being forwarded to QPI.
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 111. MSI Address Format when Remapping is Enabled

Bits	Description
31:20	FEEh
19:4	**Interrupt Handle:** IIO looks up an interrupt remapping table in main memory using this field as an offset into the table
3	**Sub Handle Valid**: When IIO looks up the interrupt remapping table in main memory, and if this bit is set, IIO 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:** IIO hardware ignores this bit
1:0	00

Table 112. MSI Data Format when Remapping is Enabled

Bits	Description
31:16	**Reserved** - IIO 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 IIO converts memory writes to this address (both PCI Express and internal sources) as a IntLogical, IntPhysical transactions on the Intel® QuickPath Interconnect.

The IIO module supports two MSI vectors per root port for hot plug, power management, and error reporting.

7.3.1 Interrupt Remapping

Interrupt remapping architecture provides for interrupt filtering for virtualization/ security usages so an arbitrary device cannot interrupt an arbitrary processor in the system.

When interrupt remapping is enabled in the IIO, then the IIO looks up a table in main memory to obtain the interrupt target processor and vector number. When the IIO receives an MSI interrupt in which the 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 IIO/PCH, and the remapping is turned on, then the IIO 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 the sub handle valid field in the 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:

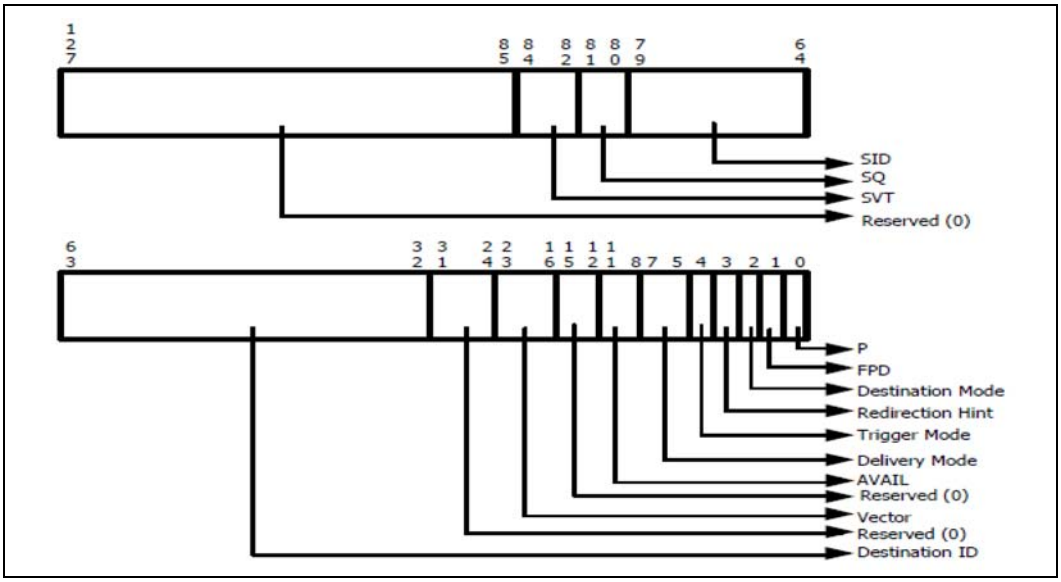
Memory Offset = Final Interrupt Handle * 16

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 the Interrupt Transformation Table Entry (IRTE).

The information that was formerly obtained directly from the MSI address/data fields is now obtained via the IRTE when remapping is turned on. In addition, the IRTE also provides a way to authenticate an interrupt via the Requester ID, i.e. the IIO 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, then the interrupt is further processed, else the interrupt is dropped and error signaled. Subsequent sections in this chapter describe how fields in either the IRTE when remapping is enabled, or MSI address/data when remapping is disabled, are used by the chipset to generate IntPhysical/Logical interrupts on Intel® QuickPath Interconnect.

Figure 66. Interrupt Transformation Table Entry (IRTE)



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

7.3.2 MSI Forwarding: IA32 Processor-based Platform

IA-32 interrupts have two modes: legacy mode and extended mode. Legacy mode has been supported in all chipsets to date. Extended mode is a new mode that allows for scaling beyond 60/255 threads in logical/physical mode operation. Legacy mode has only an 8-bit APICID; extended mode supports 32-bit APICID (obtained via IRTE).

7.3.2.1 Legacy Logical Mode Interrupts

The IIO broadcases IA32 legacy logical interrupts to all processors in the system. It is the responsibility of the CPU to drop interrupts that are not directed to one of its local APICs. The IIO supports hardware redirection for IA32 logical interrupts (see Section 7.3.2.1.1) For IA32 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 the Intel® QuickPath Interconnect NodeID. The assumption is made that the APICID field in the MSI address only includes valid/enabled APICs for that interrupt.

7.3.2.1.1 Legacy Logical Mode Interrupt Redirection - Redirection Based on Vector Number

In logical flat mode when redirection is enabled, the IIO looks at 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 corresponding 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:

- The IIO adds a value of 4 to the selected APICs address bit location. If the APIC corresponding to modulo 8 of that value is also not a valid target because the bit mask corresponding to that APIC is clear in the MSI address, then,

- The IIO adds a value of 2 to the original selected APICs address bit location. If the APIC corresponding to modulo 8 of that value is also not a valid target, then the IIO adds a value of 4 to the previous value and takes the modulo 8 of the resulting value. If that corresponding APIC is also not a valid target, then,
- The IIO adds a value of 3 to the original selected APICs address bit location. If the APIC corresponding to modulo 8 of that value is also not a valid target, then the IIO adds a value of 4 to the previous value and takes the modulo 8 of the resulting value. If that corresponding APIC is also not a valid target, then,
- The IIO adds a value of 1 to original selected APICs address bit location. If the APIC corresponding to modulo eight of that value is also not a valid target, then IIO adds a value of 4 to the previous value and takes the modulo 8 of the resulting value. If that corresponding APIC is also not a valid target, then it is an error condition.

In logical cluster mode (except when APICID[19:16] != F), the redirection algorithm works as described above except the IIO only redirects between four APICs instead of eight in the flat mode. Therefore, the IIO uses only vector number bits [5:4] by default (selectable to bits[4:3]/2:1/1:0 based on bits 4:3 of QPIPINTRC register). The search algorithm to identify a valid APIC for redirection in the cluster mode is to:

- First select the APIC that corresponds to the bit position identified with the chosen vector number bits. If corresponding bit in the MSI address bits A[15:12] is clear, then,
- The IIO adds a value of 2 to the original selected APICs address bit location. If the APIC corresponding to modulo four of that value is also not a valid target, then
- The IIO adds a value of 1 to original selected APICs address bit location. If the APIC corresponding to modulo 4 of that value is also not a valid target, then the IIO adds a value of 2 to the previous value and takes the modulo 4 of the resulting value. If that corresponding APIC is also not a valid target, then it is an error condition.

7.3.3 External IOxAPIC Support

External IOxAPICs, such as those within a PXH, PCH, etc. are supported. These devices require special decoding of a fixed address range FECx_xxxx in the IIO module. The IIO module provides these decoding ranges, which are outside the normal prefetchable and non-prefetchable windows supported in each root port. More information is in the chapter on System Address Maps.

The local APIC supports EOI messages to external IOxAPICs that need the EOI message. It also supports EOI messages to the internal IOxAPIC. The IIO module, if enabled, can be programmed to broadcast/multicast the EOI message to all downstream PCIe/DMI ports. The broadcast/multicast of the EOI message is also supported for the internal IOxAPIC. The EOI message can be disabled globally using the global EOI disable bit in the EOI_CTRL register of Device #0, or can be disabled on a per PCIe/DMI port basis.

7.4 Virtual Legacy Wires (VLW)

Discrete signals that existed on previous-generation processors (e.g.: NMI/SMI#/INTR/ INIT#/A20M#/FERR#, etc.) are now implemented as messages. This capability is referred to as “Virtual Legacy Wires” or “VLW Messages”. Signals that were discrete wires that went between the PCH and the processor are now communicated using Vendor Defined messages over the DMI interface.

In DP configurations the Vendor Defined messages are broadcast to both processor sockets. The message is routed over the Intel® QPI bus to the non-legacy socket. Only the destined local APIC of one processor socket claims the VLW message; all other local

APICs that were not specifically addressed will drop the message. There are two per core and one per thread, yielding eight local APICs in Intel® Xeon® processor C5500/C3500 series when all four cores with SMT are enabled.

7.5 Platform Interrupts

General Purpose Event (GPE) interrupts are generated as a result of hot plug and power management events. GPE interrupts are conveyed as VLW messages routed to the IOxAPIC within the PCH. In response to a GPE VLW, the PCH IOxAPIC can be programmed to send out an MSI message or legacy INTx interrupt. Either socket could be the destination of the interrupt message. The IIO module tracks and maintains the state of the three level-sensitive GPE messages (Assert/Deassert_GPE, Assert/Deassert_HPGPE, Assert/Deassert_PMEGPE). The legacy IIO module (the processor socket that directly connects to the PCH component) has this responsibility.

Various RAS events and errors can cause an IntPhysical PMI/SMI/NMI interrupt to be generated directly to the processor, bypassing the PCH.

All Correctable Platform Event Interrupts (CPEI) are routed to the IIO module. This includes PCIe-corrected errors if native handling of these errors has been disabled by the OS. In the case of a Intel® Xeon® processor C5500/C3500 series DP system, corrected errors detected by the non-legacy IIO module are routed to the legacy IIO module. The IIO module combines the corrected error messages from all sources and generates a CPEI message based on the SYSMAP register.

The legacy Intel® Xeon® processor C5500/C3500 series socket maintains a pin (ERR[0]) that represents the state of CPEI. Software can read a status bit (bit 0 of register ERRPINST: Error Pin Status Register) to detect the state of the ERR[0] pin. Once the pin has been set, further CPEI events have no effect. The status bit needs to be reset to detect any additional CPEI events. The ERR[0] pin of the legacy Intel® Xeon® processor C5500/C3500 series can be connected to the PCH to signal the IOxAPIC within the PCH to generate an INTx or MSI interrupt.

7.6 Interrupt Flow

The PCH contains an integrated IOxAPIC and additional downstream external IOxAPICs are supported.

Additionally, each Intel® Xeon® processor C5500/C3500 series contains its own IOxAPIC, in addition to the IOxAPIC that is resident in the PCH, or any additional downstream external IOxAPICs. As a result, the processor supports a flexible interrupt architecture. Interrupts from one socket can be handled by the integrated IOxAPIC within the socket, or may be programmed to be handled by the IOxAPIC within the PCH.

At power-up the Redirection Table Entries (RTE) are masked, the integrated IOxAPIC is unprogrammed, and the Don't_Route_To_PCH bit is reset so all legacy INTx interrupts are routed to the PCH's IOxAPIC.

When an INTx interrupt is disabled within the IIO module IOxAPIC, then legacy INTx interrupts are routed to the PCH IOxAPIC, regardless of the originating socket (legacy or non-legacy). The PCH IOxAPIC can then be programmed to deliver the legacy interrupt to any core within either socket, or to convert the legacy interrupt into an MSI interrupt before delivery to any core within either socket. This also applies to legacy interrupts directly received by the PCH IOxAPIC.

7.6.1 Legacy Interrupt Handled By IIO Module IOxAPIC

When an INTx interrupt is enabled within the IIO module IOxAPIC, then the IIO module IOxAPIC may be programmed to deliver the legacy interrupt depending on the mask :

Mask	DRTPCH	Behavior
0	X	Convert to MSI
1	0	Forward INTx to PCH
1	1	Pend INTx in IOxAPIC

There is no mode in which the integrated IOxAPIC delivers a legacy interrupt directly to the CPU.

If the legacy interrupt is converted to an MSI interrupt, and the Intel® VT-d engine is enabled, then the Intel® VT-d engine can be programmed to perform an interrupt address translation before delivering the interrupt to a core.

7.6.2 MSI Interrupt

MSI interrupts do not need the support of an IOxAPIC, they are routed as a message directly to the intended core. If the Intel® VT-d engine is enabled, it can be programmed to perform an interrupt address translation before delivering the interrupt to a core.

§ §

8.0 Power Management

8.1 Introduction

Intel® Xeon® processor C5500/C3500 series 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.

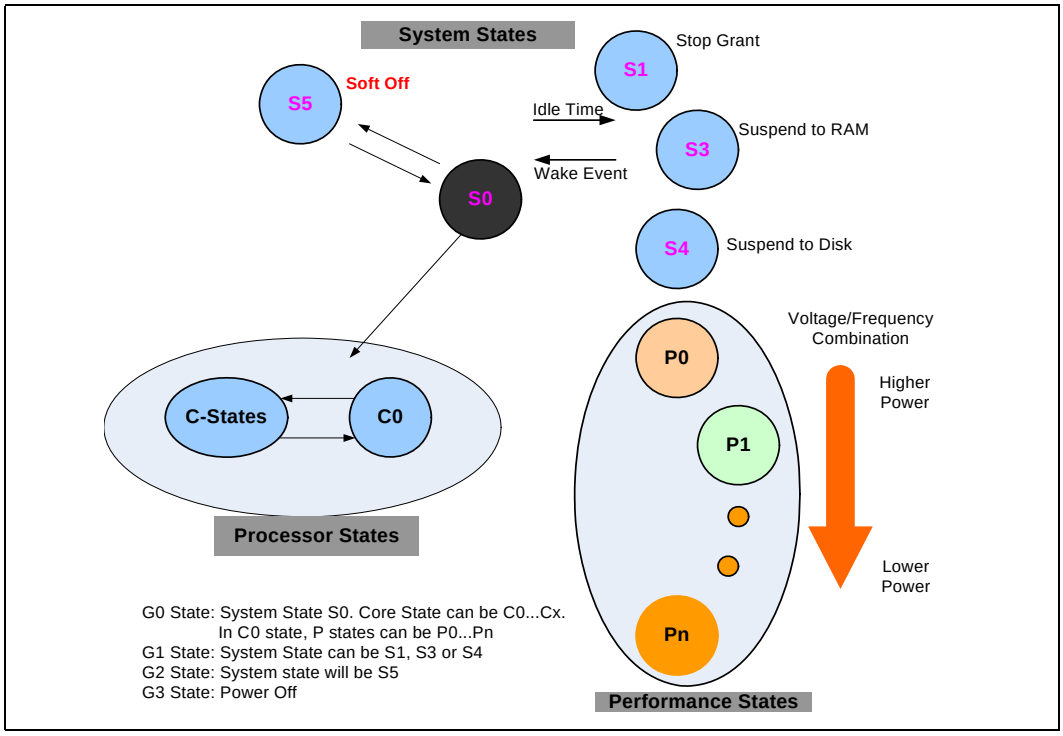
This chapter provides information on the following power management topics:

- ACPI states
- Processor core
- IMC
- Device and slot power
- PCI Express*
- DMI
- Intel® QPI
- Intel® QuickData Technology

8.1.1 ACPI States Supported

Figure 67 shows a high-level diagram of the basic ACPI system and processor states in working state (G0) and sleeping states (G1 & G2). The frequency and voltage might vary by implementation.

Figure 67. ACPI Power States in G0, G1, and G2 States



8.1.2 Supported System Power States

The system power states supported by the Intel® Xeon® processor C5500/C3500 series IIO module are enumerated in Table 113.

Table 113. Platform System States (Sheet 1 of 2)

System State	Description
S0	Full On: [Supported by the IIO module] Normal operation
S1	Stop-Grant: [Supported by the IIO module] No reset or re-enumeration required. Context preserved in caches and memory. Processor cores go to a low power idle state. See Table 120 for details. 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 PCH’s power management control register (PMBase + 04h) and then halts the “monarch”. This will cause the PCH to send the GO_S1_final DMI2 message to the IIO module. The IIO module responds with a NcMsgB-PMREQ(‘S1) handshake with the CPU’s followed by an ACK_Sx DMI2 message to the PCH. (The “monarch” is the thread that executes the S-state entry sequence.) See text for the IIO module sequence.

Table 113. Platform System States (Sheet 2 of 2)

System State	Description
S3	Suspend to RAM (STR) [Supported] This is also known as Standby. CPU, and PCI reset. All context can be lost except memory. This state is commonly known as “Suspend”.
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 that main memory. This state is commonly known as “Hibernate”. Self Refresh is not required.
S5	Soft off [Supported] Power removed.

The IIO module supports the S0 (fully active) state. This is required for full operation. The IIO module also supports a system level S1 (idle) state, but the S2 (power-on suspend) is not supported. The IIO module supports S3/S4/S5 powered-down idle sleep states. In the S3 state (suspend to RAM), the context is preserved in memory by the OS and the CPU places the memory in self-refresh mode to prevent data loss. In the S4/S5 states, platform power and clocking are disabled, leaving only one or more auxiliary power domains functional. Exit from the S3, S4, and S5 states requires a full system reset and initialization sequence.

8.1.3 Processor Core / Package States

- Core: C0, C1E, C3, C6
- Package C0, C3, C6
- Enhanced Intel SpeedStep® Technology

8.1.4 Integrated Memory Controller States

Table 114. Integrated Memory Controller States

State	Description
Power up	CKE asserted. Active mode.
Pre-charge Power down	CKE deasserted (not self-refresh) with all banks closed.
Active Power down	CKE deasserted (not self-refresh) with minimum one bank active.
Self-Refresh	CKE deasserted using device self-refresh.

8.1.5 PCIe Link States

Table 115. PCIe Link States

State	Description
L0	Full on – Active transfer state.
L0s	First Active Power Management low power state – Low exit latency.
L1	Lowest Active Power Management - Longer exit latency.
L3	Lowest power state (power-off) – Longest exit latency.

8.1.6 DMI States

Table 116. DMI States

State	Description
L0	Full on – Active transfer state.
L0s	First Active Power Management low power state – Low exit latency.
L1a	L1a is active state L1 in the DMI Specification.
L3	Lowest power state (power-off) – Longest exit latency.

8.1.7 Intel® QPI States

Table 117. Intel® QPI States

State	Description
L0s	First Active Power Management low power state – Low exit latency.
L1	Lowest Active Power Management - Longer exit latency.

8.1.8 Intel® QuickData Technology State

Table 118. Intel® QuickData Technology States

State	Description
D0	Fully-on state and a pseudo D3hot state.

8.1.9 Interface State Combinations

Table 119. G, S, and C State Combinations

Global (G) State	Sleep (S) State	Processor Core (C) State	Processor State	System Clocks	Description
G0	S0	C0	Full On	On	Full On
G0	S0	C1E	Auto-Halt	On	Auto-Halt
G0	S0	C3	Deep Sleep	On	Deep Sleep
G0	S0	C6	Deep Power Down	On	Deep Power Down
G1	S3	Power off		Off, except RTC	Suspend to RAM
G1	S4	Power off		Off, except RTC	Suspend to Disk
G2	S5	Power off		Off, except RTC	Soft Off
G3	NA	Power off		Power off	Hard off

8.1.10 Supported DMI Power States

The transitions to and from the following power management states are supported on the DMI link:

Table 120. System and DMI Link Power States

System State	CPU State	Description	Link State	Comments
S0	C0	Fully operational / Opportunistic Link Active-State	L0/L0s/L1a[1]	Active-State Power Management
S0	C1E[2]	CPU Auto-Halt	L0/L0s/L1a	Active-State Power Management
S0	C3/C6	Deep Sleep States	L0/L0s/L1a	Active-State Power Management
S1	C1E/C3	The legacy association of S1 with C2 is no longer valid.	L0/L0s/L1a	Active-State Power Management
S3/S4/S5	N/A	STR/STD/Off	L3	Requires Reset. System context not maintained in S5.

1. L1a means active state L1 in the DMI specification
2. The “E” suffix denotes additional minimum voltage-frequency P-state.

8.2 Processor Core Power Management

While executing code, Enhanced Intel Speedstep® Technology optimizes the processor’s frequency and core voltage based on workload. Each frequency and voltage operating point is defined by ACPI as a P-state. The processor is idle when not executing code. ACPI defines a low-power idle state as a C-state. In general, lower power C-states have longer entry and exit latencies.

8.2.1 Enhanced Intel SpeedStep® Technology

The following are key features of Enhanced Intel SpeedStep® Technology:

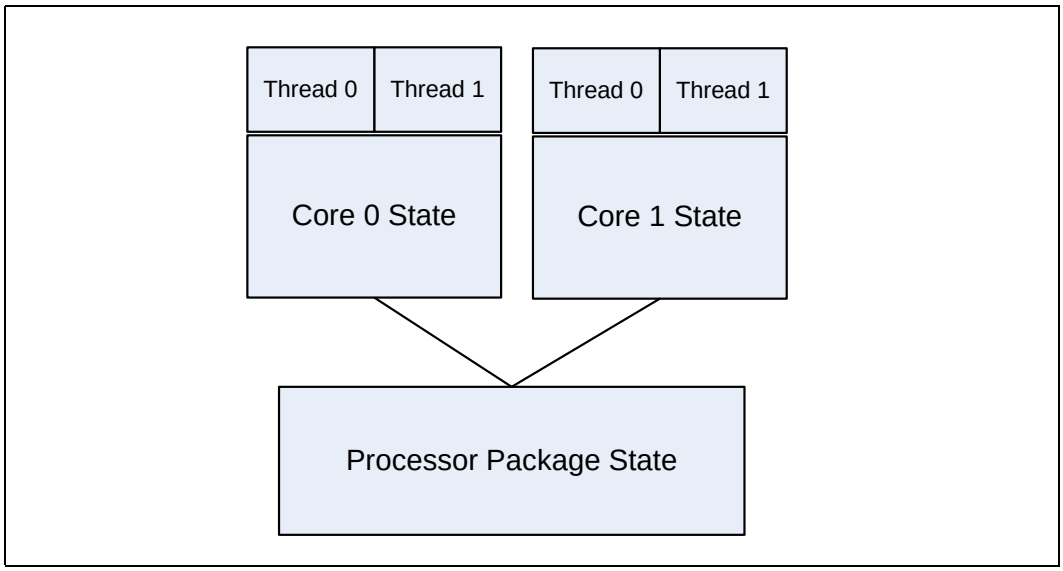
- Multiple frequency and voltage points for optimal performance and power efficiency. These operating points are known as P-states.
- Frequency selection is software-controlled by writing to processor MSRs. The voltage is optimized based on the selected frequency and the number of active processor cores.
  - If the target frequency is higher than the current frequency and a voltage change is required, the voltage is ramped up in steps to an optimized voltage. This voltage is signaled by the VID[7:0] pins to the voltage regulator. Once the voltage is established, the PLL locks on to the target frequency.
  - If the target frequency is lower than the current frequency, then the PLL locks to the target frequency, then, if needed, transitions to a lower voltage by signaling the target voltage on the VID[7:0] pins.
  - All active processor cores share the same frequency and voltage. In a multi-core processor, the highest frequency P-state requested amongst all active cores is selected.
  - Software-requested transitions are accepted at any time. If a previous transition is in progress, then the new transition is deferred until the previous transition is completed.
- The processor controls voltage ramp rates internally to ensure glitch-free transitions.

- Because there is low transition latency between P-states, a significant number of transitions per second are possible.
- The highest frequency/voltage operating point is known as the highest frequency mode (HFM).
- The lowest frequency/voltage operating point is known as the lowest frequency mode (LFM).

8.2.2 Low-Power Idle States

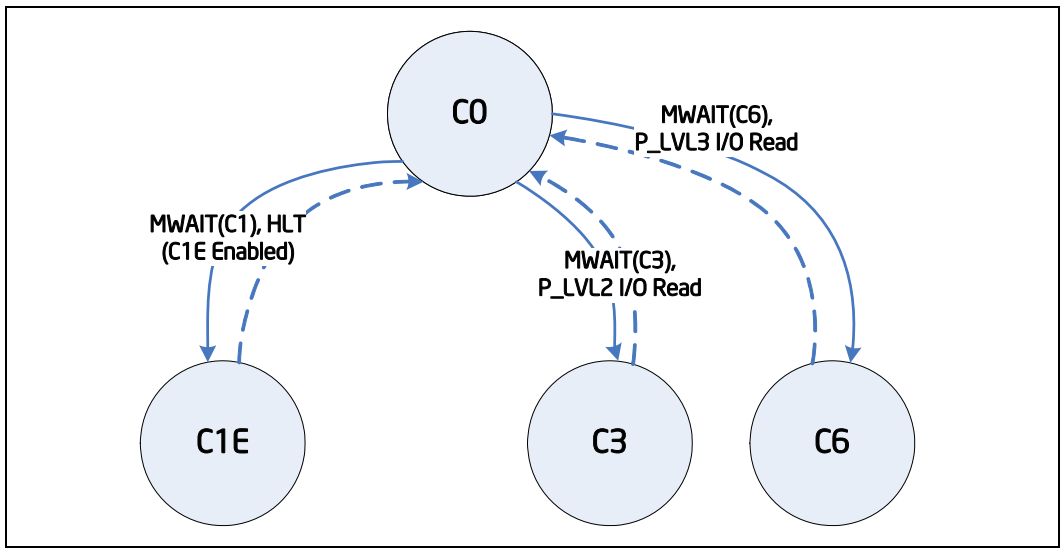
When the processor is idle, low-power idle states (C-states) are used to save power. More power savings actions are taken for numerically higher C-states. However, higher C-states have longer exit and entry latencies. Resolution of C-states occur at the thread, processor core, and processor package level. Thread-level C-states are available if Hyper-Threading Technology is enabled.

Figure 68. Idle Power Management Breakdown of the Processor Cores (Two-Core Example)



Entry and exit of the C-States at the thread and core level are shown in Figure 69.

Figure 69. Thread and Core C-State Entry and Exit



While individual threads can request low power C-states, power-saving actions only take place after the core C-state is resolved. The processor automatically resolves Core C-states. For thread and core C-states, a transition to and from C0 is required before entering any other C-state.

Table 121. Coordination of Thread Power States at the Core Level

Processor Core C-State		Thread 1			
		C0	C1E	C3	C6
Thread 0	C0	C0	C0	C0	C0
	C1E	C0	C1E[1]	C1E[1]	C1E[1]
	C3	C0	C1E[1]	C3	C3
	C6	C0	C1E[1]	C3	C6

Note:
1. If enabled, the core C-state will be C1E if all active cores have also resolved to a core C1 state or higher.

8.2.3 Requesting Low-Power Idle States

The primary software interfaces for requesting low power idle states are through the MWAIT instruction with sub-state hints and the HLT instruction (for C1E). However, software may make C-state requests using the legacy method of I/O reads from the ACPI-defined processor clock control registers, referred to as P_LVLx. This method of requesting C-states provides legacy support for operating systems that initiate C-state transitions via I/O reads.

For legacy operating systems, P_LVLx I/O reads are converted within the processor to the equivalent MWAIT C-state request. Therefore, P_LVLx reads do not directly result in I/O reads to the system. The feature, known as I/O MWAIT redirection, must be enabled in the BIOS.

Note: The P_LVLx I/O Monitor address needs to be set up before using the P_LVLx I/O read interface. Each P-LVLx is mapped to the supported MWAIT(Cx) instruction as follows.

Table 122. P_LVLx to MWAIT Conversion

P_LVLx	MWAIT(Cx)	Notes
P_LVL2	MWAIT(C3)	The P_LVL2 base address is defined in the PMG_IO_CAPTURE MSR.
P_LVL3	MWAIT(C6)	C6. No sub-states allowed

The BIOS can write to the C-state range field of the PMG_IO_CAPTURE MSR to restrict the range of I/O addresses that are trapped and redirected to MWAIT instructions. Any P_LVLx reads outside of this range does not cause an I/O redirection to MWAIT(Cx) request. They fall through like a normal I/O instruction.

Note: When P_LVLx I/O instructions are used, MWAIT substates cannot be defined. The MWAIT substate is always zero if I/O MWAIT redirection is used. By default, P_LVLx I/O redirections enable the MWAIT 'break on EFLAGS.IF' feature which triggers a wakeup on an interrupt even if interrupts are masked by EFLAGS.IF.

8.2.4 Core C-States

Changes in the Intel® Core™ i7 microarchitecture as well as changes in the platform have altered the behavior of C-states as compared to prior Intel platform generations. Signals such as STPCLK#, SLP#, and DPSLP# are no longer used, which eliminates the need for C2 state. In addition, the latency of the C6 state within the new microarchitecture is similar to that of C4 in the Intel Core microarchitecture. Therefore the C4 state is no longer necessary. The following are general rules for all core C-states, unless specified otherwise:

- A core C-State is determined by the lowest numerical thread state (e.g., thread0 requests C1E while thread1 requests C3, resulting in a core C1E state).
- A core transitions to C0 state when:
  - An interrupt occurs.
  - There is an access to the monitored address if the state was entered via an MWAIT instruction.
- For core C1E, and core C3, an interrupt directed toward a single thread wakes only that thread. However, since both threads are no longer at the same core C-state, the core resolves to C0.
- For core C6, an interrupt coming into either thread wakes both threads into C0 state.
- Any interrupt coming into the processor package may wake any core.

Note: The core “C” state resolves to the highest power dissipation “C” state of the threads.

8.2.4.1 Core C0 State

The normal operating state of a core where code is being executed.

8.2.4.2 Core C1E State

C1E is a low power state entered when all threads within a core execute a HLT or MWAIT(C1E) instruction.

A System Management Interrupt (SMI) handler returns execution to either the normal state or the C1E state. See the *Intel® 64 and IA-32 Architecture Software Developer’s Manual, Volume 3A/3B: Stem Programmer’s Guide* for more information.

While a core is in C1E state, it processes bus snoops and snoops from other threads. For more information on C1E, see Section 8.2.5.2.

8.2.4.3 Core C3 State

Individual threads of a core can enter the C3 state by initiating a P_LVL2 I/O read to the P_BLK or an MWAIT (C3) instruction. A core in C3 state flushes the contents of its instruction cache, data cache, and Mid-Level Cache (MLC) to the Last Level Cache (LLC), while maintaining its architectural state. All core clocks are stopped at this point. Because the core's caches are flushed, the processor does not wake any core that is in the C3 state when either a snoop is detected or when another core accesses cacheable memory.

8.2.4.4 Core C6 State

Individual threads of a core can enter the C6 state by initiating a P_LVL3 I/O read or an MWAIT (C6) instruction. Before entering Core C6, the core will save its architectural state to a dedicated SRAM. Once complete, a core can lower its voltage to any voltage, even as low as zero volts. During exit, the core is powered on and its architectural state is restored.

8.2.4.5 C-State Auto-Demotion

In general, deeper C-states, such as C6, have long latencies and higher energy entry/exit costs. The resulting performance and energy penalties become significant when the entry/exit frequency of a deeper C-state is high. Therefore incorrect or inefficient usage of deeper C-states have a negative impact on power savings. In order to increase residency and improve battery life in deeper C-states, the processor supports C-state auto-demotion.There are two C-State auto-demotion options:

- C6 to C3
- C6/C3 To C1E

The decision to demote a core from C6 to C3 or C3/C6 to C1E is based on each core's immediate residency history. Upon each core C6 request, the core C-state is demoted to C3 or C1E until a sufficient amount of residency has been established. At that point, a core is allowed to go into C3/C6. Each option can be run concurrently or individually.

This feature is disabled by default. The BIOS must enable it in the PMG_CST_CONFIG_CONTROL register. The auto-demotion policy is also configured by this register.

8.2.5 Package C-States

The processor supports C0, C1E, C3,and C6 power states. The following is a summary of the general rules for package C-state entry. These apply to all package C-states unless specified otherwise:

- A package C-state request is determined by the lowest numerical core C-state amongst all cores.
- A package C-state is automatically resolved by the processor depending on the core idle power states and the status of the platform components.
  - Each core can be at a lower idle power state than the package if the platform does not grant the processor permission to enter a requested package C-state.
  - The platform may allow additional power savings to be realized in the processor. If given permission, the DRAM will be put into self-refresh in the package C3 and C6.

The processor exits a package C-state when a break event is detected. If DRAM was allowed to go into self-refresh in package C3 or C6 state, it will be taken out of self-refresh. Depending on the type of break event, the processor does the following:

- If a core break event is received, the target core is activated and the break event message is forwarded to the target core.
  - If the break event is not masked, then the target core enters the core C0 state and the processor enters package C0.
  - If the break event is masked, then the processor attempts to re-enter its previous package state.
- If the break event was due to a memory access or snoop request...
  - But the platform did not request to keep the processor in a higher power package C-state, then the package returns to its previous C-state.
  - And if the platform requests a higher power C-state, then the memory access or snoop request is serviced and the package remains in the higher power C-state.

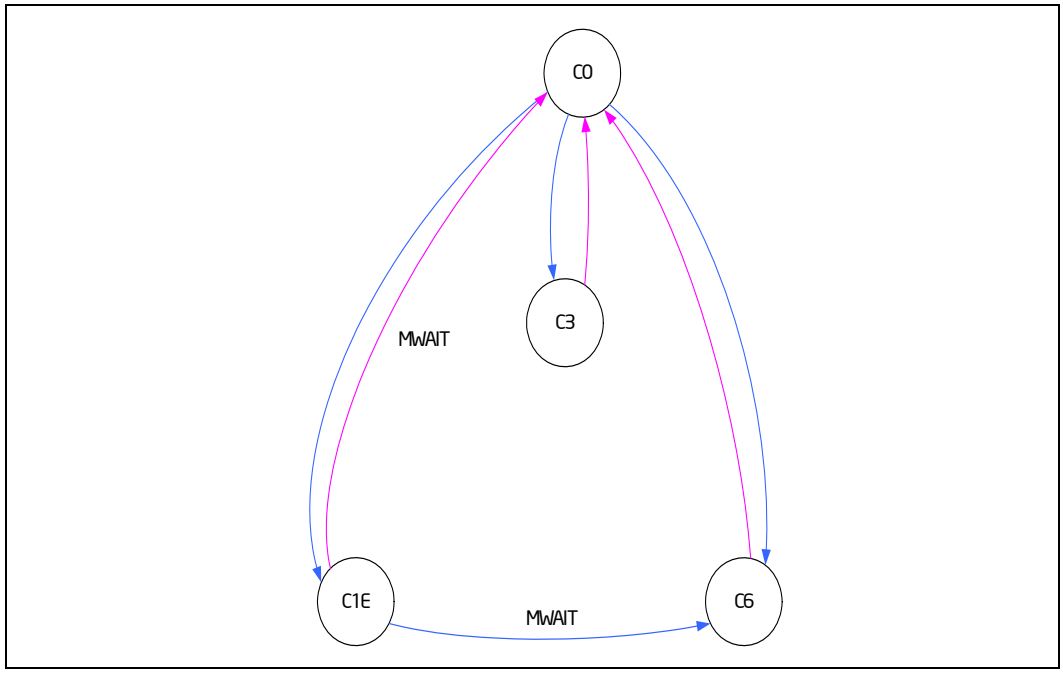
Table 123 shows package C-state resolution for a dual-core processor. Figure 70 summarizes package C-state transitions.

Table 123. Coordination of Core Power States at the Package Level

Package C-State		Core 1			
		C0	C1E[1]	C3	C6
Core 0	C0	C0	C0	C0	C0
	C1E[1]	C0	C1E[1]	C1E[1]	C1E[1]
	C3	C0	C1E[1]	C3	C3
	C6	C0	C1E[1]	C3	C6

Note:
1. If enabled, the package C-state will be C1E if all actives cores have resolved a core C1 state or higher.

Figure 70. Package C-State Entry and Exit



Note: The package C state resolves to the highest power dissipation C state of the cores.

8.2.5.1 Package C0

This is the normal operating state for the processor. The processor remains in the normal state when at least one of its cores is in the C0 or C1E state or when the platform has not granted permission to the processor to go into a low power state. Individual cores may be in lower power idle states while the package is in C0.

8.2.5.2 Package C1E

The Intel® Xeon® processor C5500/C3500 series supports the package C1E state.

8.2.5.3 Package C3 State

A processor enters the package C3 low power state when:

- At least one core is in the C3 state.
- The other cores are in a C3 or lower power state and the processor has been granted permission by the platform.
- The platform has not granted a request to a package C6 state but has allowed a package C6 state.

or

- All cores may be in C6 but the package may be in package C3, i.e. other socket of a DP system is in C3.

In the package C3-state, the LLC is snoopable.

8.2.5.4 Package C6 State

A processor enters the package C6 low power state when:

- All cores are in C6 and the processor has been granted permission by the platform.

In the package C6 state, all cores save their architectural state and have their core voltages reduced. The LLC is still powered and snoopable in this state. The processor remains in package C6 state as long as any part of the LLC is still active.

8.3 IMC Power Management

The main memory is power managed during normal operation and in low power ACPI Cx states.

8.3.1 Disabling Unused System Memory Outputs

Any system memory (SM) interface signal that goes to a memory module connector in which it is not connected to any actual memory devices, (such as an unpopulated or single-sided DIMM connector) is tri-stated. The benefits of disabling unused SM signals are:

- Reduced power consumption.
- Reduced possible overshoot/undershoot signal quality issues seen by the processor I/O buffer receivers caused by reflections from potentially un-terminated transmission lines.

When a given rank is not populated, as determined by the DRAM Rank Boundary register values, then the corresponding chip select and SCKE signals are not driven.

SCKE tri-state should be enabled by the BIOS where appropriate, since at reset all rows must be assumed to be populated.

8.3.2 DRAM Power Management and Initialization

The processor implements extensive support for power management on the SDRAM interface. There are four SDRAM operations associated with the Clock Enable (CKE) signals, which the SDRAM controller supports. The processor drives CKE pins to perform these operations.

8.3.2.1 Initialization Role of CKE

During power-up, CKE is the only input to the SDRAM whose level is recognized, other than the DDR3 reset pin, once power is applied. It must be driven LOW by the DDR controller to make sure the SDRAM components float DQ and DQS during power-up. CKE signals remain LOW while any reset is active, until the BIOS writes to a configuration register. With this method, CKE is guaranteed to remain inactive for longer than the specified 200 micro-seconds after power and clocks to SDRAM devices are stable.

8.3.2.2 Conditional Self-Refresh

Intel® Rapid Memory Power Management (Intel® RMPM), which conditionally places memory into self-refresh in the C3 and above states, is based on the state of the PCI Express links.

When entering the Suspend-to-RAM (STR) state, the processor core flushes pending cycles and then enters all SDRAM ranks into self-refresh. In STR, the CKE signals remain LOW so the SDRAM devices perform self-refresh.

The target behavior is to enter self-refresh for C3 and above states as long as there are no memory requests to service. The target usage is shown in Table 124.

Table 124. Targeted Memory State Conditions

Mode	Memory State
C0, C1E	Dynamic memory rank power down based on idle conditions.
C3, C6	Dynamic memory rank power down based on idle conditions If there are no memory requests, then enter self-refresh. Otherwise use dynamic memory rank power down based on idle conditions.
S1	S1 HP (high power - Intel® QPI in L1 is not supported): Dynamic memory rank power down based on idle conditions. S1 LP (low power - Intel® QPI in L1 supported): Dynamic memory rank power down based on idle conditions. If there are no memory requests, then enter self-refresh. Otherwise use dynamic memory rank power down based on idle conditions.
S3	Self Refresh Mode
S4	Memory power down (contents lost)
S5	Memory power down (contents lost)

8.3.2.3 Dynamic Power Down Operation

Dynamic power-down of memory is employed during normal operation. Based on idle conditions, a given memory rank may be powered down. The IMC implements aggressive CKE control to dynamically put the DRAM devices into a power-down state. The processor core controller can be configured to put the devices in *active power down* (CKE deassertion with open pages) or *precharge power down* (CKE deassertion with all pages closed). Precharge power down provides greater power savings but has a larger performance impact since all pages will be closed before putting the devices in power-down mode.

If dynamic power-down is enabled, then all ranks are powered up before doing a refresh cycle and all ranks are powered down at the end of refresh.

8.3.2.4 DRAM I/O Power Management

Unused signals shall be disabled to save power and reduce electromagnetic interference. This includes all signals associated with an unused memory channel. Clocks can be controlled on a per DIMM basis. Exceptions are made for per DIMM control signals such as CS#, CKE, and ODT for unpopulated DIMM slots.

The I/O buffer for an unused signal shall be tri-stated (output driver disabled), the input receiver (differential sense-amp) should be disabled, and any DLL circuitry related ONLY to unused signals shall be disabled. The input path must be gated to prevent spurious results due to noise on the unused signals (typically handled automatically when input receiver is disabled).

8.3.2.5 Asynch DRAM Self Refresh (ADR)

The Asynchronous DRAM Refresh (ADR) feature in the Intel® Xeon® processor C5500/C3500 series may be used to provide a mechanism to enable preservation of key data in DDR3 system memory. ADR uses an input pin to the processor, DDR_ADR, to trigger ADR entry. ADR entry places the DIMMs in self-refresh. Any data that is not committed to memory when ADR activates is lost, i.e. in-flight data to/from memory, caches, etc. are not preserved. In DP platforms, both processors need to be placed into ADR at approximately the same time to prevent spurious memory requests. Otherwise, a processor that is not in ADR may generate memory requests to the other processor's memory (in ADR).

The Intel® Xeon® processor C5500/C3500 series contains the following integrated ADR feature elements:

- Level-sensitive pin, DDR_ADR, that triggers DDR3 self-refresh entry.
- BIOS re-initialization of the memory controller triggers exit from DDR3 self-refresh.

A complete and robust memory backup implementation involves many areas of the platform, e.g. hardware, BIOS, firmware OS, application, etc. The ADR mechanism described in this section does not provide such a solution by itself. Although the ADR mechanism can be used to implement a battery backup memory, the usage as described in this section is focused on allowing rapid DDR3 self-refresh entry/exit to facilitate system recovery during re-boot by preserving critical portions of memory. It is assumed for the purposed of this section, that full power delivery is available uninterruptedly during the ADR sequence. Since internal caches, buffers, etc. are not committed to memory, the platform needs to implement protected software data structures as appropriate.

Warning: The simplified ADR application described in this chapter is different from the storage application of ADR. In the application described in this section, only data that is committed to DDR3 memory when ADR is invoked is preserved; there are no provisions for preservation of processor state, in-flight data, etc. In contrast, the storage application of the ADR provides for preservation of certain data that is not in DDR3 memory when ADR is invoked.

Additional restrictions are that ADR is not supported in S2/S3/S5 or C3/C6 ACPI states nor under memory controller RAS modes of sparing, lockstep, mirroring, x8, or while using unbuffered DIMMs. Continual (back-to-back) inbound reads or writes of the same location are not permitted.

ADR entry is quick (~20µS under non-throttling DDR3 conditions). In comparison, S3 entry takes significantly more time to drain the I/O and flush processor caches before putting the memory into self-refresh. ADR is primarily targeted for systems and thermal conditions in which the DDR3 does not throttle (assume closed-loop DDR3 monitor/control) because ADR entry time can increase significantly under DDR3 throttling conditions.

This document covers the ADR features integrated into the Intel® Xeon® processor C5500/C3500 series and provides an overview of key platform and system software requirements for a ADR solution.

8.3.2.5.1 Intel® Xeon® Processor C5500/C3500 Series ADR Use Model

The usage model is to allow preservation of memory contents with a fairly rapid entry/exit latency. Data preservation of DDR3 memory is accomplished by placing the DDR3 memory into self-refresh. Only data that is committed to the memory when ADR is invoked will be preserved. Other data, e.g. in flight data, interval processor context, etc. will be lost. It is the platform software's responsibility to implement appropriate data structures to ensure that the DDR3 data of interest is correctly preserved prior to using this data after ADR exit.

The key benefits of this usage model are rapid self-refresh entry/exit with low overhead. A typical application for the ADR feature usage is the preservation of a large, complex data structure that requires a relatively long time to create and may persist across re-boots. An example of such a data structure is a routing table. For the purposes of this usage model, it is assumed that full power delivery is sustained to the Intel® Xeon® processor C5500/C3500 series during the entire ADR envelope.

Warning: In DP platforms, both processors must be placed into ADR at approximately the same time.

8.3.2.5.2 Pin-Triggered Self-Refresh Entry

ADR provides an external pin, DDR_ADR, that places the DDR3 into self-refresh. The critical data, now all in the DDR3, can be preserved as long as power is maintained to the DIMMs in self-refresh.

The interface and sequence for placing DDR3 in self-refresh is part of the existing JEDEC DDR3 specification. DDR3 self-refresh entry involves commands being issued from the memory controller to the DDR3 ending in the CKE signals for each DIMM/RANK being driven low.

Pin-triggered ADR entry is initiated by a platform that has detected an abnormal condition requiring system reboot. (During this time, the platform sustains full power delivery to the Intel® Xeon® processor C5500/C3500 series.) Upon completion of ADR entry, the DDR3 is in self-refresh mode. The ADR trigger pin is disabled by default and therefore must be configured/enabled by the BIOS.

The platform must not set the DDR_ADR signal to the Intel® Xeon® processor C5500/C3500 series until the BIOS has configured the DDR3 memory. The BIOS needs to provide an indication that the memory is not yet configured to the platform. The BIOS may use one of the GPIO pins on the Intel® 3420 chipset for this purpose. In this case, the BIOS will program one of the GPIOs as an output. The BIOS will drive this GPIO active after the DDR3 memory configuration has been completed. The platform will use this GPIO to conditionalize the presentation of DDR_ADR to the processor, i.e. DDR_ADR will only be activated to the processor after the DDR3 memory is configured. The programming of this GPIO will be persistent after the initial power up. I.e. it will only be reset to the default after a power-down.

8.3.2.5.3 Power-On, Entry, and Exit Sequence

Three sequences are presented below: first power on, ADR entry, and ADR exit.

- First Power-ON Sequence
  - The BIOS initializes the memory controller. Right before enabling CKE, the BIOS redefines one of the Intel® 3420 chipset GPIO pins (pre-selected by the platform implementation) as an output and drives it active. The BIOS then scrubs all DRAM. The BIOS also writes the H[2:0]_REFRESH_THROTTLE_SUPPORT registers to arm the DDR_ADR pin to trigger the ADR entry sequence. (The arming BIOS writes must be timed such that the DDR3 is fully active - ~200 clocks after CKE rising.)
  - Once armed, an DDR_ADR event will put the DDR3 memory into self-refresh as described in the following section.
- ADR Entry Platform Sequence
  - The platform sets DDR_ADR to Intel® Xeon® processor C5500/C3500 series.
  - Intel® Xeon® processor C5500/C3500 series issues the command sequence for self-refresh entry to the DDR3 memory eventually ending in all CKEs=0.
  - The DDR3 memory enters and stays in self-refresh mode until the ADR exit sequence is performed.
  - The BIOS re-programs the selected GPIO to input mode and sets its state to the power-on default, i.e. inactive.

Warning: The Intel® Xeon® processor C5500/C3500 series will not respond to the DDR_ADR signal unless the ADR trigger enable bit is set (see the CHL_CR_REFRESH_THROTTLE_SUPPORT register description). The BIOS is expected to enable ADR triggering only after the DDR is fully enabled (~200 clocks post CKE rising per DDR3 spec).

- ADR Exit Sequence

— The BIOS initializes the memory controller. Just before enabling CKE the BIOS redefines (pre-selected by the platform implementation) one of the Intel® 3420 chipset GPIO pins as an output and drives it active. The BIOS also writes the H[2:0]_REFRESH_THROTTLE_SUPPORT registers to arm the DDR_ADR pin to trigger the ADR entry sequence. (The arming BIOS writes must be timed such that the DDR3 is fully active - ~200 clocks after CKE rising.) If an ADR event occurs after this point, it will result in an ADR trigger and ADR re-entry. Normal recovery proceeds with the BIOS restoring the memory controller settings from NVRAM The DDR3 is not initialized/scrubbed because it contains the preserved ADR data.

8.3.2.5.4 Non-Volatile Save/Restore of MCU Configuration

The first time a system boots (no valid data assumed in DIMM), the BIOS is expected to initialize and scrub memory. There is not sufficient time for software to save the memory controller (MCU) register contents during a triggered ADR entry into self-refresh sequence. Therefore, just like for S3, the MCU register settings must be stored to non-volatile memory (flash or battery backed NVRAM) on first boot. Upon exit from self-refresh, (just like S3 resume) the BIOS must make sure that it does not re-initialize or scrub memory but instead restores the memory controller contents and begins using the DDR3 memory (with knowledge of the memory space that it can overwrite and which space should be left untouched).

Since with ADR the system could have been asynchronously taken down, unlike normal S3 recovery, the OS cannot assume that its data structures in memory are valid and must boot from scratch. Further, the OS must be aware of which memory space was preserved, ascertain the integrity of this space (via its software protected data structure), and handle re-allocating the protected space to the application(s).

8.3.2.5.5 Target ADR Entry Time

After the DDR_ADR signal is asserted, the DDR3 self-refresh entry sequence is initiated by the Intel® Xeon® processor C5500/C3500 series memory controller (MCU). The end of this sequence is where the MCU drives the CKE pins low, thus causing the DDR3 DIMMs to enter self-refresh mode. The system must sustain in-spec power delivery to the processor rails continuously during ADR entry/exit and during the entire time that the platform is in ADR.

The Intel® Xeon® processor C5500/C3500 series ADR entry target is 20µS, assuming the DDR3 is operating in closed loop throttling mode and is not throttling. Open or closed loop throttling conditions will significantly increase ADR entry time.

Figure 71. DDR_ADR to Self-Refresh Entry

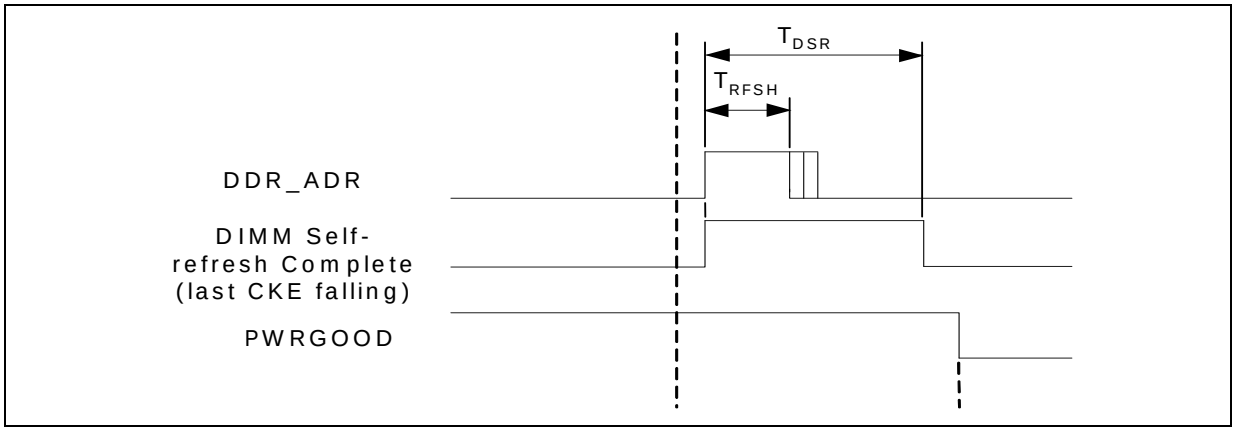


Table 125. ADR Self-Refresh Entry Timing - AC Characteristics (CMOS 1.5 V)

Symbol	Parameter	Min.	Typ.	Max.	Unit	Figure	Notes
T_{RFSH}	Time required to sample DDR_ADR input as active	8			Clock	Figure 71	1
T_{DSR}	Time required to complete DIMM self-refresh activation from DDR_ADR input assertion (last CKE falling)	20			µs	Figure 71	2

Notes:
1. Input is synchronized internally; no setup and hold times are required relative to clocks.
2. Assumes closed loop throttling mode and thermal conditions such that the DDR3 interface is not in a throttling mode.

8.4 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 (10 W for add-in edge-connected cards). When the BIOS updates the slot power limit register of the root ports within the IIO module, the IIO module will automatically transmit a Set_Slot_Power_Limit message with corresponding information to the attached device. It is the responsibility of the platform BIOS to properly configure the slot power limit registers in the IIO module and failure to do so may result in attached end-points remaining completely disabled to comply with the default power limitations associated with their form-factors.

8.4.1 DMI Power Management Rules for the IIO Module

1. The IIO module must send the ACK-Sx for the Go_S0, G0_S1_temp(final), GO_S1_RW, G0_S3/4/5 messages.
2. The IIO module is never permitted to send an ACK-Sx unless it has received one of the above Go-S* messages.
3. The IIO module is permitted to send the RESET-WARN-ACK message at any time after receiving the RESET-WARN message.

8.4.2 Support for P-States

The platform does not coordinate P-state transitions between CPU sockets with hardware messages. As such, the IIO module supports, but is uninvolved with, P-state transitions.

8.4.3 S0 -> S1 Transition

1. The OSPM performs the following functions:
   a. To enter an S state the OS will send a message to all drivers that a sleep event is occurring.
   b. When the drivers have finished handling their devices and completed all outstanding transactions, they each respond back to the OS.
   c. OSPM after this will:
      — Disable Interrupts (except SMI which is invisible to OS)
      — Set TPR (Task Priority Register) high.
      — Write the fake **SLP_EN,** which triggers BIOS (SMI Handler).
      — Set up ACPI registers in the PCH.
   d. Since the sleep routine in the OS was a call, the OSPM returns to the calling code and waits in a loop polling on the wake status (**WAK_STS)** bit (until S0 state is resumed). The Wake Status bit (PCH) can only be set by PCH after the PCH has

entered an S-state. It must be cleared by SW. If SW were to leave this bit asserted, then the CPU will attempt to go to Sx by writing the Sleep Enable bit, do the **RET**, read the Wake Status bit as **'1'** and continue through the code before the PMReq(S1) had been delivered. When the **PMReq(S1)** is delivered the CPU will be executing some code and get halted in the middle.

e. There will never be a C-state and S-state transition simultaneously. The OS code must never attempt to do a C-state transition after writing the Sleep Enable bit in the PCH. C states are only allowed in S0. Likewise S state requests must not be followed by **MWAIT**.

f. The BIOS writes the Sleep Type and Sleep Enable bits in the PCH, using IO write cycles. After this, last remaining thread (Monarch Thread) halts itself.

2. The PCH sends **Go_S1_Final** on DMI since S1 is final state desired by PCH.
3. On receiving **Go_S1_Final,** IIO multicasts **PMReq(S1)** over Intel® QPI to CPUs (for DP systems).
4. CPUs respond by **CmpD(S1)** and acknowledges the receipt. Since Interrupts have already been disabled, no interrupts will be received by CPU though normal read/write to memory may be received by uncore in S1 state.
   a. All cores are halted.
   b. After sending **CmpD(S1)**, uncore may try to bring Intel® QPI link to L1 if no activity is detected and queues are idle.
5. IIO module responds to **CmpD(S1)** from CPU by sending **Ack_Sx** to PCH over DMI.
6. IIO and PCH may transition the DMI link to L0s autonomously from this sequence when their respective active state L0s entry timers expire.

8.4.4 S1 -> S0 Transition

1. The PCH will get a wake event, such as interrupt, PBE (Pending Break Event), etc. that causes it to force the system back to S0. For the S1 to S0 return, there is handshake with internal agents so they know the system is in S0 again.
   a. PCH does all internal hand shakes before it sends **Go_S0** up the DMI.
2. The PCH generates **Go_S0** VDM.
3. In response to its reception of **Go_S0**, IIO module multicasts a **PMReq(S0)** message to all CPUs. Intel® QPI links may need to be brought back to L0 before the message/s can be sent.
4. After receiving the response from all CPUs (**CmpD(S0)**), the IIO module sends **Ack_Sx** Vendor Defined Message to PCH.

Note: The CPU has two modes of S1 states (low-power and high-power S1). In the low-power S1, the CPU shuts off its core PLLs when the Intel® QPI link transitions to L1 due to inactivity. Hence, it cannot respond to any other message such as VLW, including interrupts from the low power S1 mode. To wake up the platform to S0, the CPU must see a Go_S0 message issued first by the PCH before anything else.

8.4.5 S0 -> S3/S4/S5 Transition

The universe comprehended by the DMI specification consists of a single IIO and a single PCH. It does not comprehend multiple IIO modules and PCHs.

In the S3 sleep state, the system context is maintained in memory. The IIO module, DMI link and all standard PCI Express links will transition to L3 Ready before power is removed, which then places the link in L3.

8.5 PCIe Power Management

The IIO module supports the following link/device states and events:

- L0s as receiver and transmitter.
- L1 link state.
- ASPM L1 link state.
- L3 link state.
- MSI or GPE event on power manage events internally generated (on a PCI Express port hotplug event) or received from PCI Express.
- D0 and D3 hot states on a PCI Express port.
- Wake from D3-hot on a hot plug event at a PCI Express port.

The IIO module does not support the following link states or events:

- No support L1a.
- No support L2 (i.e. no aux power to IIO module).
- No support for the in-band beacon on PCI Express link.

8.5.1 Power Management Messages

When the Intel® Xeon® processor C5500/C3500 series receives PM_PME messages on its PCI Express port, including any internally generated PM_PME messages on a hotplug event at a root port, it either propagates it to the PCH over the DMI link as an Assert/De-assert_PMEGPE message or generates an MSI interrupt or generates Assert/De-assert_Intx message. See the *PCI Express Base Specification,* Revision 1.1 for details of when a root port internally generates PM_PME message on a hotplug event. When the 'Enable ACPI mode for PM' Miscellaneous Control and Status Register (MISCCTRLSTS) bit is set, GPE messages are used for conveying PM events on PCI Express, otherwise MSI or INTx is generated.

The rules for GPE messages are similar to the standard PCI Express rules for Assert_INTx and De-assert_INTx:

- Conceptually, the Assert_PMEGPE and De-assert_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 IIO communicates this change to the PCH using the appropriate Assert_PMEGPE or de-assert_PMEGPE messages.

Note: Duplicate Assert_PMEGPE and De-assert_PMEGPE messages have no affect, but are not errors.

- The IIO tracks the state of the virtual wire on each port independently and presents a "collapsed" version (Wire-OR'ed) of the virtual wires to the PCH.

See the IIO interrupts section for details of how these messages are routed to the legacy PCH.

8.6 DMI Power Management

- Active power management support using L0/L0s/L1a state.
- All inputs and outputs disabled in L3 Ready state.

See Section 8.1.10, "Supported DMI Power States" for details.

8.7 Intel® QPI Power Management

- L0 – Full performance, full power.
- L1 – Turn off the link, longer latency back to L0.

Note: There is no L0s support in the internal Intel® QPI link.

8.8 Intel® QuickData Technology Power Management

The Intel® Xeon® processor C5500/C3500 series implements with Intel® QuickData Technology support different device power states. The Intel® QuickData Technology device supports the D0 device power state that corresponds to the fully-on state and a pseudo D3 hot 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 power management capable operating system does not put the Intel® QuickData Technology device into a lower device power (e.g. D3) state while its client I/O device is fully powered on (i.e. 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.

8.8.1 Power Management w/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 (e.g. D3) state while any of its client I/O devices are fully powered on (D0) and actively using Intel® QuickData Technology . E.g., the operating system might attempt to transition the device to D3 while placing the system into the S4 (hibernate) system power state. In that process, it must not 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.

§ §

9.0 Thermal Management

For thermal specifications and design guidelines, see the *Intel® Xeon® Processor C5500/C3500 Series Thermal Mechanical Design Guide*.

§ §

10.0 Reset

10.1 Introduction

This chapter describes specific aspects of various hardware resets.

10.1.1 Types of Reset

These are the types of reset:

- Power Good Reset

  Power good reset is invoked by the de-assertion of the VCCPWRGOOD signal and is part of power-up reset. This reset clears sticky bits, clears all system states, and downloads fuses. Power-good reset destroys program state, can corrupt memory contents, and destroys error logs.

- Warm Reset

  Warm reset is invoked by the assertion of the PLTRST# signal and is part of both the power-up and power good reset sequences. Warm reset is the “normal” component reset, with relatively short latency and fewer side-effects than power good reset. It preserves sticky bits (e.g. error logs and power-on configuration). Warm reset destroys the program state and can corrupt memory contents, so it should only be used as a means to un-hang a system while preserving error logs that might provide a trail to the fault that caused the hang. Warm reset can be initiated by code running on a processor, SMBus, or PCI agents. Warm reset is not guaranteed to correct all illegal configurations or malfunctions. Software can configure sticky bits in the IIO to disable interfaces that will not be accessible after a warm reset. Signaling errors or protocol violations prior to reset (from Intel® QPI, DMI, or PCI-Express) may hang interfaces that are not cleared by a warm reset.

- PCI Express* Reset

  A 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.

- SMBus Reset

  An SMBus reset resets only the slave SMBus controller. The slave SMBus controller consists of the protocol engine and SMBus-specific “data state,” such as the command stack. An SMBus reset does not reset any state that is observable through any other interface into the component.

- CPU Only Reset (also known as CPU warm reset)

  Software can reset the processing cores and uncore independant of the IIO by setting the IIO.SYRE.CPURESET bit. The BIOS uses this for changing Intel® QPI frequency.

10.1.2 Trigger, Type, and Domain Association

Table 126 indicates which core reset domains are affected by each reset type, and which reset triggers initiated each reset type.

Table 126. Core Trigger, Type, Domain Association

		Reset Domain															
Reset Trigger	Reset Type	PLL VCOs	Arrays	Straps sampled	Fuses sampled	Analog I/O Compensation	SMBus Protocol Engine	Sticky configuration Bits	SYRE.SAVCFG, QPILCL.1, Configuration Bit	Tri-statable Outputs	Fuse Downloader	Array Initialization Engines	Misc. State Machines	PCI Express Logic	QPI Link Logic Layer	DMI Logic	Internal CPU Reset (RESETO_N) Signal
COREPWRGOOD signal de-assertion	Power good		x	x	x		x	x	x	x	x	x	x	x	x	x	x
PLTRST# assertion	Warm						x		x				x	x	x	x	x
Receive Link Initialization Packet	Link QPI														x		
IIO.BCTRL.Secondary Bus Reset	PCI Express													x			
SMBus protocol	SMBus						x										
IIO.SYRE.CPURESET	CPU Warm																x

10.2 Node ID Configuration

A dual-socket Intel® Xeon® processor C5500/C3500 series system (see Figure 72) requires a single PCH to be connected to the system. The processor CPU that has the PCH connected to the DMI port is referred to as the legacy CPU. The DMI port on the other processor is unused and is referred to as the non-legacy CPU.

A dual socket Intel® Xeon® processor C5500/C3500 series system requires four Intel® QPI node IDs - two for the integrated processor modules (one on each processor) and two for the integrated IO modules (one on each processor). Thus, each processor socket is assigned two Intel® QPI node IDs.

The node ID assignment is made based on the DMI_PE_CFG# pin.

The DMI_PE_CFG# strap indicates whether the PCH is connected to CPU socket or not. The Intel® Xeon® processor C5500/C3500 series CPU that connects to the PCH will be the legacy CPU and the DMI_PE_CFG# pin will be true for that socket.

The following node IDs will be used by platform:

- 000: Legacy IIO (IIO connected to the PCH)
- 001: Legacy CPU/uncore
- 010: Non-Legacy CPU/uncore
- 100: Non-Legacy IIO (not connected to PCH)

The Intel® Xeon® processor C5500/C3500 series will support either the legacy or the non-legacy CPU being the boot processor. Selection of the boot processor is controlled by the BIOS.

The Legacy IIO is always the firmware agent and either of the processors can fetch code from the flash. The processors may then use a semaphore register in the IIO to determine which processor is designated as the boot processor.

10.3 CPU-Only Reset

The BIOS typically requires a CPU-only reset for several functions, such as configuring the CPU speed. This CPU-only reset occurs after the platform cold reset. If all CPUs (one socket and two socket configurations) in the system are connected directly to the IIO, then the flow for the CPU-only reset is straightforward and as described below.

To set the core frequency correctly, each socket BSP writes the range of supported frequencies in an IIO scratch pad register (PLATFORM_INFO MSR). Each node BSP reads the values written by the other node and computes the common frequency. Since both BSPs use the same algorithm, both arrive at the same least common frequency feasible. Each node BSP then updates its own FLEX_RATIO_MSR. Other conditions that require CPU-only reset are handled in a similar fashion and the appropriate MSRs are set at this point. A CPU-only reset is required for all of the new setting to take effect. The legacy BSP sets the IIO.SYRE.CPURESET bit to force a CPU warm reset. In response to setting the IIO.SYRE.CPURESET bit, the IIO asserts the internal CPU Reset (RESETO_N) signal to warm reset the CPU only. Since each socket has its own IIO with its own internal reset (RESETO_N) signal, the IIO drives the internal reset signal to the socket to force the CPU-only reset deterministically.

In a dual-socket Intel® Xeon® processor C5500/C3500 series, when the system BSP is ready for a CPU-only reset it follows the sequence:

1. Sets the IIO.SYRE.NL_SYNC_RESET_CPU_ONLY bit in the non-legacy IIO
2. Then sets the IIO.SYRE.CPURESET bit in the legacy IIO.

When the IIO.SYRE.CPURESET bit is set in the legacy IIO, the legacy IIO must ensure that its own RESETO_N and the RESETO_N on the non-legacy Intel® Xeon® processor C5500/C3500 series are asserted deterministically. To achieve this, the Legacy IIO drives DP_SYNCRST# to the non-legacy IIO. This is the same pin used during initial cold reset.

The non-legacy IIO samples DP_SYNCRST# asserted and distinguishes between a CPU-only reset and all other reset conditions (power-on, powergood etc) by using the IIO.SYRE.NL_SYNC_RESET_CPU_ONLY bit.

- If IIO.SYRE.NL_SYNC_RESET_CPU_ONLY bit is clear, then the DP_SYNCRST# assertion is a cold reset or a warm reset, the non-legacy IIO gets reset and then RESETO_N is asserted to the non-legacy CPU
- If IIO.SYRE.NL_SYNC_RESET_CPU_ONLY bit is set, then the DP_SYNCRST# assertion is for a CPU-only warm reset. The non-legacy IIO is not reset. The non-legacy IIO drives the RESETO_N to the non-legacy core complex.

This flow ensures that the RESETO_N to the legacy CPU is asserted at a known fixed offset w.r.t. the cycle on which RESETO_N is asserted to the non-legacy CPU. The 96-cycle RESET de-assertion heartbeat ensures determinism.

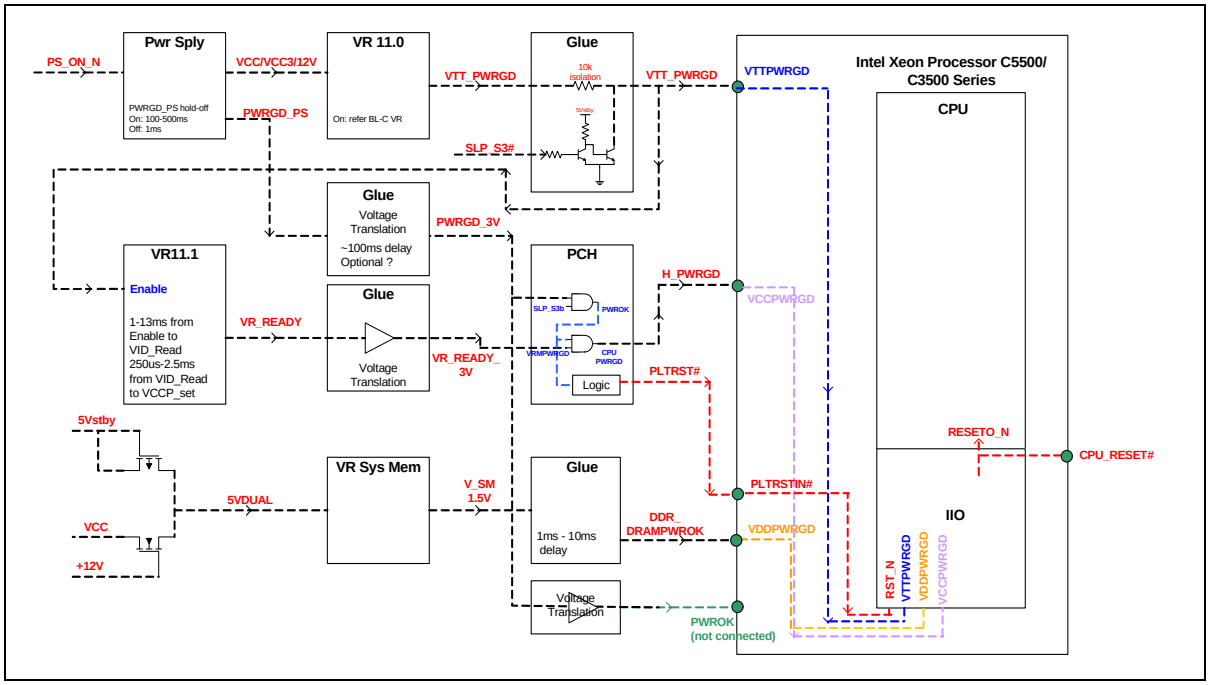
10.4 Reset Timing Diagrams

For clarification, the different voltages used in the system are:

- VCC = Ungated power to core.
- VTT = Ungated power to uncore, IIO.
- VDD = Dram power.

See the following figure.

Figure 72. Intel® Xeon® Processor C5500/C3500 Series System Diagram



10.4.1 Cold Reset, CPU-Only Reset Timing Sequences

The PCH asynchronously deasserts PLTRST#. On the Intel® Xeon® processor C5500/C3500 series, this PLTRST# deassertion is synchronized by the legacy processor and sent to the non-legacy processor using the DP_SYNCRST# pin.

When the BIOS writes the IIO.SYRE.CPURESET bit (in legacy Intel® Xeon® processor C5500/C3500 series) and triggers a CPU only Reset, the legacy IIO will ensure that its own internal RESETO_N and the RESETO_N on the non-legacy processor are deasserted deterministically.

10.4.2 Miscellaneous Requirements and Limitations

- Power rails and stable QPICLK and PECLK master clocks remain within specifications through all but power-up reset.
- Frequencies described in this chapter are nominal.
- Warm reset can be initiated by code running on a processor, SMBus, or PCI agents.
- Warm reset is not guaranteed to correct all illegal configurations or malfunctions. Software can configure sticky bits in the IIO to disable interfaces that will not be accessible after a warm reset. Signaling errors or protocol violations prior to reset

(from Intel® QPI, DMI, or PCI-Express) may hang interfaces that are not cleared by a warm reset.

- System activity is initiated by a request from a processor link. No I/O devices will initiate requests until configured by a processor to do so.

The requirements for DDR_DRAMPWROK assertion are:

- Signal must be monotonic.
- 100 ns minimum delay between VDDQ @ 1.425 V to DDR_DRAMPWROK @ Vihminspec (0.627 V).
- DDR_DRAMPWROK must be asserted no later than VCCPWRGOOD assertion.
- No relationship between DDR_DRAMPWROK and VccP ramp.

§ §

11.0 Reliability, Availability, Serviceability (RAS)

11.1 IIO RAS Overview

This chapter describes the features provided by the Intel® Xeon® processor C5500/C3500 series IIO module for the development of high RAS (Reliability, Availability, Serviceability) systems. RAS refers to three main features associated with system's robustness. These features are summarized as:

- **Reliability:** How often errors occur, and whether the system can recover from an error condition.
- **Availability:** How flexible the system resources can be allocated or redistributed for the system utilizations and system recovery from errors.
- **Serviceability:** How well the system reports and handles events related to error, power management, and hot plug.

IIO RAS features aim to achieve the following:

- Soft, uncorrectable error detection (Intel® QPI, PCIe) and recovery (PCIe) on links. CRC is used for error detection (Intel® QPI, PCIe), and error recovered by packet retry (PCIe).
- Clearly identify non-fatal errors whenever possible and minimize 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 the software to contain and recover from errors.
  - — Error logging/reporting to quickly identify failures, contain and recover from errors.
- PCIe hot add/remove to provide better serviceability.

The processor IIO RAS features can be divided into five categories. These features are summarized below and detailed in the subsequent sections:

1. **System level RAS**
   - — Platform or system level RAS for inband and outband system management features.
   - — On-line hot add/remove for serviceability.
   - — Memory mirroring, and sparing for memory protection.
2. **IIO RAS**
   - — IIO RAS features for error protection, logging, detection and reporting.
3. **Intel® QuickPath Interconnect RAS**
   - — Standard Intel® QuickPath Interconnect RAS features as specified in the Intel® QuickPath Interconnect specification.
4. **PCI Express RAS**
   - — Standard PCIe RAS features as specified in the PCIe specification.
5. **Hot Add/Remove**
   - — PCIe hot plug/remove support.

11.2 System Level RAS

11.2.1 Inband System Management

Inband system management is accomplished by firmware running in high privileged mode (SMM) and accessing system configuration registers for system event services. In the event of error, fault, or hot add/remove, firmware is required to determine the system condition and service the event accordingly. Firmware may enter SMM mode for these events, so that it has the privilege to access the OS invisible configuration registers.

11.2.2 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, are assumed to be secured and have the right to access all registers within a component.

SMBus connected globally to CPUs, IIOs, and PCHs — through a common shared bus hierarchy for SMBus. By using the outband signals, an outband agent can handle events like hot plug or error recovery. Outband signals provide the BMC with a global path to access the CSRs in the system components, even when the CSRs become inaccessible to CPUs through the inband mechanisms.The SMBus is mastered by the Baseboard Management Controller (BMC) by a platform-specific mechanism.

To support outband system management, the IIO provides SMBus interface with access to the configuration registers in the IIO itself or in the downstream IO devices (PCICFG).

11.3 IIO Error Reporting

The IIO logs and reports the 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 CPU or an outband signaling to the platform.

In the case of inband messaging, the CPU is notified of the error by the inband message (interrupt, failed response, etc.). The CPU responds to the inband message and takes the appropriate action to handle the error.

Outband signaling (error pins) informs an external agent of the error events. An external agent, such as BMC, may collect the errors from the error pins to determine the health of the system and sends interrupts to CPU accordingly. In some cases of severe errors, when the system is no longer responding to inband messages, the outband signalling provides a way to notify the outband system manager of the error. The system manager can then perform system reset to recover the system functionality.

The IIO detects errors from the PCIe link, DMI link, Intel® QuickPath Interconnect link, or IIO core itself. An error is first logged and mapped to an error severity, and then mapped to a system event(s) for error reporting.

IIO error report features are summarized below and detailed in the following sections:

- Detect and logs Coherency Interface, PCIe/DMI, Intel® QuickData Technology DMA and IIO 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 severity.
- Allows flexible mapping of the error severity to different report mechanisms.
- Supports PCIe error reporting mechanism.

11.3.1 Error Severity Classification

Errors are classified into three severities in the IIO: Correctable, Uncorrectable, and Fatal. This classification separates those errors resulting in functional failures from those errors resulting in degraded performance. In the IIO, each severity can trigger a system event according to the mapping defined by the error severity register. This mechanism provides the software with the flexibility to map an error to the suitable error severity. For example, a platform might choose to respond to a uncorrectable ECC error 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 129. The software/firmware can choose to alter the default mapping after power-on.

11.3.1.1 Correctable Errors (Severity 0 Error)

Hardware correctable errors include those error conditions in which the system can recover without loss of information. Hardware corrects these errors and no software intervention is required. For example, a Link CRC error that is corrected by Data Link Level Retry is considered a correctable error.

— Error is corrected by the hardware without software intervention. System operation may be degraded but its functionality is not compromised.

— Correctable error may be logged and reported in a implementation specific manner:

Upon the immediate detection of the correctable error, or

Upon the accumulation of errors reaching to a threshold.

11.3.1.2 Recoverable Errors (Severity 1 Error)

Recoverable errors are software-correctable or software/hardware-uncorrectable errors that cause a particular transaction to be unreliable but the system hardware is otherwise fully functional. Isolating recoverable from fatal errors provides system management software with 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 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 CPU.

— OS/Firmware takes the action to contain the error.

11.3.1.2.1 Software Correctable Errors

Software correctable errors are considered as “recoverable” error. These errors include 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 CPU.

— Firmware or other system software layers take corrective actions.

— Data integrity is not compromised with such errors.

11.3.1.3 Fatal Errors (Severity 2 Error)

Fatal errors are uncorrectable error conditions which render the IIO hardware unreliable. For fatal error, inband reporting to the CPU is still possible. A reset might be required to return to reliable operation.

— System integrity is compromised and continued operation may not be possible.

— System interface is compromised.

— Inband reporting may be possible.
e.g. Uncorrectable tag error in cache, or Permanent PCIe link failure.

— Requires immediate logging and reporting of the error to CPU or legacy IIO.

11.3.2 Inband Error Reporting

Inband error reporting signals the system of a detected error via inband cycles. There are two complementary inband mechanisms in the IIO. The first mechanism is synchronous reporting along with transaction responses/completion. The second mechanism is asynchronous reporting of inband error message or interrupt. These mechanisms are summarized as follows:

Synchronous inband error reporting:

- Reported through the transaction. Data Poison bit indication.
  — Generally for uncorrectable data errors (e.g. uncorrectable data ECC error).
- Response status field in response header.
  — Generally for uncorrectable error related to a transaction (e.g. 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 become unreliable. The IIO silently drops the transaction. The requester will eventually time out and report an error.

Asynchronous inband error reporting:

- Reported through inband error or interrupt messages. A detected error triggers an inband message to the legacy IIO or CPU.
- Errors are mapped to three error severities.
- Each severity can generates one of the following inband message.
  — CPEI
  — NMI
  — SMI
  — None
- Each error severity can also cause Error pin (ERR[2:0]) assertion in addition to the above inband message.
- Fatal severity can cause viral in addition to the above inband message and error pin assertion.

Note: The Intel® Xeon® processor C5500/C3500 series does not support viral alert generation.

- IIO PCIe root ports can generate MSI, or forward MSI/INTx from downstream devices as per the PCIe specification.

11.3.2.1 Synchronous Inband 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. Synchronous reporting includes the following.

11.3.2.1.1 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 the request was completed without an error. Conversely, a “failed” status denotes that an error has occurred as the result of processing the request.

11.3.2.1.2 No Response

For errors that have corrupted the requester’s information (e.g. requester/source ID in the header), the IIO will not send a response to the requester. This will eventually cause the requester to time-out and trigger an error at the requester.

11.3.2.1.3 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 a “poisoned” indication. Both PCIe 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 the posted requests. Requests that require completion with data can also indicate poisoned data.

Since the IIO can be programmed to signal (interrupt or error pin) the detection of the 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 — the one that poisoned the data.

In general, the IIO 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.

11.3.2.1.4 Time-out

A 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, etc. In the IIO, if a transaction failed to complete within the time-out value, then an error is logged to indicate the failure.

Software has the option to either enable or disable the signaling (via error pin or interrupt) of the time-out error. On a forwarded transaction for Intel® QuickPath Interconnect or PCIe, the transaction is completed with a completer abort (PCIe) response status. On IIO-initiated transactions (such as DMA or interrupts), the IIO drops the transaction. Depending on the cause of the error, the fail/time-out response may be elevated to a fatal error, resulting in system/partition reset.

11.3.2.2 Asynchronous Error Reporting

Asynchronous error reporting is used to signal the system of detected errors. For errors that require immediate attention, errors not associated with a transaction, or error events requiring system handling, an asynchronous report is used. Asynchronous error reporting is controlled through the IIO error registers. These registers enable the IIO to report various errors via system events (e.g., SMI, CPEI, etc.). In addition, the IIO provides standard sets of error registers as specified in PCIe specification.

IIO error registers provide software with the flexibility to map an error to one of three error severities. Software associates each error severity with one of the supported inband messages or be disabled for inband messaging. The error pin assertion is also enabled/disabled for each error severity. Upon detection of a given error severity, associated events are triggered, which conveys the error indication through inband and/or outband signalling. Asynchronous error reporting methods are described as follows.

11.3.2.2.1 NMI (Non-Maskable Interrupt)

In past platforms, NMI reported fatal error conditions, typically through PCH component SERR mapping. Since the IIO provides direct mapping of an error to NMI, SERR reporting is obsolete. When an error triggers an NMI, the IIO broadcasts an NMI virtual legacy wire cycle to the CPUs. The PCH reports the NMI through assertion of the NMI pin. The IIO converts the NMI pin assertion to the Intel® QuickPath Interconnect legacy wire cycle on behalf of the PCH.

11.3.2.2.2 CPEI (Correctable Platform Event Interrupt)

CPEI is associated with a PCH component, programmed interrupt vector. When CPEI is needed for error reporting, the non-legacy IIO is configured to send the CPEI message to the legacy IIO. The message converts in the legacy IIO to Error[2:0] pin assertion conveying the CPEI event when enabled. As a result, the PCH sends a CPU interrupt with the specific interrupt vector and type defined for CPEI.

11.3.2.2.3 SMI (System Management Interrupt)

SMI reports fatal and recoverable error conditions. When an error triggers an SMI, the IIO broadcasts a SMI legacy wire cycle to the CPUs.

11.3.2.2.4 None

The IIO provides the flexibility to disable inband messages on the detection of an error. By disabling the inband messages and enable error pins, the IIO can be configured to report the errors exclusively via error pins.

11.3.2.2.5 Error Pins

The IIO contains three open-drain error pins for the purpose of error reporting — one pin for each error severity. The error pin can be used in a certain class of platforms to indicate various error conditions and can also be used when no other reporting mechanism is appropriate. For example, an error signal can be used to indicate error conditions (even hardware correctable error conditions) that may require error pin assertion to notify outband components, such as the BMC.

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, then the BMC performs steps to return the system to a functional state. However, if the system is unresponsive, then the outband agent can assert reset to force the system back to a functional state.

The IIO allows the 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 error pin assertion, the corresponding error pin is asserted. Software must clear the error pin assertion by clearing the global error status. 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.

The error pins are asynchronous signals.

11.3.2.2.6 PCIe INTx and MSI

PCIe INTx and MSI are supported through the PCIe standard error reporting. The IIO forwards the MSI and INTx generated downstream to the Coherency Interface. The IIO PCIe ports themselves generate MSI interrupt for error reporting if enabled. See the PCIe specification for more details on the PCIe standard and advance error capability.

11.3.2.2.7 PCIe/DMI "Stop and Scream"

There is a enable bit per PCIe port that controls "stop and scream" mode. In this mode the desire is to disallow sending 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/DMI devices that have ignored the poison bit and committed the data that can corrupt the I/O device.

11.3.2.2.8 PCIe "Live Error Recovery"

PCI Express ports support the Live Error Recover (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 down the associated link and automatically trains the link up.

11.3.3 IIO Error Registers Overview

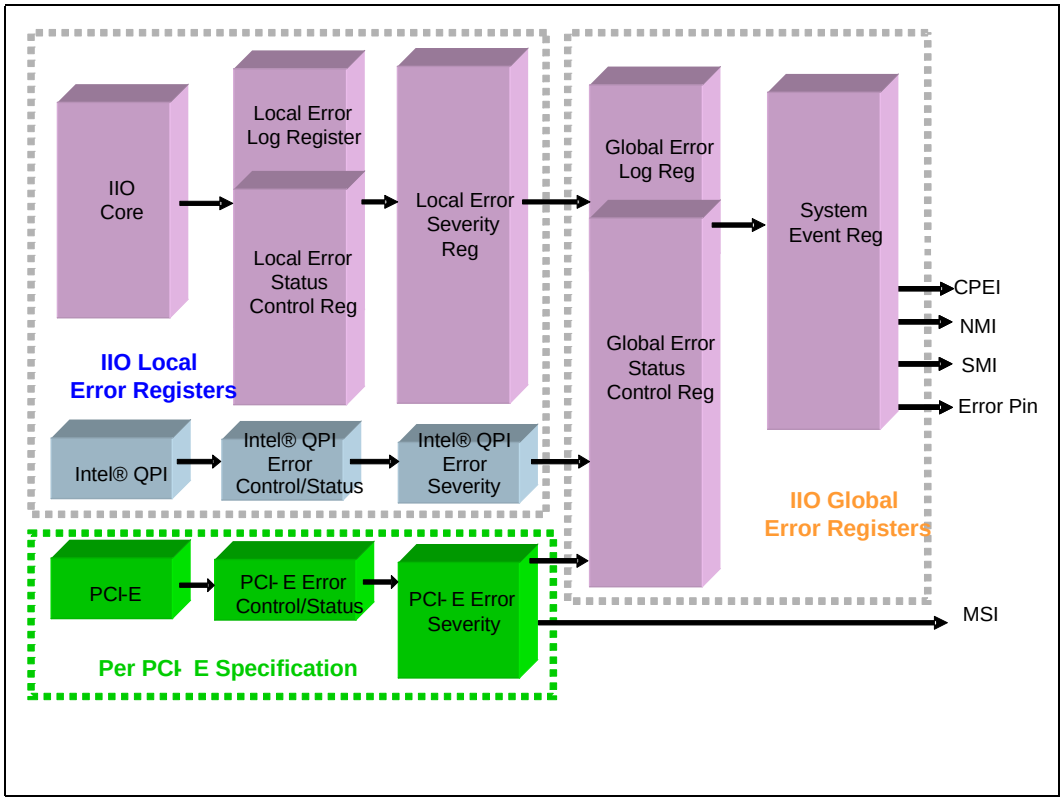
The IIO contains a set of error registers (Device 8, Function 2) to support error reporting.

- Global Error registers
- Local Error registers
- IIO System Control Status registers

These error registers are assumed to be sticky unless specified otherwise. 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 IIO local clusters (e.g. PCIe, DMI, Intel® QuickPath Interconnect, DMA, and IIO core logic). The global error registers collect the errors reported by the local error registers and map them to system events. Figure 73 illustrates the high level view of the IIO error registers. Figure 74 through Figure 79 illustrate the function of each error register.

Figure 73. IIO Error Registers



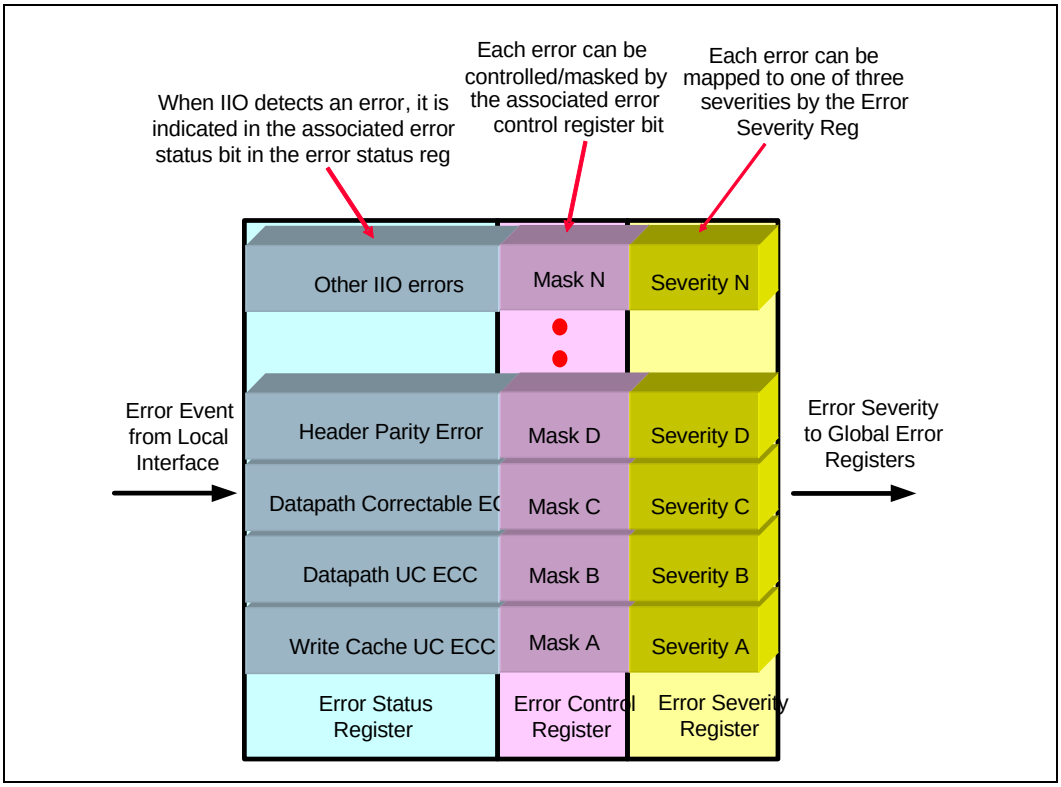
11.3.3.1 Local Error Registers

Each IIO local interface contains a set of local error registers. PCIe ports (including DMI) local error registers are defined per the PCIe specification. The Intel® QuickData Technology DMA has a predefined set of error registers. See the PCIe specifications for more details.

Since Intel® QuickPath Interconnect has not defined a set of standard error registers, the IIO has defined the error registers for the Intel® QuickPath Interconnect port using the same error control and report mechanism as the IIO core. This is described as follows:

- **IIO Local Error Status Register**
  The IIO core provides the local error status register for the errors associated with the IIO component itself. When a specific error occurred in the IIO core, its corresponding bit in the error status register is set. Each error can be individually masked by the error control register.
- **IIO Local Error Control (Mask) Register**
  The IIO core provides the local error control/mask register for the errors associated with the IIO component itself. Each error detected by the local error status register can be individually masked by the error control register. If an error is masked, the corresponding status bit will not be set for any subsequent detected error. The error control register is non-sticky and is cleared upon hard reset (all errors are masked). Figure 74 illustrates the IIO core Error Control/Status Register.

Figure 74. IIO Core Local Error Status, Control and Severity Registers



- **Local Error Severity Register**
  The IIO core provides a local error severity register for the errors associated with the IIO core itself. IIO internal errors can be mapped to three error severity levels. Intel® QuickPath Interconnect and PCIe error severities are mapped according to Table 128.
- **Local Error Log Register**
  The IIO core provides a local error log register for errors associated with the IIO component itself. When the IIO detects an error, the information related to the error is stored in the log register.
  IIO core errors are first separated into Fatal and Non-Fatal (Correctable, Recoverable) categories. Each category contains two sets of log registers: FERR and NERR. FERR logs the first occurrence of an error and NERR logs the subsequent occurrences. However, NERR does not log header/ address or ECC syndrome. FERR/NERR does not log a masked error. 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 the software. The **ERRST registers are only cleared by writing to the.corresponding local error status register. For example, clearing bit 0 in QPIPERRST0 clear the bit in this register as well as bit 0 in: QPIPFFERRST0, QPIPFNERRST0, QPINFERRST0, QPINNERRST0.

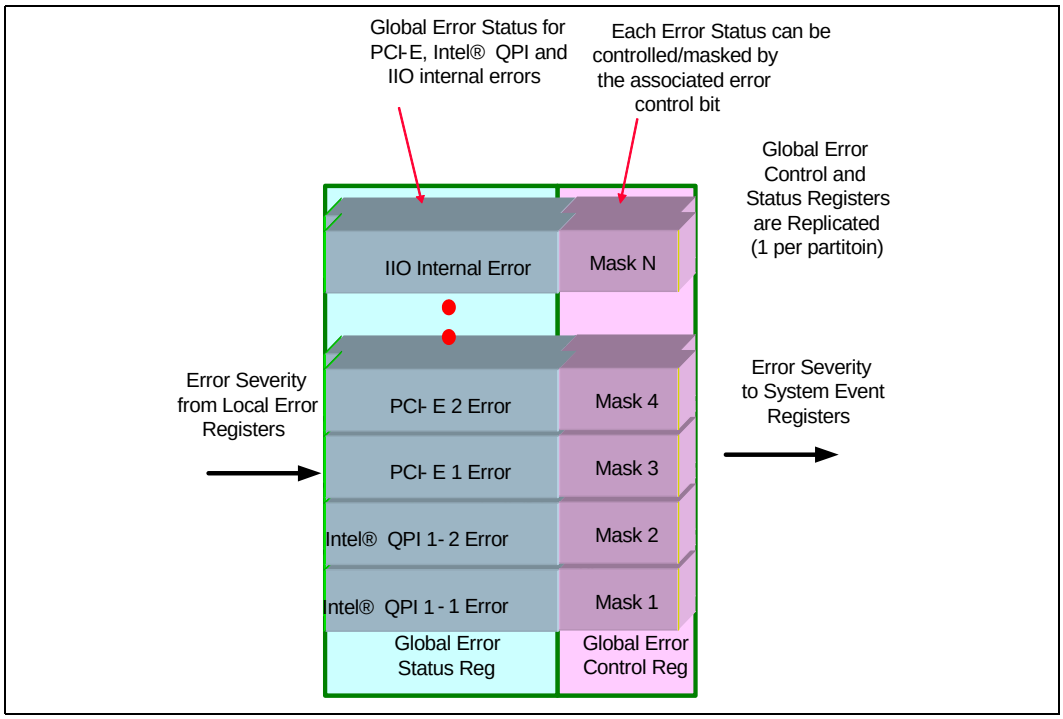
11.3.3.2 Global Error Registers

Global error registers collect errors reported by local interface and convert the errors to system events.

- **Global Error Control / Status Register**
  The IIO provides two global error status register to collect errors reported by the IIO clusters: Global Fatal Error Status and Global Non-fatal Error Status. Each register has an identical format that each bit in the register represents the fatal or non-fatal error reported by its associated interface: the Intel® QuickPath Interconnect port, PCIe port, DMA, or IIO core logic.

Local clusters, maps detected errors to three error severities and report them to global error logic. These errors are sorted into Fatal and Non-fatal and reported to respective global error status register, with severity 2 as fatal, and 0 & 1 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. Software clears the error bit by writing 1 to the bit. Each error is individually masked by global error control registers. If an error is masked, the corresponding status bit is not set for any subsequent reported error. The global error control register is non-sticky and cleared by reset.

Figure 75. IIO Global Error Control/Status Register



- **Global Log Registers**

  The global error log registers log the errors reported by the IIO clusters. Local clusters map the detected errors to three error severities and report them to the global error logic. The three error severities are divided into fatal and non-fatal errors that are logged separately by the FERR and NERR registers. Each bit in the FERR/NERR register is associated with an specific interface/cluster (e.g. a PCIe port). Each bit can be individually cleared by writing 1 to the bit. FERR logs the first report of an error, while NERR logs the subsequent reports of other errors. The time stamp log for the FERR and ERR provides the time of when the error was logged. Software can read this register to find out which of the local interfaces have reported the error. FERR log remains valid and unchanged from the first error detection until the clearing of the corresponding error bit in the FERR by the software.

- **Global System Event Register**

  Errors collected by the global error registers are mapped to system events. The system event status bit reflects the OR 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 transitions 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 76.

  Each severity type can be associated with one of the system events: SMI, CPEI, or NMI. In addition, the error pin registers allow error pin assertion for an error. When an error is reported to the IIO, the IIO uses the severity level associated with the error to look up the system event that should be sent to the system. For example, error severity 2 may be mapped to SMI with error[2] pin enabled. If an error with severity level 2 is reported and logged by the Global Log Register,

then an SMI is dispatched to the CPU and IIO error[2] is asserted. The CPU 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 register are initialized to their default values. The default mapping of severity and system event is set to be consistent with Table 129. Firmware can choose to use the default values or modify the mapping according to the system requirements.

The system event control register is non-sticky register that is cleared by hard reset.

Figure 76. IIO System Event Register

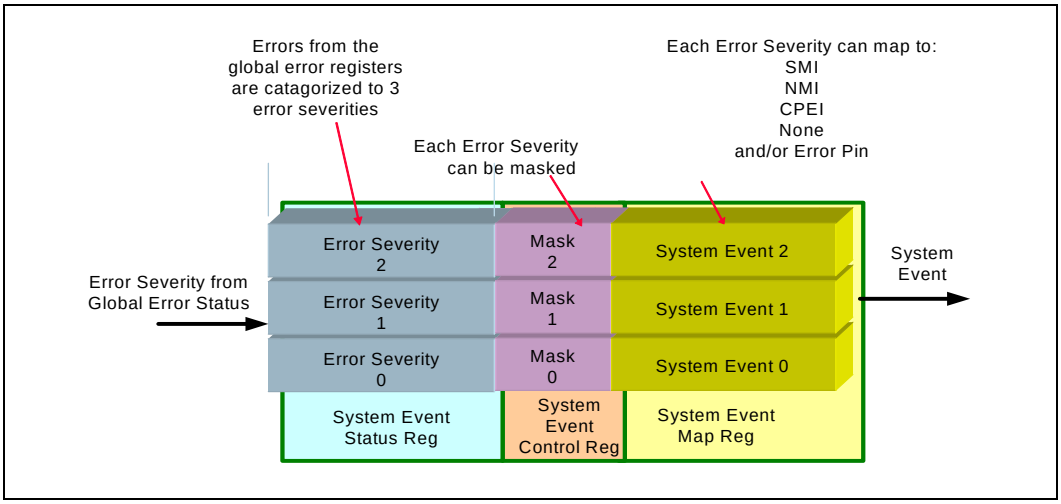


Figure 77 shows an example how an error is logged and reported to the system by the IIO.

Figure 77. IIO Error Logging and Reporting Example

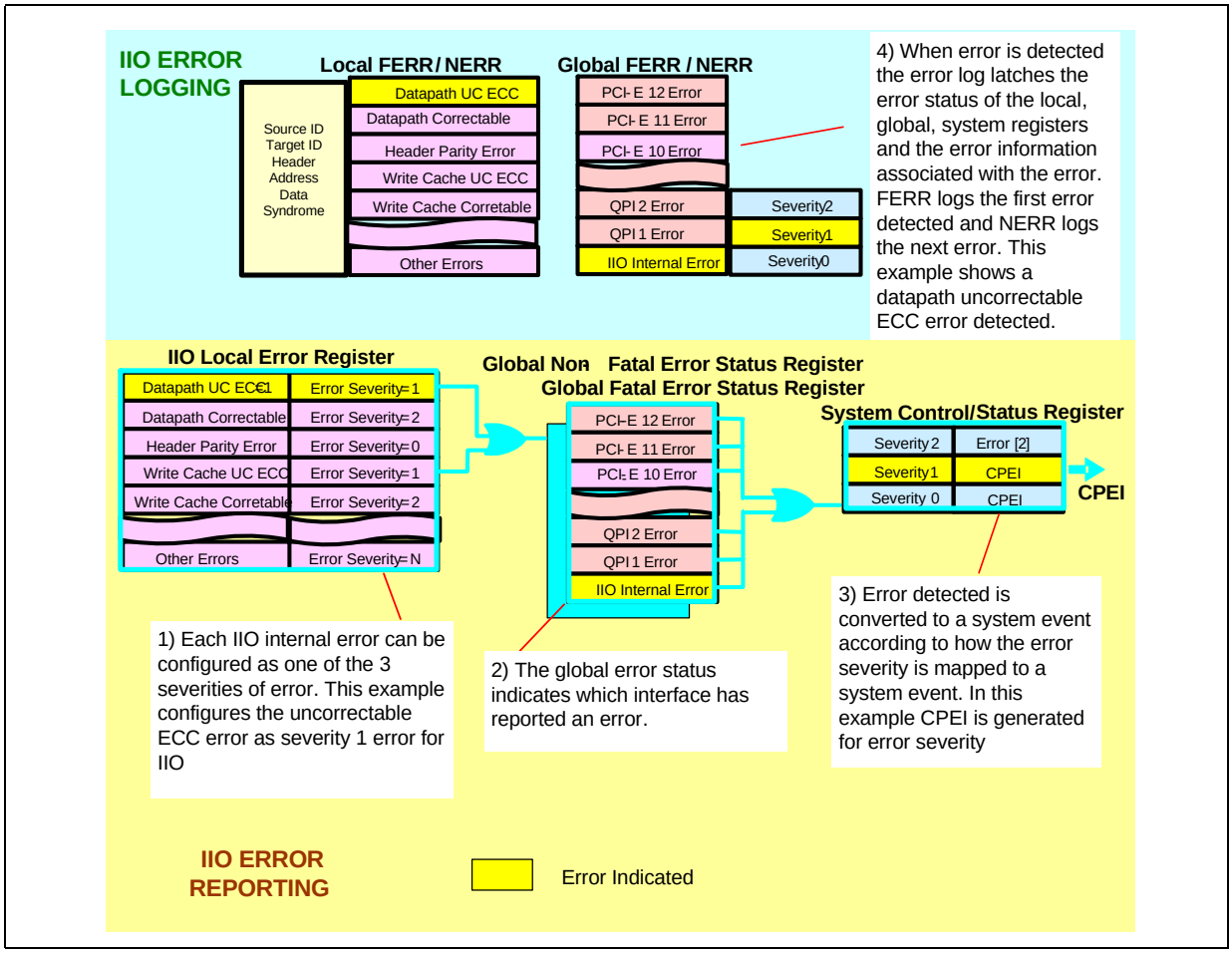
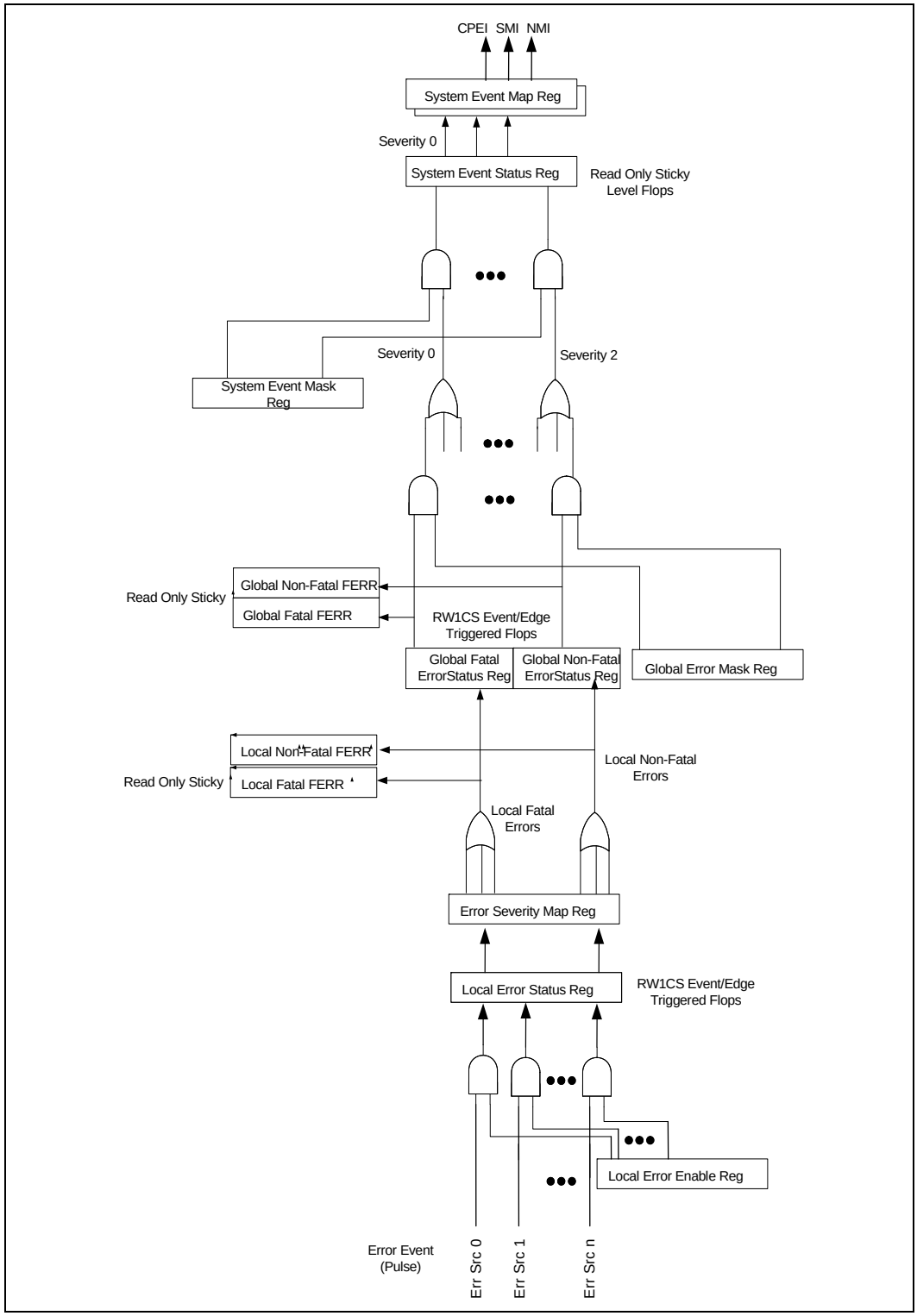


Figure 78 shows the logic diagram of the IIO local and global error registers.

Figure 78. Error Logging and Reporting Example



11.3.3.3 First and Next Error Log Registers

This section describes local error logging for Intel® QuickPath Interconnect and IIO core errors, and it describes global error logging. The log registers are named *FERR and *NERR in the IIO Register Specification. PCIe specifies its own error logging mechanism, This will not be described here. See the PCIe specification for details.

For error logging, the IIO categorizes detected errors into Fatal and Non-Fatal based on the error severity: Fatal for severity 2, Non-fatal for severity 0 and 1. Each category includes two sets of error logging: FERR (first error register) and NERR (next error register). The FERR register stores the information associated with the first detected error and NERR stores the information associated with subsequent errors.

Both FERR and NERR log the error status in the same format. They indicate errors that can be detected by the IIO in the format bit vector with one bit assigned to each error. The first error event is indicated by setting the corresponding bit in the FERR status register, a subsequent error(s) is indicated by setting the corresponding bit in the NERR register. In addition, the local FERR registers logs the ECC syndrome, address, and header of the erroneous cycle. The FERR indicates only one error, while the NERR can indicate multiple errors. Both the first error and next errors trigger system events.

Once the first error and the next error have been indicated and logged, the log registers for that error remain valid until either: 1) The first error bit is cleared 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.

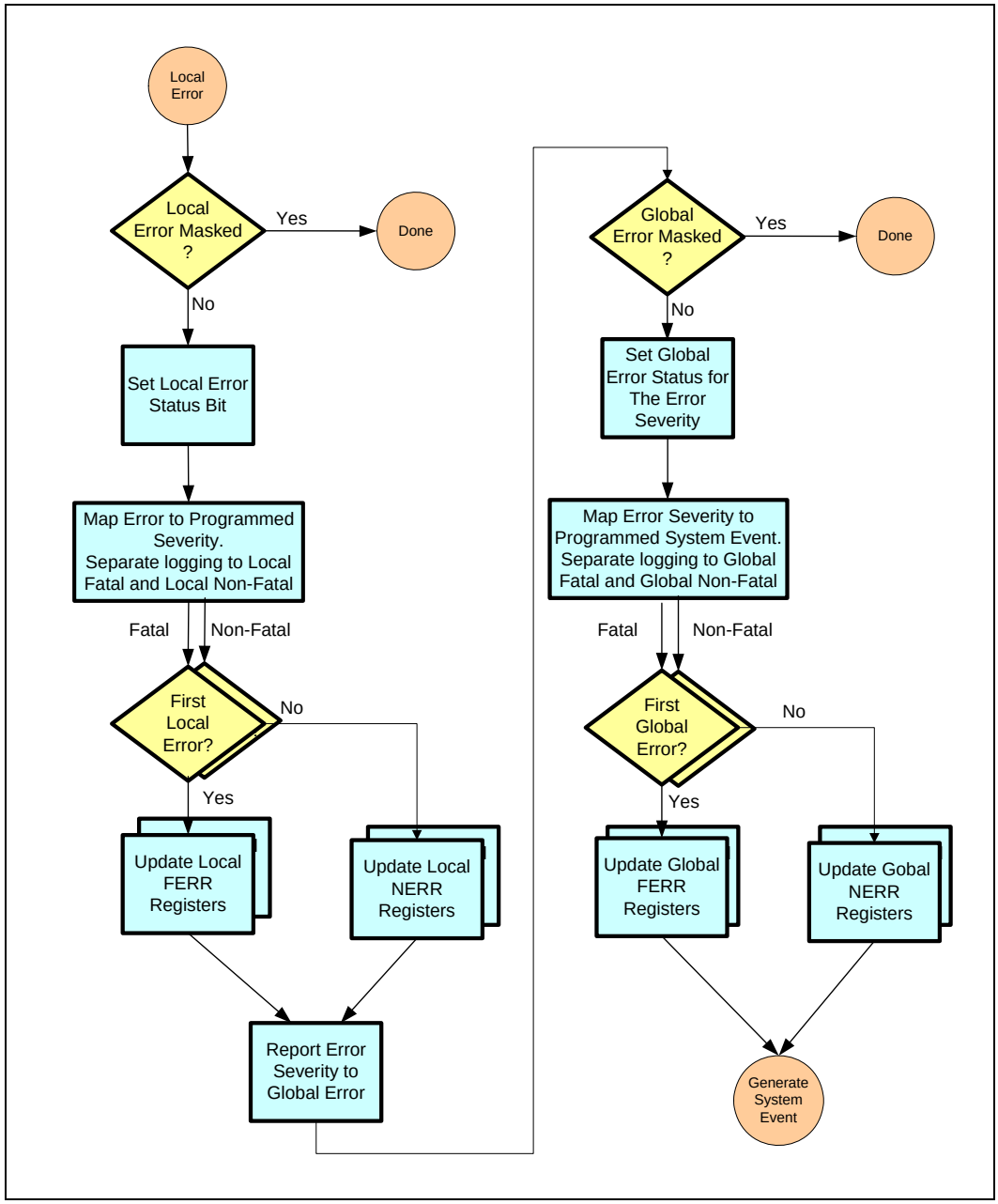
The hardware rules for updating the FERR and NERR registers and error logs are as follows:

1. The first error event is indicated by setting the corresponding bit in the FERR status register. A subsequent 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.
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. A fatal error has the highest priority, followed by recoverable errors, and then correctable errors.
5. Updates to the 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 software.
7. Error control registers are cleared by reset. The error status and log registers are cleared only by the power-on reset. The contents of error log registers are preserved across a reset, while PWRGOOD remains asserted.

11.3.3.4 Error Logging Summary

The following flow chart summarizes the error logging flow for the IIO. As illustrated in the flow chart, the left half depicts the local error logging flow and the right half depicts the global error logging flow. The local and the global error logging are similar. For simultaneous events, the IIO serializes the events with higher priority on the more severe error.

Figure 79. IIO Error Logging Flow



11.3.3.5 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 is forgotten.
2. The local error is mapped to its associated error severity defined by the error severity map register. Setting the local error status bit causes the logging of the error. Severity 0, 1, and 3 is logged in the local Non-Fatal FERR/NERR registers and severity 2 is logged in the local Fatal FERR/NERR registers. PCIe errors are logged according to the PCIe 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 is forgotten. The global FERR logs the first occurrence of local FERR/NERR event in the IIO and 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 and severity 2 is logged in the global Fatal FERR/NERR registers.
5. Global error register reports errors with associated error severity to the system event status register. The system event status is set if the system event reporting is enabled for error severity; otherwise the bit is not set and error is not reported.
6. Setting the system event bit triggers a system event generation according mapping defined in the system event map register. An associated system event is generated for the error severity and dispatched to CPU/BMC of the error (interrupt for CPU or error pin for BMC).
7. The global and local log registers provide information to identify source of the error. Software can read the log registers and clear the global and local error status bits.
8. Since error status bits are edge-triggered, 0 to 1 transition is required for bit reset. While the error status bit (local, global, or system event) is set to 1, all incoming error reporting to respective error status register are ignored (no 0 to 1 transition).
   a. When 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. When write to clear 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 errors. Clear local error bits. This does not cause generation of an interrupt with global bit still set. Then, clear the global error bit and write 0s (zeros) to the local error register. Writing 0s to the local status does not clear any status bit, but causes a re-evaluation of the error status bits. An error will be reported if there is any unclear local error bit.
   b. Read the 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 the IIO to report the error.
10. FERR logs the information for the first error detected by the associated error status register (local or global). The FERR log remains unchanged until all bits in the respective error status register are cleared by software. When all error bits are cleared, then FERR logging is re-enabled.

11.3.3.6 Error Containment

The IIO attempts to isolate and contain errors. For structures that can be contained, the error detected by the structure reports errors.

The IIO also provides an optional mode in which poisoned data received from either Intel® QuickPath Interconnect or peer PCI Express port is never sent out on PCI Express. I.e. any packet with poisoned data is dropped internally in by the IIO and an error is generated.

11.3.3.7 Error Counters

This feature allows the system management controller to monitor the component's health by periodically reporting the correctable error count. 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 read and clear the error logs in time to detect short bursts of errors across the chip. Over a long time period, the software uses these values to monitor the rate of change in error occurrences. This can help to identify potential component degradations, especially with respect to the memory interface.

11.3.3.7.1 Feature Requirements

A register with one-hot encoding will select which error types participate in error counting. It is unlikely that more than one error will occur within a cluster at a given time. Therefore, it is not necessary to count more than one occurrence in one clock cycle. The selection register will OR together 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.

- The 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 IIO 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.
- Each x4 PCI Express port contains one 7-bit counter (ERRCNT[6:0]) with a correctable error status selection register.
  - — Bit[7] is an overflow bit; all bits are sticky with a write logic 1 to clear.
- The DMI port contains one 7-bit counter (ERRCNT[6:0]) with a correctable error status selection register.
  - — Bit[7] is an overflow bit; all bits are sticky with a write logic 1 to clear.

11.3.3.8 Stop on Error

The System Event Map register selects the severity levels that activate Stop on Error (error freeze). A reset is required to clear the event, or a configuration write (using SMBus) to the stop on error bit in the selection register. Continued operation after an error freeze is *not* guaranteed. See the System Event Map register (SYSMAP).

11.4 IIO Intel® QuickPath Interconnect Interface RAS

The following sections provide an overview of the IIO Intel® QuickPath Interconnect RAS features. IIO CSI RAS features are summarized as shown in Table 127

Table 127. IIO Intel® QPI RAS Feature Support

Feature	IIO Intel® QPI 0 (Internal Between CPU and IIO)	Intel® QPI 1 (External)
Link Level 8-bit CRC	No	Yes
Link Level Retry	No	Yes
Dynamic Link Retraining and Recovery	No (x20 link width only)	No (x20 only)
Detection, logging and Reporting	Yes (Only for Protocol and Routing Support)	No

11.4.1 Intel® QuickPath Interconnect Error Detection, Logging, and Reporting

The IIO implements Intel® QuickPath Interconnect error detection and logging that follows the IIO local and global error reporting mechanism described in this chapter. These registers provide the control and logging of the errors detected on the Intel® QuickPath Interconnect interface. The IIO Intel® QuickPath Interconnect error detection, logging, and reporting provides the following features:

- Error indication by interrupt (CPEI, SMI, NMI).
- Error indication by response status field in response packets.
- Error indication by data poisoning.
- Error indication by error pin.
- Hierarchical time-out for fault diagnosis and FRU isolation.

For the physical and link layers there is an error log register per port. In the protocol and routing layers, there is a single error log.

11.5 PCI Express* RAS

The *PCI Express Base Specification,* Revision 2.0 defines a standard set of error reporting mechanisms and the IIO supports them all, including the error poisoning and Advanced Error Reporting. Any exceptions are called out where appropriate. The IIO PCIe ports support the following features:

- Link level CRC and retry.
- Dynamic link width reduction on link failure.
- PCIe error detection and logging.
- PCIe error reporting.

11.5.1 PCI Express* Link CRC and Retry

PCIe supports link CRC and link level retry for CRC errors. See the *PCI Express Base Specification,* Revision 2.0 for details.

11.5.2 Link Retraining and Recovery

The PCIe interface provides a mechanism to recover from a failed link. The PCIe link is capable of operating in different link width. The IIO will support PCIe port operation in x8, x4, x2, and x1. In case of a persistent link failure, the PCIe link can fall back to a smaller link width in and attempt to recover from the error. A PCIe x8 link can fall back to a x4 link. A PCIe x4 can fall back to x2 link, and then to X1 link. This mechanism enables the continuation of system operation in case of PCIe link failures. See the PCIe Base Specification, Revision 1.0a for details.

11.5.3 PCI Express Error Reporting Mechanism

The IIO supports standard and advanced PCIe error reporting for its PCIe ports. Since the IIO belongs to the root complex, its PCIe ports are implemented as root ports. See the *PCI Express Base Specification,* Revision 2.0 for details of PCIe error reporting. The following sections highlight the important aspects of PCIe error reporting mechanism.

11.5.3.1 PCI Express Error Severity Mapping in IIO

The errors reported to the IIO PCIe root port can optionally signal to the IIO global error logic according to their severities through the programming of the PCIe root control register (ROOTCON). When system error reporting is enabled for the specific PCIe error type, the IIO maps the PCIe error

to the IIO error severity and reports it to the global error status register. PCIe 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 IIO's error classification: Correctable as Correctable, Non-Fatal as Recoverable, and Fatal as Fatal.

11.5.3.2 Unsupported Transactions and Unexpected Completions

If the IIO receives a legal PCIe-defined packet that is not included in PCIe supported transactions, then the IIO treats that packet as an unsupported transaction and follows the PCIe rules for handling unsupported requests. If the IIO receives a completion with a requester ID set to the root port requester ID and there is no matching request outstanding, then this is considered an "Unexpected Completion". Also, the IIO detects malformed packets from PCI Express and reports them as errors per the PCI Express specification rules.

If the IIO 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, then the message is handled by the IIO as an Unsupported Request with appropriate error escalation, as defined in PCI Express specification. For Type 1 Vendor_Defined messages which terminate at the root complex, the IIO discards the message with no further action.

11.5.3.3 Error Forwarding

PCIe has a concept called Error Forwarding or Data Poisoning that allows a PCIe device to forward data errors across the interface without it being interpreted as an error originating on that interface. The IIO forwards the poison bit from the Intel® QuickPath Interconnect to PCIe and vice-versa, and also between PCI Express ports on peer-to-peer. Poisoning is accomplished by setting the EP bit in the PCIe TLP header.

11.5.3.4 Unconnected Ports

If a transaction targets a PCIe link that is not connected to any device or the link is down (DL_Down status), then the IIO treats that as a master abort situation. This is required for PCI bus scans to non-existent devices to go through without creating other side effects. If the transaction is non-posted, then the IIO synthesizes an Unsupported Request response status 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. Software accesses to the root port registers corresponding to a down PCIe interface does not generate an error.

11.6 IIO Errors Handling Summary

The following tables provide a summary of the errors that are monitored by the IIO. The IIO 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 128, 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 (e.g. severity 0 error triggers Error[0], severity 1 triggers Error[1],..., etc.).

Table 128 shows the default error severity mapping in the IIO and how each error severity is reported. Table 129 summarizes the default logging and responses on the IIO-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 those indicated in Table 129.

Table 128. IIO Default Error Severity Map

Error Severity	IIO	Intel® QPI	PCIe	Inband Error Reporting (Programmable)
0	Hardware Correctable Error	Hardware Correctable Error	Correctable Error.	NMI/SMI/CPEI IIO default: CPEI
1	Recoverable Error	Recoverable Error	Non-Fatal Error.	NMI/SMI/CPEI IIO default: CPEI
2	Fatal Error	Unrecoverable Error	Fatal Error	NMI/SMI/CPEI IIO default: SMI

Table 129. IIO Error Summary (Sheet 1 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
IIO Core Errors				
11	IIO access to non-existent address (Internal datapath coarse address decoders are unable to decode the target of the cycle).	1	For PCIe and Intel® QPI initiated transactions. This case includes snoops from Intel® QPI. A master abort convert to normal responses on Intel® QPI and additionally returning a data of all Fs on reads. SMBus to IIO accesses requests: The IIO returns UR status on SMBus.	FERR/NERR is logged in the IIO Core and Global Non-Fatal Error Log Registers: IIONFERRST IIONFERRHD IIONFERRSYN IIONNERRST GNERRST GNFERRST GNFERRTIME GNNERRST IIO core header is logged.
12	Intel® QPI transactions that cross 64B boundary.	1	Intel® QPI read: IIO returns all '1s' and normal response to Intel® QPI to indicate master abort. Intel® QPI write: IIO returns normal response and drops the write data. PCIe read: Completer Abort is returned on PCIe. PCIe non-posted write: Completer abort is returned on PCIe. The write data is dropped PCIe posted write: IIO drops the write data. SMBus to IIO accesses requests: IIO returns CA status on SMBus.	FERR/NERR is logged in IIO Core and Global Non-Fatal Error Log Registers: IIONFERRST IIONFERRHD IIONNERRST GNERRST GNFERRST GNFERRTIME GNNERRST IIO core header is logged.

Table 129. IIO Error Summary (Sheet 2 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
25	Core header queue parity error	2	Undefined	FERR/NERR is logged in IIO Core and Global Non-Fatal Error Log Registers: IIONFERRST IIONNERRST GFERRST GFFERRST GFFERRTIME GFNERRST No Header logging for this errors.
13	MSI address error on root port generated MSI's (i.e. MSI address is not equal to 0xFEEx_xxxx)	1	Drop the MSI interrupt	FERR/NERR is logged in IIO Core and Global Non-Fatal Error Log Registers: IIONFERRST IIONFERRHD IIONNERRST GNERRST GNFERRST GNFERRTIME GNNERRST IIO core header is logged.
C4	Master Abort Address error	1	IIO sends completion with MA status and logs the error.	FERR/NERR is logged in IIO Core and Global Non-Fatal Error Log Registers: IIONFERRST IIONFERRHD IIONNERRST GNERRST GNFERRST GNFERRTIME GNNERRST IIO core header is logged.
C5	Completer Abort Address Error		IIO sends completion with CA status and logs the error.	
C6	FIFO Overflow/ Underflow error	1	IIO logs the error.	FERR/NERR is logged in IIO Core and Global Non-Fatal Error Log Registers: IIONFERRST IIONFERRHD IIONNERRST GNERRST GNFERRST GNFERRTIME GNNERRST IIO core header is not logged.
Miscellaneous Errors				

Table 129. IIO Error Summary (Sheet 3 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
20	IIO Configuration Register Parity Error (not including Intel® QPI, PCIe or DMA registers which are covered elsewhere)	2	No Response. This error is not associated with a cycle. IIO detects and logs the error.	FERR/NERR is logged in Miscellaneous and Global Fatal Error Log Registers: MIFFERRST MIFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST No header is logged.
21	Persistent SMBus retry failure.	2	No Response. This error is not associated with a cycle. IIO detects and logs the error.	FERR/NERR is logged in Miscellaneous and Global Fatal Error Log Registers: MIFFERRST MIFFERRHD MIFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST No header is logged for this error.
22	Reserved			
23	Virtual Pin Port Error. (IIO encountered persistent VPP failure. The VPP is unable to operate.)			

Table 129. IIO Error Summary (Sheet 4 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
DMA Errors[2]				
40	DMA Transfer Source Address Error	1	IIO halts the corresponding DMA channel and aborts the current channel operation.	Log the error in corresponding CHANERRx_INT/ CHANERRPTRx registers and also DMAGLBERRPTR register. If error is forwarded to the global error registers, it is logged in global non-fatal log registers: GNERRST GNFERRST GNFERRTIME GNNERRST
41	DMA Transfer Destination Address Error			
42	DMA Next Descriptor Address Error			
43	DMA Descriptor Error			
44	DMA Chain Address Value Error			
45	DMA CHANCMD Error			
46	DMA Chipset Uncorrectable Data Integrity error (i.e. DMA detected uncorrectable data ECC error)			
47	DMA Uncorrectable Data Integrity error (i.e. DMA detected uncorrectable data ECC error)			
48	DMA Read Data Error			
49	DMA Write Data Error			
4A	DMA Descriptor Control Error			
4B	DMA Descriptor Length Error			
4C	DMA Completion Address Error			
4D	DMA Interrupt Configuration Errors - a) MSI address not equal to 0xFEEx_xxxx b) writes from non-MSI sources to 0xFEEx_xxxx			
4E	DMA CRC or XOR error			

Table 129. IIO Error Summary (Sheet 5 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
62	DMA configuration register parity error	2	N/A since the error is not associated with a specific transaction.	Log the error in corresponding DMAUNCERRSTS/ DMAUNCERRPTR registers and also DMAGLBERRPTR register. If error is forwarded to the global error registers, it is logged in global fatal log registers: GFERRST GFFERRST GFFERRTIME GFNERRST
63	DMA miscellaneous fatal errors (lock sequence error etc.)			
	PCIe/DMI Errors			
70	PCIe Receiver Error	0	Respond per PCIe specification	Log error per PCI Express AER requirements for these correctable errors/message. Log in XPGLBERRSTS, XPGLBERRPTR registers. If PCIe correctable error is forwarded to the global error registers, it is logged in global non-fatal log registers - GNERRST, GNFERRST, GNNERRST, GNFERRTIME.
71	PCIe Bad TLP			
72	PCIe Bad DLLP			
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. IIO detects and logs the error.	Log per 'Link bandwidth change notification mechanism' ECN. Log in XPCORERRSTS register. Log in XPGLBERRSTS, XPGLBERRPTR registers. If error is forwarded to the global error registers, it is logged in global non-fatal log registers - GNERRST, GNFERRST, GNNERRST, GNFERRTIME.
77	PCIe ECC correctable error (PCIe cluster detected internal ECC correctable error)			Log in XPCORERRSTS register. Log in XPGLBERRSTS, XPGLBERRPTR registers. If error is forwarded to the global error registers, it is logged in global non-fatal log registers - GNERRST, GNFERRST, GNNERRST, GNFERRTIME.

Table 129. IIO Error Summary (Sheet 6 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
80	Received 'Unsupported Request' completion status from downstream device	1	Intel® QPI to PCIe read: IIO returns all '1s' and normal response to Intel® QPI to indicate master abort. Intel® QPI to PCIe NP write: IIO returns normal response. PCIe to PCIe read/NP-write: 'Unsupported request' is returned[3] to original PCIe requester. SMBus accesses: IIO returns 'UR' response status on SMBus.	Log in XPUNCERRSTS register. Log in XPGLBERRSTS, XPGLBERRPTR registers. If error is forwarded to the global error registers, it is logged in global non-fatal log registers - GNERRST, GNFERRST, GNNERRST, GNFERRTIME.
81	IIO encountered a PCIe 'Unsupported Request' condition, on inbound address decode, as listed in Table 3-6, with the exception of SAD miss (see C6 for SAD miss), and those covered by entry #11		PCIe read: 'Unsupported request' completion is returned on PCIe. PCIe non-posted write: 'Unsupported request' completion is returned on PCIe. The write data is dropped. PCIe posted write: IIO drops the write data.	Log error per PCI Express AER requirements for unsupported request.[4] 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 - GNERRST, GNFERRST, GNNERRST, GNFERRTIME.
82	Received 'Completer Abort' completion status from downstream device		Intel® QPI to PCIe read: IIO returns all '1s' and normal response to Intel® QPI. Intel® QPI to PCIe NP write: IIO returns normal response. PCIe to PCIe read/NP-write: 'Completer Abort' is returned[5] to original PCIe requester. SMBus accesses: IIO returns 'CA' response status on SMBus.	Log in XPUNCERRSTS register. Log in XPGLBERRSTS, XPGLBERRPTR registers. If error is forwarded to the global error registers, it is logged in global non-fatal log registers - GNERRST, GNFERRST, GNNERRST, GNFERRTIME.
83	IIO encountered a PCIe 'Completer Abort' condition, on inbound address decode, as listed in Table 3-6.		PCIe read: 'Completer Abort' completion is returned on PCIe. PCIe non-posted write: 'Completer Abort' completion is returned on PCIe. The write data is dropped. PCIe posted write: IIO drops the write data.	Log error per PCI Express AER requirements for completer abort.[6] 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 - GNERRST, GNFERRST, GNNERRST, GNFERRTIME.

Table 129. IIO Error Summary (Sheet 7 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
84	Completion timeout on NP transactions outstanding on PCI Express/DMI	1	Intel® QPI to PCIe read: IIO returns normal response to Intel® QPI and all 1's for read data. Intel® QPI to PCIe non-posted write: IIO returns normal response to Intel® QPI. PCIe to PCIe read/non-posted write: UR[3] is returned on PCIe. SMBus reads: IIO returns a UR status on SMbus.	Log error per PCI Express AER requirements for the corresponding error. 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 - GNERRST, GNFERRST, GNNERRST, GNFERRTIME Note: a) A poisoned TLP received from PCIe is always treated as advisory-nonfatal error, if the associated severity is set to non-fatal. Also, received poisoned TLPs that are not forwarded over Intel® QPI are always treated as advisory-nonfatal errors, if severity is set to non-fatal. b) When a poisoned TLP is transmitted down a PCIe link, IIO does not log that condition in the AER registers.
85	Received PCIe Poisoned TLP		Intel® QPI to PCIe read: IIO returns normal response and poisoned data to Intel® QPI, if Intel® QPI has poisoned poison enabled. If poison is disabled then this error will be treated as a "QPI Parity Error". PCIe to Intel® QPI write: IIO forwards poisoned indication to Intel® QPI, if Intel® QPI has poisoned poison enabled. If poison is disabled then this error will be treated as a "QPI Parity Error". PCIe to PCIe read: IIO 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 (i.e. no packets on or after the poisoned data is allowed to go to the link). PCIe to PCIe posted/non-posted write: IIO 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 (i.e. no packets on or after the poisoned data is allowed to go to the link) and a UR[3] response is returned to the original requester, if the request is non-posted. SMBus to IIO accesses requests: IIO returns a UR response status on smbus	

Table 129. IIO Error Summary (Sheet 8 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
86	Received PCIe unexpected Completion	1	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 - GNERRST, GNFERRST, GNNERRST, GNFERRTIME.
87	PCIe Flow Control Protocol Error[7]			
88	Received ERR_NONFATAL Message from downstream device			
89	PCIe ECC Uncorrectable data Error (PCIe cluster detected internal ECC Uncorrectable data error)	2	Outgoing PCIe write (regardless of source): IIO drops the packet[8] and brings the link down as to not let any further transactions to proceed to link. A normal response (UR) is returned on Intel® QPI (PCIe) if request is non-posted. PCIe to Intel® QPI read requests (error encountered when on outbound completion datapath): IIO drops[8] the packet and brings the link down as to not let any further transactions to proceed to link. Inbound PCIe writes and Read completions (for outbound reads): IIO drops the packet. SMBus reads: IIO returns a UR status on SMbus.	Log in XPUNCERRSTS register. Log in XPGLBERRSTS, XPGLBERRPTR registers. If error is forwarded to the global error registers, it is logged in global fatal log registers - GFERRST, GFFERRST, GFNERRST, GFFERRTIME.
90	PCIe Malformed TLP[7]		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[7]			
92	PCIe Receiver Overflow			
93	Surprise Down			
94	Received ERR_FATAL message from downstream device.			
96	XP cluster internal configuration parity error. Note: XP Cluster is PCIe/DMI		No Response. This error is not associated with a cycle. The IIO detects and logs the error.	Log in XPUNCERRSTS register. Log in XPGLBERRSTS, XPGLBERRPTR registers. If error is forwarded to the global error registers, it is logged in global non-fatal log registers - GFERRST, GFFERRST, GFNERRST, GFFERRTIME.

Table 129. IIO Error Summary (Sheet 9 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
97	XP header queue parity error. Note: XP Cluster is PCIe/DMI	2	Undefined.	Log in XPUNCERRSTS register. Log in XPGLBERRSTS, XPGLBERRPTR registers. If error is forwarded to the global error registers, it is logged in global non-fatal log registers - GFERRST, GFFERRST, GFNERRST, GFFERRTIME.
98	MSI writes greater than a DWORD.		Drop the transaction.	Log in XPUNCERRSTS register. Log in XPGLBERRSTS, XPGLBERRPTR registers. 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 Errors				
A1	All faults except ATS spec defined CA faults. See the Intel® VT-d spec for complete details.	1	Unsupported Request response for the associated transaction on the PCI Express interface.	Error logged in VT-d Fault Record register. Error logged in VTUNCRRSTS and 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 IIO global error logic and logged in the global non-fatal registers. GNERRST GNFERRST GNFERRTIME GNNERRST
A3	Fault Reason Encoding 0xFF - Miscellaneous errors that are fatal to Intel® VT-d unit operation (e.g. parity error in a Intel® VT-d cache)	2	Drop the transaction. Continued operation of IIO is not guaranteed.	Error logged in VT-d fault record register. Error also logged in the VTUNCERRSTS register and in VTUNCERRPTR registers. These errors can also be routed to the IIO global error logic and logged in the global fatal registers. GFERRST GFFERRST GFFERRTIME GFNERRST
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 IIO 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 IIO global error logic and logged in the global fatal registers. GFERRST GFFERRST GFFERRTIME GFNERRST

Table 129. IIO Error Summary (Sheet 10 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
A4	Data parity error while doing a L2 lookup	2		Log in VTUNCERRSTS and VTUNCERRPTR registers. These errors can also be routed to the IIO 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 IIO 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 IIO 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 IIO global error logic and logged in the global fatal registers. GFERRST GFFERRST GFFERRTIME GFNERRST
A4	Unsuccessful status received in Intel® QPI read completion	1		Log in VTUNCERRSTS and VTUNCERRPTR registers. These errors can also be routed to the IIO 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 IIO global error logic and logged in the global fatal registers. GFERRST GFFERRST GFFERRTIME GFNERRST

Table 129. IIO Error Summary (Sheet 11 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
Intel® QPI Errors (Intel® QPI 0 - internal Intel® QPI between CPU & IIO)				
B2	Intel® QPI Physical Layer Detected a Intel® QPI Inband Reset (either received or driven by the IIO) and re-initialization completed successfully with no degradation in Width	0	No Response. This event is not associated with a cycle. The IIO detects and logs the event.	FERR/NERR is logged in Intel® QPI and Global Non-Fatal Error Log Registers and Intel® QPI physical layer register: QPINFERRST QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST QPIPHPIS QPIPHPPS
B3	Intel® QPI Protocol Layer Received CPEI message from Intel® QPI .	0	Normal Response. Note: This is really not an error condition but exists for monitoring by an external management controller.	Logging need to allow legacy IIO to assert Err_Corr to PCH. Non-legacy IIO should be programmed to mask this error to prevent duplicate error reporting. FERR/NERR is logged in Intel® QPI and Global Non-Fatal Error Log Registers: QPINFERRST QPINFERRHD QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST QPI header is logged
B4	Intel® QPI Write Cache Detected ECC Correctable Error.	0	IIO processes and responds the cycle as normal.	FERR/NERR is logged in Intel® QPI and Global Non-Fatal Error Log Registers: QPIPNFERRST QPIPNNERRST GNERRST GNFERRST GNFERRTIME GNNERRST No header is logged for this error.

Table 129. IIO Error Summary (Sheet 12 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
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”. IIO processes and responds the cycle as normal.	FERR/NERR is logged in Intel® QPI and Global Non-Fatal Error Log Registers and Intel® QPI Link layer register: QPINFERRST QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST
C1	Intel® QPI Protocol Layer Received Poisoned packet	1	Intel® QPI to PCIe write: IIO returns normal response to Intel® QPI and forwards poisoned data to PCIe. Intel® QPI to IIO write: IIO returns normal response to Intel® QPI and drops the write data. PCIe to Intel® QPI read: IIO forwards the poisoned data to PCIe IIO to Intel® QPI read: IIO drops the data. IIO to Intel® QPI read for RFO: IIO completes the write. If the bad data chunk is not overwritten, IIO corrupts write cache ECC to indicate the stored data chunk (64-bit) is poisoned.	FERR/NERR is logged in Intel® QPI and Global Non-Fatal Error Log Registers: QPIPNFERRST QPIPNFERRHD QPIPNNERRST GNERRST GNFERRST GNFERRTIME GNNERRST Intel® QPI header is logged
C2	IIO Write Cache uncorrectable Data ECC error	1	Write back includes poisoned data.	FERR/NERR is logged in Intel® QPI and Global Fatal Error Log Registers: QPIPFNFERRST QPIPFNERRST GNERRST GFFERRST GFFERRTIME GFNERRST
C3	IIO CSR access crossing 32-bit boundary.	1	Intel® QPI read: IIO returns all ‘1s’ and normal response to Intel® QPI to indicate master abort. Intel® QPI write: IIO returns normal response and drops the write	FERR/NERR is logged in IIO Core and Global Non-Fatal Error Log Registers: QPIPNFERRST QPIPNFERRHD QPIPNNERRST GNERRST GNFERRST GNFERRTIME GNNERRST Intel® QPI header is logged

Table 129. IIO Error Summary (Sheet 13 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
C7	Intel® QPI Physical Layer Detected an Intel® QPI Inband Reset (either received or driven by the IIO) and re-initialization completed successfully but width is changed	1	No Response -- This event is not associated with a cycle. IIO detects and logs the event.	FERR/NERR is logged in Intel® QPI and Global Non-Fatal Error Log Registers and Intel® QPI physical layer register: QPINFERRST QPINNERRST GNERRST GNFERRST GNFERRTIME GNNERRST QPIPHPIS QPIPHPPS
D3	Intel® QPI Link Layer Detected Control Error (Buffer Overflow or underflow, illegal or unsupported LL control encoding, credit underflow). Sub-status will be logged in QPI[1:0]DBGERRST (D13:F0-1:F34h) register.	2	No Response -- This error is not associated with a cycle. IIO detects and logs the error.	FERR/NERR is logged in Intel® QPI and Global Non-Fatal Error Log Registers: QPIFFERRST QPIFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST No header logged for this error
D4	Intel® QPI Parity Error in link layer (See Section 11.7.3 for details) or poisoned in LL Tx (Inbound) when poison is disabled. Sub-status will be logged in QPI[1:0]PARERRLOG register.	2	No Response -- This error is not associated with a cycle. IIO detects and logs the error.	FERR/NERR is logged in Intel® QPI and Global Fatal Error Log Registers: QPIFFERRST QPIFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST
D5	Intel® QPI Protocol Layer Detected Time-out in ORB	2	Intel® QPI read: return completer abort. Intel® QPI non-posted write: IIO returns completer abort. Intel® QPI posted write: no action	FERR/NERR is logged in Intel® QPI and Global Non-Fatal Error Log Registers: QPIPFFERRST QPIPFFERRHD QPIPFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST Intel® QPI header is logged (D6 Only)
D6	Intel® QPI Protocol Layer Received Failed Response			

Table 129. IIO Error Summary (Sheet 14 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
D7	Intel® QPI Protocol Layer Received Unexpected or Illegal Response/ Completion	2	Drop Transaction, No Response. This will cause time-out in the requester.	FERR/NERR is logged in Intel® QPI and Global Non-Fatal Error Log Registers: QPIPFFERRST QPIPFFERRHD QPIPFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST Intel® QPI header is logged (D8 Only)
D8	Intel® QPI Protocol Layer Received illegal packet field or incorrect target Node ID or poisoned in LL Rx (outbound) when poison is disabled			
DA	Intel® QPI Protocol Layer Queue/Table Overflow or Underflow. Sub-status will be logged in QPI[1:0]DBGPRERRST (D13:F0-1:F38h) register.	2	No Response -- This error is not associated with a cycle. IIO detects and logs the error.	FERR/NERR is logged in Intel® QPI and Global Fatal Error Log Registers: QPIPFFERRST QPIPFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST No header logged for this error
DB	Intel® QPI Protocol Parity Error. Sub-status will be logged in QPI[1:0]PRPARERRLOG register. See Section 11.7.3 for details.	2	No Response -- This error is not associated with a cycle. IIO detects and logs the error.	FERR/NERR is logged in Intel® QPI and Global Fatal Error Log Registers: QPIPFFERRST QPIPFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST

Table 129. IIO Error Summary (Sheet 15 of 15)

ID	Error	Default Error Severity	Transaction Response	Default Error Logging[1]
DC	IIO 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. (e.g. completion time-out in Intel® QPI request agent, or PCIe request agent.)	FERR/NERR is logged in Intel® QPI and Global Fatal Error Log Registers: QPIPFFERRST QPIPFFERRHD QPIPFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST Intel® QPI header is logged
DE	IIO Routing Table pointed to a disabled Intel® QPI port			
DF	Illegal inbound request (includes VCp/VC1 request when they are disabled)			
DG	Intel® QPI Link Layer detected unsupported/ undefined packet (e.g., RSVD_CHK, message class, opcode, vn, viral) Note: do not support Viral Alert Generation	2	No Response -- This error is not associated with a cycle. IIO detects and logs the error.	FERR/NERR is logged in Intel® QPI and Global Fatal Error Log Registers: QPIFFERRST QPIFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST No header logged for this error
DH	Intel® QPI Protocol Layer Detected unsupported/ undefined packet Error (message class, opcode and vn only) -	2	No Response -- This error is not associated with a cycle. IIO detects and logs the error.	FERR/NERR is logged in Intel® QPI Protocol and Global Fatal Error Log Registers: QPIPFFERRST QPIPFNERRST GFERRST GFFERRST GFFERRTIME GFNERRST

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. IIO does not detect any Intel® QuickData Technology DMA unaffiliated errors and hence these errors are not listed in the subsequent DMA error discussion
3. 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.
4. In some cases, IIO might not be able to log the error/header in AER when it signals UR back to the PCIe device.
5. 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.
6. In some cases, IIO might not be able to log the error/header in AER when it signals CA back to the PCIe device.
7. Not all cases of this error are detected by IIO.
8. If error is detected too late for IIO to drop the packet internally, it needs to 'EDB' the transaction.

11.7 Hot Add/Remove Support

The Intel® Xeon® processor C5500/C3500 series has hot add/remove support for PCIe devices. This feature allows physical hot plug/removal of a PCIe device connected to the processor IIO. In addition, physical hot add/remove for other IO devices downstream to the IIO may be supported by downstream bridges. Hot plug of PCIe and IO devices are defined in the PCIe/PCI specifications.

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 in Intel® Xeon® processor C5500/C3500 series:

- **Physical Hot add/remove**
  This is the conventional hot plug of a physical component in 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 physically 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 Intel® Xeon® processor C5500/C3500 series supports both physical and logical hot add/remove of various components in the system. These include:

- **PCIe and IO Devices**
  Intel® Xeon® processor C5500/C3500 series-based platforms support PCIe and IO device hot add/remove. This feature allows physical hot plug/removal of an PCIe device connected to the IIO. In addition, physical hot plug/remove for other IO devices downstream to IIO may be supported by downstream bridges. Hot plug of PCIe and IO devices are defined in the PCIe/PCI specifications.

11.7.1 Hot Add/Remove Rules

1. The final system configuration after hot add/remove must not violate any of the topology rules.
2. Legacy bridge (PCH) itself cannot be hot added/removed from the IIO (no DMI hot plug support).

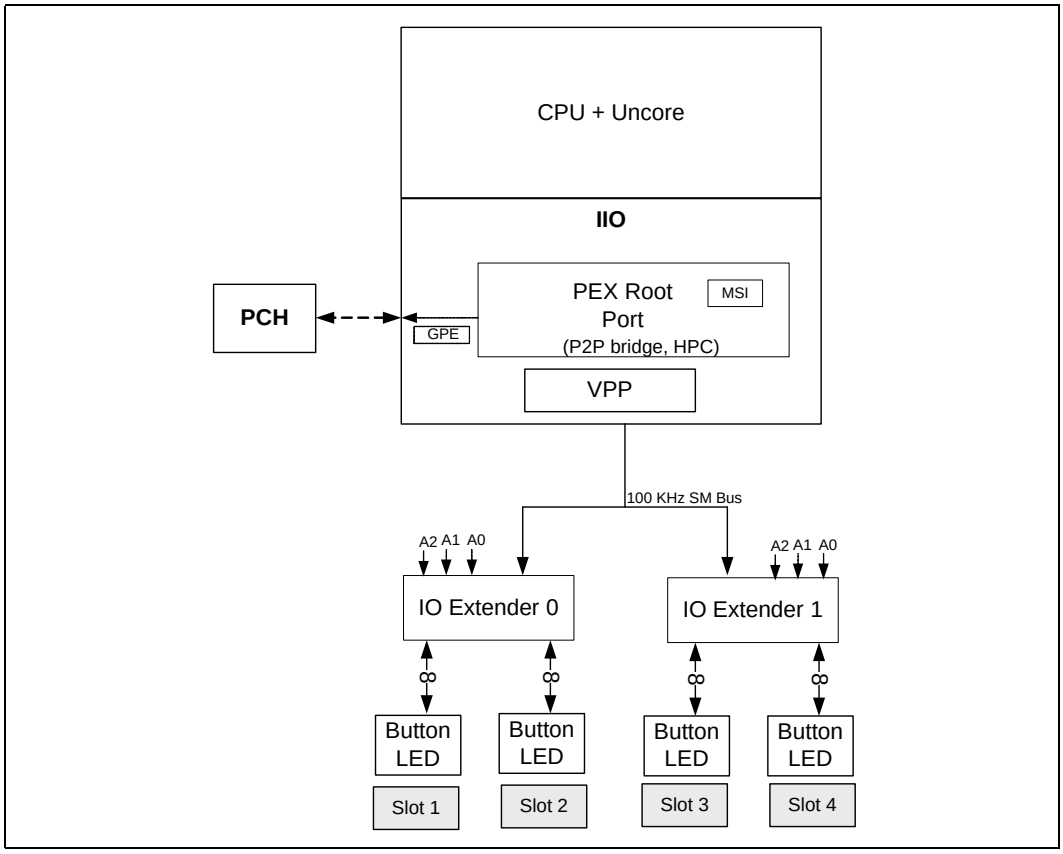
11.7.2 PCIe Hot Plug

PCIe hot plug is supported through the standard PCIe native hot plug. The Intel® Xeon® processor C5500/C3500 series IIO only supports the sideband hot plug signals and does not support the inband hot plug messages. The IIO contains a virtual pin port (VPP) that serially shifts in and out the sideband PCIe hot plug signals. External platform logic is required to convert IIO serial stream to parallel. The virtual pin port is implemented via a dedicated SMBus port as shown in Figure 80.

Summary of IIO PCIe hot plug support:

- Support for up to five hot plug slots selectable by BIOS.
- Support for serial mode hot plug only using smbus devices like 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.
- IIO does not support inband hot plug messages on PCIe.
  — The IIO does not issue them and the IIO discards them silently if received.
- A hot plug event cannot change the number of ports of the PCIe interface (i.e. bifurcation).

Figure 80. IIO PCI Express Hog Plug Serial Interface



11.7.2.1 PCI Express Hot Plug Interface

Table 130 describes how Intel® Xeon® processor C5500/C3500 series provides these signals serially to external controller, these signals are controlled and reflected in the PCIe root port hot plug registers.

Table 130. 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 see the *PCI Express Base Specification,* Revision 1.1.	Indicator can be off, on, or blinking. The required state for the indicator is specified with the Attention Indicator Register. IIO blinks this LED at 1 Hz.
PWRLED	This indicator is connected to the Power LED on the baseboard. For a precise definition see the *PCI Express Base Specification,* Revision 1.1.	Indicator can be off, on, or blinking. The required state for the indicator is specified with the Power Indicator Register. The IIO blinks this LED at 1 Hz.
BUTTON#	Input signal per slot which indicates that the user wishes to hot remove or hot add a PCIe 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 PCH is sent.[1]

Table 130. Hot Plug Interface (Sheet 2 of 2)

Signal Name	Description	Action
PRSNT#	Input signal that indicates if a hot pluggable PCIe 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 PCH.[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 PCH.[1]
PWREN#	Output signal allowing software to enable or disable power to a PCIe slot.	If the Power Controller Register is set, the IIO 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 in place and is operated by software. MRL is used for card-edge and EMLSTS# is used for SIOM form factors.	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 PCH.[1]
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. See the <*Blue*>*PCI Express Server/Workstation Module Electromechanical Spec Rev 1.0* for details of the timing requirements of this pin output.	Supported for the serial interface and is used only for the SIOM form-factor.

1. For legacy operating systems, the described Assert_HPGPE/Deassert_HPGPE mechanism is used to interrupt the platform for PCIe hotplug events. For newer operating systems, this mechanism is disabled and the MSI capability is used by the IIO instead.

11.7.2.2 PCI Express Hot Plug Interrupts

The Intel® Xeon® processor C5500/C3500 series IIO generates either an MSI or an Assert/ Deasset_HPGPE message to the PCH over the DMI link when a hot plug event occurs on standard PCIe interfaces. The GPE messages are selected when bit 3 in MISCCTRLSTS: Misc. Control and Status Register is set. If this bit is clear, then MSI method is selected (the MSI Enable bit in the (MSIX)MSGCTRL register does not control selection of GPE vs. MSI method). See the *PCI Express Base Specification,* Revision 1.1 for details of MSI generation on a PCIe hotplug event.

A hot plug event is 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 has a corresponding bit in the PCIe slot status, control registers. The IIO processes hot plug events using the wired-OR (collapsed) mechanism of the various bits across the ports to emulate the level sensitive need for the legacy interrupts on DMI.

When the output of the wired-OR logic is set, the Assert_HPGPE is sent to the PCH. IIO combines the virtual message from all the ports and then presents a collapsed set of virtual wire messages to the PCH. When software clears all the associated register bits (that are enabled to cause an event) across the ports, the IIO will generate a Deassert_HPGPE message to the PCH.

Figure 81. MSI Generation Logic at each PCI Express Port for PCI Express Hot Plug

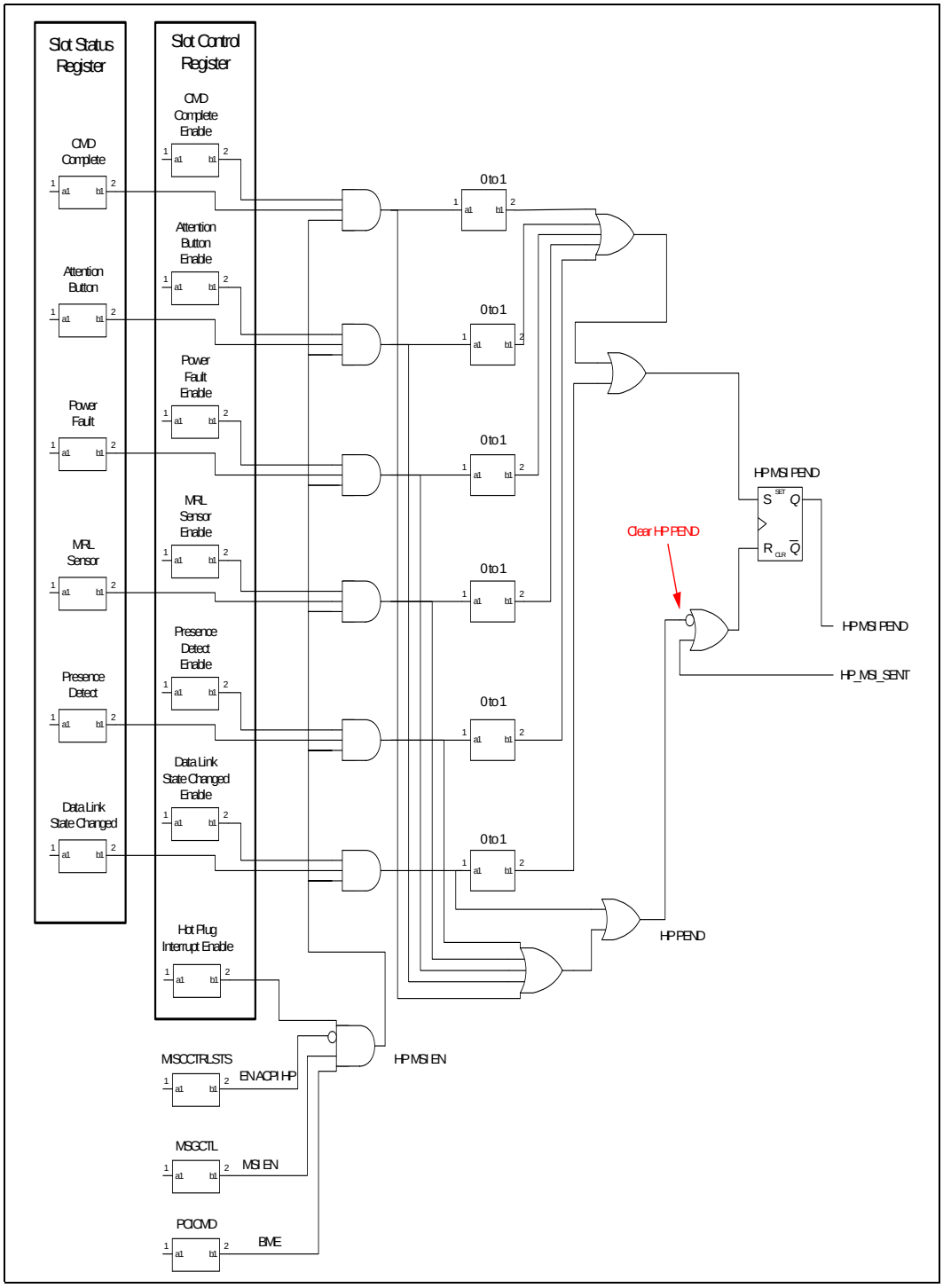
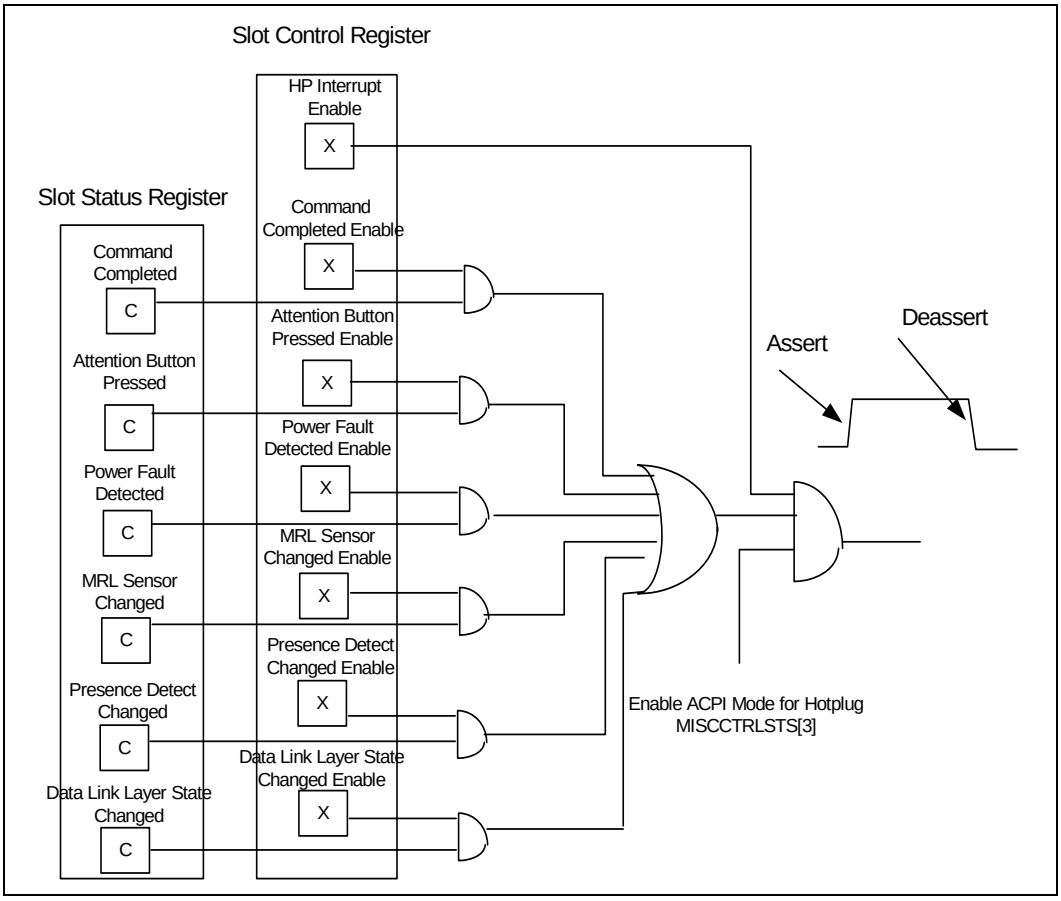


Figure 82. GPE Message Generation Logic at each PCI Express Port for PCI Express Hot Plug



11.7.2.3 Virtual Pin Ports (VPP)

The Intel® Xeon® processor C5500/C3500 series IIO contains a virtual pin port (VPP) that serially shifts in and out the sideband PCIe hot plug signals. VPP is a 100 KHz SMBus interface that connects to a variable number of serial to parallel I/O ports.

Example: the Phillips* PCA9555. Each 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 IIO supports up to five PCIe hot plug ports through the VPP interface with maximum of two PCA9555 populated.

The IIO VPP only supports SMBus devices with the command sequence shown Table 131. Each PCIe port is associated with one of these 8-bit ports. The mapping is defined by a Virtual Pin Port register field for each PCIe slot. The VPP register holds the SMBus address and Port (0 or 1) of the I/O port associated with the PCIe. A[1:0] pins on each I/O extender (i.e. PCA9555) connected to the IIO must be strapped uniquely.

Table 131. I/O Port Registers in On-Board SMBus devices Supported by IIO

Command	Register	IIO 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 IIO
5	Polarity Inversion Port 1	
6	Configuration Port 0	Direction (Input/Output)
7	Configuration Port 1	

11.7.2.4 Operation

When the Intel® Xeon® processor C5500/C3500 series IIO comes out of Powergood reset, the I/O ports are inactive. The IIO is not aware of how many I/O extenders are connected to VPP, what their addresses are, nor what PCIe port are hot-pluggable. The IIO does not master any commands on the SMBus until one VPP enable bit is set.

For a PCI Express slot, an additional FF (Form Factor) bit (see “MISCCTRLSTS: Misc. Control and Status Register6”) is used to differentiate card, module or cable hotplug support. When the BIOS sets the VPP Enable bit (see “VPPCTL: VPP Control”), the IIO initializes the associated VPP corresponding to that root port with direction and logic Level configuration. From then on, the IIO continually scans in the inputs corresponding to that port and scans out the outputs corresponding to that port. VPP registers for PCI Express ports that do not have the VPP enable bit set are invalid and ignored.

Table 132 defines how the eight hot-plug signals are mapped to pins on the I/O extender’s GPIO pins. When the IIO is not doing a direction or logic level write, which would happen when a PCIe 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 and to reduce logic complexity, 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 hotplug capable root port, the value read from the other port of the external device are throw away and only de-asserted values are shifted out for the outputs (see Table 132 for the list of output signals and their polarity).

Table 132. Hot Plug Signals on a Virtual Pin Port (Sheet 1 of 2)

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	Low_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

Table 132. Hot Plug Signals on a Virtual Pin Port (Sheet 2 of 2)

Bit	Direction	Voltage Logic Table	Signal	Logic True Meaning	Logic False Meaning
Bit 5	Input	Low_True	PWRFLT#	PWR Fault in the VRM	NO PWR Fault in the VRM
Bit 6	Input	High_True	MRL/EMILS	MRL is open/EMILS is disengaged	MRL is closed/EMILS is engaged
Bit 7	Output	High_True	EMIL	Toggle interlock state -Pulse output 100ms when '1' is written	No effect

Table 133 describes the sequence generated for a write to an IO 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 PCIe Slot Control register for the associated PCIe slot.

Table 133. Write Command

Bits	IIO Drives	IO Port Drives	Comment
1	Start		SDL falling followed by SCL falling
7	Address[6:0]		[6:3] = 0100 [2:0] = "VPPCTL: VPP Control"
1	0		indicates write
1		ACK	If NACK is received, IIO completes with stop and sets status bit in "VPPSTS: VPP Status Register".
8	Command Code		Register Address see Table 131 [7:3]=00000,[2:1] = 01 for Output, 11 for Direction [0] = 0
1		ACK	If NACK is received, IIO completes with stop and sets in "VPPSTS: VPP Status Register".
8	Data		One bit for each IO as per Table 132
1		ACK	If NACK is received, IIO completes with stop and sets in "VPPSTS: VPP Status Register".
8	Data		One bit for each IO as per Table 132
1		ACK	If NACK is received, IIO completes with stop and sets in "VPPSTS: VPP Status Register".
1	Stop		

The IIO 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 IIO always reads inputs from both 8-bit ports. If the VPP is valid, then the IIO 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 IO Port. Results from invalid VPPs are discarded. Table 134 defines the read command format.

11.7.2.5 Miscellaneous Notes

Table 134. Read Command

Bits	IIO Drives	IO Port Drives	Comment
1	Start		SDL falling followed by SCL falling
7	Address[6:0]		[6:3] = 0100 [2:0] = "VPPCTL: VPP Control"
1	0		indicates write
1		ACK	If NACK is received, IIO completes with stop and sets in "VPPSTS: VPP Status Register".
8	Command Code		Register Address [2:0] = 000
1		ACK	If NACK is received, IIO completes with stop and sets in "VPPSTS: VPP Status Register".
1	Start		SDL falling followed by SCL falling
7	Address[6:0]		[6:3] = 0100 [2:0] = "VPPSTS: VPP Status Register"
1	1		indicates read
8		Data	One bit for each IO as per Table 132. The IIO always reads from both ports. Results for invalid VPPs are discarded
1	ACK		
8		Data	One bit for each IO as per Table 132. The IIO always reads from both ports. Results for invalid VPPs are discarded
1	NACK		
1	Stop		

11.7.2.5.1 VPP Port Reset

The VPP port logic in the IIO is reset immediately when a PWRGOOD reset happens. When a hard reset happens, the IIO internally delays resetting the VPP logic until the currently running transaction on the VPP port reaches a logical termination point, i.e. reaches a transaction boundary. Then the VPP logic is reset within a timeout. This delayed reset of VPP logic guarantees that the VPP port is not hung after reset, which can happen if a transaction was terminated randomly while the VPP device like PCA9555 was still actively listening on the bus while IIO is being reset. The rest of the IIO could be in reset while the VPP port is still active. After a hard reset, IIO would start activity on the VPP port provided the VPP port was configured before hard reset was asserted. This is because the VPP port control registers are all sticky.

Some caveats relating to VPP port reset:

- If the Powergood signal was toggled without actually removing power, there is a potential to still hang the VPP port since the VPP device would not be reset whereas IIO would be:
  - The board needs to work around this issue by not toggling powergood without removing power to PCA9555 (a FET on the power input to PCA9555 that is controlled by powergood would do the trick).
- There is a potential that the EMIL signal remains stuck at 1 if the IIO is reset in the middle of pulsing that signal. This can potentially cause malfunction of the electro-mechanical latch. To prevent that board must AND the EMIL output of IIO with the appropriate reset signal before feeding to the latch.

11.7.2.5.2 Attention Button

The IIO implements the attention button signal as an edge triggered signal, i.e. the attention button status bit in the Slot Status register is set when an asserting edge on the signal is detected. If an asserting edge on attention button is seen in the same clock, then software clears the attention button status bit, the bit should remain set and if MSI is enabled, another MSI message should be generated.

Also, debounce logic on the attention button signal is to be implemented on the board.

11.7.2.5.3 Power Fault

IIO implements the Power Fault signal as a level signal with the following property. When the signal asserts, the IIO sets the Power Fault status bit in the Slot Status register (and a 0->1 edge on the status bit would cause an MSI interrupt, if enabled). When software clears the status bit, IIO re-samples the power fault signal and if it is still asserted, the status bit is set once more and it triggers one more MSI interrupt, if enabled.

11.7.3 Intel® QPI Hot Plug

The Intel® Xeon® processor C5500/C3500 series does not support Intel® QPI Hot Plug.

§ §

12.0 Packaging and Signal Information

12.1 Signal Descriptions

This chapter describes the processor signals. They are arranged in functional groups according to their associated interface or category. All straps are only a weak pulldown needed if a Vss is desired. The following notations describe the signal types:

Notations	Signal Type
I	Input pin
O	Output pin
I/O	Bi-directional input/output pin
Analog	Analog reference or output

12.1.1 Intel® QPI Signals

Table 135. Intel® QPI Signals

Signal Names	I/O Type	Description
QPI_CLKRX_DP QPI_CLKRX_DN	I	Intel® QuickPath Interconnect Received Clock.
QPI_CLKTX_DP QPI_CLKTX_DN	O	Intel® QuickPath Interconnect Forwarded Clock.
QPI_COMP[1:0]	I	Intel® QuickPath Interconnect Compensation: Used for the external impedance matching resistors. Must be terminated on the system board using precision resistor.
QPI_RX_DN[19:0] QPI_RX_DP[19:0]	I	Intel® QuickPath Interconnect Data Input.
QPI_TX_DN[19:0] QPI_TX_DP[19:0]	O	Intel® QuickPath Interconnect Data Output.

12.1.2 System Memory Interface

12.1.2.1 DDR Channel A Signals

Table 136. DDR Channel A Signals

Signal Names	I/O Type	Description
DDRA_BA[2:0]	O	Bank Address Select: These signals define which banks are selected within each SDRAM rank.
DDRA_CAS#	O	CAS Control Signal: Used with DDRA_RAS# and DDRA_WE# (along with DDRA_CS#) to define the SDRAM commands.
DDRA_CKE[3:0]	O	Clock Enable: (one per rank) used to: • Initialize the SDRAMs during power-up. • Power-down SDRAM ranks. • Place all SDRAM ranks into and out of self-refresh during STR.
DDRA_CLK_DN[3:0] DDRA_CLK_DP[3:0]	O	SDRAM Differential Clock: Channel A SDRAM differential clock signal pair. The crossing of the positive edge of DDRA_CLK_DPx and the negative edge of its complement DDRA_CLK_DNx are used to sample the command and control signals on the SDRAM.
DDRA_CS[7:0]#	O	Chip Select: (one per rank) Used to select particular SDRAM components during the active state. There is one chip-select for each SDRAM rank.
DDRA_DQ[63:0]	I/O	Data Bus: Channel A data signal interface to the SDRAM data bus.
DDRA_DQS_DN[17:0] DDRA_DQS_DP[17:0]	I/O	Data Strobes: DDRA_DQS[17:0] and its complement signal group make up a differential strobe pair. The data is captured at the crossing point of DDRA_DQS_DP[17:0] and its DDRA_DQS_DN[17:0] during read and write transactions. Different numbers of strobes are used depending on whether the connected DRAMs are x4,x8 or have checkbits.
DDRA_ECC[7:0]	I/O	Check Bits - An Error Correction Code is driven along with data on these lines for DIMMs that support that capability.
DDRA_MA[15:0]	O	Memory Address: These signals are used to provide the multiplexed row and column address to the SDRAM.
DDRA_MA_PAR	O	Odd parity across address and command.
DDRA_ODT[3:0]	O	On Die Termination: Active Termination Control. Enables various combinations of termination resistance in the target and non-target DIMMs when data is read or written.
DDRA_PAR_ERR[2:0]#	I	Parity Error detected by Registered DIMM (one per DIMM).
DDRA_RAS#	O	RAS Control Signal: Used with DDRA_CAS# and DDRA_WE# (along with DDRA_CS#) to define the SRAM commands.
DDRA_RESET#	O	Resets DRAMs. Held low on power up, held high during self refresh, otherwise controlled by configuration register.
DDRA_WE#	O	Write Enable Control Signal: Used with DDRA_RAS# and DDRA_CAS# (along with DDRA_CS#) to define the SDRAM commands.

12.1.2.2 DDR Channel B Signals

Table 137. DDR Channel B Signals

Signal Names	I/O Type	Description
DDRB_BA[2:0]	O	Bank Address Select: These signals define which banks are selected within each SDRAM rank.
DDRB_CAS#	O	CAS Control Signal: Used with DDRB_RAS# and DDRB_WE# (along with DDRB_CS#) to define the SDRAM Commands.
DDRB_CKE[3:0]	O	Clock Enable: (one per rank) used to: • Initialize the SDRAMs during power-up. • Power-down SDRAM ranks. • Place all SDRAM ranks into and out of self-refresh during STR.
DDRB_CLK_DN[3:0] DDRB_CLK_DP[3:0]	O	SDRAM Differential Clock: Channel B SDRAM Differential clock signal pair. The crossing of the positive edge of DDRB_CLK_DPx and the negative edge of its complement DDRB_CLK_DNx are used to sample the command and control signals on the SDRAM.
DDRB_CS[7:0]#	O	Chip Select: (one per rank) Used to select particular SDRAM components during the active state. There is one chip-select for each SDRAM rank.
DDRB_DQ[63:0]	I/O	Data Bus: Channel B data signal interface to the SDRAM data bus.
DDRB_DQS_DN[17:0] DDRB_DQS_DP[17:0]	I/O	Data Strobes: DDRB_DQS[17:0] and its complement signal group make up a differential strobe pair. The data is captured at the crossing point of DDRB_DQS_DP[17:0] and its DDRB_DQS_DN[17:0] during read and write transactions. Different numbers of strobes are used depending on whether the connected DRAMs are x4,x8 or have checkbits.
DDRB_ECC[7:0]	I/O	Check Bits - An Error Correction Code is driven along with data on these lines for DIMMs that support that capability.
DDRB_MA[15:0]	O	Memory Address: These signals are used to provide the multiplexed row and column address to the SDRAM.
DDRB_MA_PAR	O	Odd parity across address and command.
DDRB_ODT[3:0]	O	On Die Termination: Active Termination Control. Enables various combinations of termination resistance in the target and non-target DIMMs when data is read or written.
DDRB_PAR_ERR[2:0]#	I	Parity Error detected by Registered DIMM (one per DIMM).
DDRB_RAS#	O	RAS Control Signal: Used with DDRB_CAS# and DDRB_WE# (along with DDRB_CS#) to define the SRAM commands.
DDRB_RESET#	O	Resets DRAMs. Held low on power up, held high during self refresh, otherwise controlled by configuration register.
DDRB_WE#	O	Write Enable Control Signal: Used with DDRB_RAS# and DDRB_CAS# (along with DDRB_CS#) to define the SDRAM commands.

12.1.2.3 DDR Channel C Signals

Table 138. DDR Channel C Signals

Signal Names	I/O Type	Description
DDRC_BA[2:0]	O	Bank Address Select: These signals define which banks are selected within each SDRAM rank.
DDRC_CAS#	O	CAS Control Signal: Used with DDRC_RAS# and DDRC_WE# (along with DDRC_CS#) to define the SDRAM commands.
DDRC_CKE[3:0]	O	Clock Enable: (one per rank) used to: • Initialize the SDRAMs during power-up. • Power-down SDRAM ranks. • Place all SDRAM ranks into and out of self-refresh during STR.
DDRC_CLK_DN[3:0] DDRC_CLK_DP[3:0]	O	SDRAM Differential Clock: Channel C SDRAM Differential clock signal pair. The crossing of the positive edge of DDRC_CLK_DPx and the negative edge of its complement DDRC_CLK_DNx are used to sample the command and control signals on the SDRAM.
DDRC_CS[7:0]#	O	Chip Select: (one per rank) Used to select particular SDRAM components during the active state. There is one chip-select for each SDRAM rank.
DDRC_DQ[63:0]	I/O	Data Bus: Channel C data signal interface to the SDRAM data bus.
DDRC_DQS_DN[17:0] DDRC_DQS_DP[17:0]	I/O	Data Strobes: DDRC_DQS[17:0] and its complement signal group make up a differential strobe pair. The data is captured at the crossing point of DDRC_DQS_DP[17:0] and its DDRC_DQS_DN[17:0] during read and write transactions. Different numbers of strobes are used depending on whether the connected DRAMs are x4,x8 or have checkbits.
DDRC_ECC[7:0]	I/O	Check Bits - An Error Correction Code is driven along with data on these lines for DIMMs that support that capability.
DDRC_MA[15:0]	O	Memory Address: These signals are used to provide the multiplexed row and column address to the SDRAM.
DDRC_MA_PAR	O	Odd parity across address and command.
DDRC_ODT[3:0]	O	On Die Termination: Active Termination Control. Enables various combinations of termination resistance in the target and non-target DIMMs when data is read or written.
DDRC_PAR_ERR[2:0]#	I	Parity Error detected by Registered DIMM (one per DIMM).
DDRC_RAS#	O	RAS Control Signal: Used with DDRC_CAS# and DDRC_WE# (along with DDRC_CS#) to define the SRAM commands.
DDRC_RESET#	O	Resets DRAMs. Held low on power up, held high during self refresh, otherwise controlled by configuration register.
DDRC_WE#	O	Write Enable Control Signal: Used with DDRC_RAS# and DDRC_CAS# (along with DDRC_CS#) to define the SDRAM commands.

12.1.2.4 System Memory Compensation Signals

Table 139. DDR Miscellaneous Signals

Signal Names	I/O Type	Description
DDR_COMP[2:0]	I	System Memory Compensation: See the *Picket Post: Intel® Xeon® Processor C5500/C3500 Series with the Intel® 3420 Chipset Platform Design Guide (PDG)* for implementation information.

12.1.3 PCI Express* Signals

Table 140. PCI Express Signals

Signal Names	I/O Type	Description
PE_CFG[2:0]	I/O	PCI Express* Port Bifurcation Configuration: 111 = One x16 PCI Express I/O. 110 = Two x8 PCI Express I/O. 101 = Four x4 PCI Express I/O. 100 = Wait for BIOS to configure PCI Express I/O. 011 = One x8 (port 1-2) and two x4 PCI Express I/O. 010 = Two x4 and one x8 (port 3-4) PCI Express I/O. 001 = Reserved. 000 = Reserved.
PE_GEN2_DISABLE#	I/O	PCI Express Gen2 Speed Disable: Will force Gen 1 (2.5 GT/s) negotiation across all Processor PCI Express ports. ***Note:*** Per port speed negiation via BIOS will not override this strap setting.
PE_ICOMPI	Analog	PCI Express current compensation.
PE_ICOMPO	Analog	PCI Express current compensation.
PE_NTBXL	I/O	PCI Express Non-Transparent Bridge Cross Link Configuration. The PE_NTBXL configuration is required when two processor's PCI Express NTB ports are connected together and configured as back to back NTB's. ***Note:*** For PE_NTBXL configuration via BIOS, board level strapping is not required and the PE_NTBXL straps must be left as 'No Connects" on each of the processors.
PE_RBIAS	Analog	PCI Express resistor bias control.
PE_RCOMPO	Analog	PCI Express resistance compensation.
PE_RX_DN[15:0] PE_RX_DP[15:0]	I	PCI Express Receive differential pair.
PE_TX_DN[15:0] PE_TX_DP[15:0]	O	PCI Express Transmit differential pair.

12.1.4 Processor SMBus Signals

Table 141. Processor SMBus Signals

Signal Names	I/O Type	Description
PE_HP_CLK	O	PCI Express Hot Plug SMBus Clock.
PE_HP_DATA	I/O	PCI Express Hot Plug SMBus Address/Data.
SMB_CLK	I/O	SMBus Clock.
SMB_DATA	I/O	SMBus Address/Data.

12.1.5 DMI / ESI Signals

Table 142. DMI / ESI Signals

Signal Names	I/O Type	Description
DMI_COMP	I/O	DMI/ESI Configuration: Pulled to Vss = ESI (AC coupling required). Pulled to processor Vtt = DMI (DC coupling required). ***Note:*** The processor and the PCH must both be configured appropriately to support the same mode of operation.
DMI_PE_CFG#	I/O	DMI/ESI or PCI Express Configuration: No Connect = x4 interface set as DMI/ESI for the legacy (boot) processor. This signal has an internal weak 10K pullup that's activated for power on straps. Pulled to Vss = x4 interface set as PCI Express (2.5 GT/s) on the non-legacy (application) processor. ***Note:*** DMI/ESI is not supported on the non-legacy (application) processor. ***Note:*** PCI Express is not supported on the legacy (boot) processor.
DMI_PE_RX_DN[3:0] DMI_PE_RX_DP[3:0]	I	DMI/ESI input from PCH: receive differential pair. DMI is when DC coupling is used and DMI_COMP signal is set to DMI. ESI is when AC coupling is used and DMI_COMP signal is set to ESI.
DMI_PE_TX_DN[3:0] DMI_PE_TX_DP[3:0]	O	DMI/ESI output to PCH: Direct Media Interface transmit differential pair. DMI is when DC coupling is used. ESI is when AC coupling is used.

12.1.6 Clock Signals

Table 143. PLL Signals

Signal Names	I/O Type	Description
BCLK_BUF_DN BCLK_BUF_DP	O	Differential bus clock output from the processor. Reserved for possible future use.
BCLK_DN BCLK_DP	I	Differential bus clock input to the processor.
BCLK_ITP_DN BCLK_ITP_DP	O	Buffered differential bus clock pair to ITP.
PE_CLK_DN PE_CLK_DP	I	Differential PCI Express / DMI Clock In: These pins receive a 100-MHz Serial Reference clock from an external clock synthesizer. This clock is used to generate the clocks necessary for the support of PCI Express and DMI.

12.1.7 Reset and Miscellaneous Signals

Table 144. Miscellaneous Signals

Signal Names	I/O Type	Description
DP_SYNCRST#	I/O	Dual Processor Synchronous Reset: signal driven from the legacy (boot) processor to the non-legacy (application) processor. This signal is only needed for a dual socket configuration.
COMP0	I	Must be termianted on the system board using precision resistor.
EKEY_NC		Used to prevent damage to an unsupported processor if plugged into the platform. No-Connect
EXTSYSTRG	I/O	External System Trigger. Debug trigger input mechanism.
PM_SYNC	I	Power Management Sync: A sideband signal to communicate power management status from the platform to the processor.
RSTIN#	I	Reset In: When asserted this signal will asynchronously reset the processor logic. This signal is connected to the PLTRST# output of the PCH.
DDR_ADR	I	Asynchronous DRAM Refresh: When asserted this signal will cause the processor to go into Asynchronous DRAM Refresh.

12.1.8 Thermal Signals

Table 145. Thermal Signals (Sheet 1 of 2)

Signal Names	I/O Type	Description
CATERR#	I/O	Catastrophic Error: This signal indicates that the system has experienced a catastrophic error and cannot continue to operate. The processor will set this for non-recoverable machine check errors or other unrecoverable internal errors.
PECI	I/O	PECI (Platform Environment Control Interface) is the serial sideband interface to the processor and is used primarily for thermal, power and error management. Details regarding the PECI electrical specifications, protocols and functions can be found in the *Platform Environment Control Interface Specification*.
PECI_ID#	I	PECI client address identifier. Assertion (active low) of this pin results in a PECI client address of 0x31 (versus the default 0x30 client address when pulled high). This pin is primarily useful for PECI client address differentiation in DP platforms and must be pulled up to VTT on one socket and down to VSS on the other. Single-socket platforms should always pull this pin high.
DDR_THERM#	I	External Thermal Sensor Input: If the system temperature reaches a dangerously high value then this signal can be used to trigger the start of system memory throttling.
PROCHOT#	I/O	PROCHOT# will go active when the processor temperature monitoring sensor(s) detects that the processor has reached its maximum safe operating temperature. This indicates that the processor Thermal Control Circuit has been activated, if enabled. This signal can also be driven to activate the Thermal Control Circuit. This signal does not have on-die termination and must be terminated on the system board.

Table 145. Thermal Signals (Sheet 2 of 2)

Signal Names	I/O Type	Description
PSI#	O	Processor Power Status Indicator: This signal is asserted when maximum possible processor core current consumption is less than 20 A. Assertion of this signal is an indication that the VR controller does not currently need to be able to provide ICC above 20 A, and the VR controller can use this information to move to more efficient operating point. This signal will de-assert at least 3.3 us before the current consumption will exceed 20 A. The minimum PSI# assertion and de-assertion time is 1 BCLK.
SYS_ERR_STAT[2:0]#	O	Error output signals: Three signals per partition. Minimum assertion time is 12 cycles.
THERMTRIP#	O	Thermal Trip: The processor protects itself from catastrophic overheating by use of an internal thermal sensor. This sensor is set well above the normal operating temperature to ensure that there are no false trips. The processor will stop all execution when the junction temperature exceeds approximately 125 C. This is signaled to the system by the THERMTRIP# pin. See the appropriate platform design guide for termination requirements. Once activated, THERMTRIP# remains latched until RSTIN# is asserted. While the assertion of the RSTIN# signal may de-assert THERMTRIP#, if the processor's junction temperature remains at or above the trip level, THERMTRIP# will again be asserted after RSTIN# is de-asserted.

12.1.9 Processor Core Power Signals

Table 146. Power Signals (Sheet 1 of 2)

Signal Names	I/O Type	Description
ISENSE	Analog	Current sense from VRD11.1 Compliant Regulator to the processor core.
VCC	Analog	Processor core power supply. The voltage supplied to these pins is determined by the VID pins.
VCC_SENSE	Analog	VCC_SENSE and VSS_SENSE provide an isolated, low impedance connection to the processor core voltage and ground. They can be used to sense or measure voltage near the silicon.
VCCPLL		VCCPLL provides isolated power for internal processor PLLs.
VDDQ		Processor I/O supply voltage for DDR3.
VID[7:0]	I/O	VID[7:0] (Voltage ID) are used to support automatic selection of power supply voltages (VCC). See the appropriate platform design guide or Voltage Regulator-Down (VRD) 11.1 Design Guidelines for more information. The voltage supply for these signals must be valid before the VR can supply VCC to the processor. Conversely, the VR output must be disabled until the voltage supply for the VID signals become valid. The VR must supply the voltage that is requested by the signals, or disable itself. VID7 and VID6 should be tied to Vss via a 1k resistor during reset (This value is latched on the rising edge of VTTPWRGOOD).
VSS	Analog	VSS are the ground pins for the processor and should be connected to the system ground plane.
VSS_SENSE	Analog	VCC_SENSE and VSS_SENSE provide an isolated, low impedance connection to the processor core voltage and ground. They can be used to sense or measure voltage near the silicon.
VSS_SENSE_VTT	Analog	VTT_SENSE and VSS_SENSE_VTT provide an isolated, low impedance connection to the processor VTT voltage and ground. They can be used to sense or measure voltage near the silicon.
VTTA	Analog	Processor power for the memory controller, shared cache and I/O (1.1 V).

Table 146. Power Signals (Sheet 2 of 2)

Signal Names	I/O Type	Description
VTTD	Analog	Processor power for the memory controller, shared cache and I/O (1.1 V).
VTTD_SENSE	Analog	VTTD_SENSE and VSS_SENSE_VTT provide an isolated, low impedance connection to the processor VTT voltage and ground. They can be used to sense or measure voltage near the silicon.
VTT_VID[4:2]	O	VTT_VID[4:2] is used to support automatic selection of power supply voltages (V_{TT}). The VR must supply the voltage that is requested by the signal after VTTPWRGOOD is asserted. Before VTTPWRGOOD is asserted, the VRM must supply a safe “boot voltage”.

12.1.10 Power Sequencing Signals

Table 147. Reset Signals

Signal Names	I/O Type	Description
DDR_DRAMPWROK	I	DDR_DRAMPRWOK processor input: connects to PCH DRAMPWROK.
SKTOCC#	O	SKTOCC# (Socket Occupied): will be pulled to ground on the processor package. There is no connection to the processor silicon for this signal. System board designers may use this signal to determine if the processor is present.
VCCPWRGOOD	I	VCCPWRGOOD (Power Good) Processor Input: The processor requires these signals to be a clean indication that VCC, VCCPLL, VCCA, VTT supplies are stable and within their specifications and that BCLK is stable and has been running for a minimum number of cycles. 'Clean' implies that the signal will remain low (capable of sinking leakage current), without glitches, from the time that the power supplies are turned on until they come within specification. These signals must then transition monotonically to a high state. These signals can be driven inactive at any time, but BCLK and power must again be stable before a subsequent rising edge of VCCPWRGOOD. These signals should be tied together and connected to the CPUPWRGD output signal of the PCH.
VTTPWRGOOD	I	The processor requires this input signal to be a clean indication that the VTT power supply is stable and within specifications. 'Clean' implies that the signal will remain low (capable of sinking leakage current), without glitches, from the time that the power supplies are turned on until they come within specification. The signal must then transition monotonically to a high state. It is not valid for VTTPWRGOOD to be deasserted while VCCPWRGOOD is asserted.

12.1.11 No Connect and Reserved Signals

Table 148. No Connect Signals

Signal Names	Description
NC_x	This signal must be left unconnected.
RSVD_x	Reserved Signals. Signal can be left unconnected or routed to a test point.

12.1.12 ITP Signals

Table 149. ITP Signals

Signal Names	I/O Type	Description
BPM[7:0]#	I/O	Breakpoint and Performance Monitor Signals: Outputs from the processor that indicate the status of breakpoints and programmable counters used for monitoring processor performance.
PRDY#	O	PRDY# is a processor output used by debug tools to determine processor debug readiness.
PREQ#	I	PREQ# is used by debug tools to request debug operation of the processor.
TCLK	I	TCK (Test Clock) provides the clock input for the processor Test Bus (also known as the Test Access Port).
TDI	I	TDI (Test Data In) transfers serial test data into the CPU.
TDI_M	I	TDI_M (Test Data In) transfers serial test data into the processor.
TDO	O	TDO (Test Data Out) transfers serial test data out of the CPU.
TDO_M	O	TDO_M (Test Data Out) transfers serial test data out of the processor. Note: One of the TDI pin needs to be connected to one of the TDO pins on the board.
TMS	I	TMS (Test Mode Select) is a JTAG specification support signal used by debug tools.
TRST#	I	TRST# (Test Reset) resets the Test Access Port (TAP) logic. TRST# must be driven low during power on Reset. .

12.2 Physical Layout and Signals

The full signal map is provided in Table 150, Table 151, and Table 152.

Table 153 provides an alphabetical listing of all signal locations. Table 154 provides an alphabetical listing of all processor signals.

Table 150. Physical Layout, Left Side (Sheet 1 of 3)

	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29
BA				RSVD_BA40	VSS	PE_TX_DP[5]	PE_TX_DN[3]	PE_TX_DP[3]	DMI_PE_CFG#					VCC	VSS
AY		VSS	RSVD_AY41	RSVD_AY40	PE_TX_DP[6]	PE_TX_DN[5]	VSS	PE_TX_DN[2]	NC_AY35	PE_CFG[2]	PE_CFG[0]	VSS	VCC	VCC	VSS
AW		RSVD_AW42	RSVD_AW41	PE_TX_DP[7]	PE_TX_DN[6]	PE_TX_DN[4]	PE_TX_DP[4]	PE_TX_DP[2]	VSS	NC_AW34	DMI_COMP	VSS	VCC	VCC	VSS
AV	RSVD_AV43	RSVD_AV42	VSS	PE_TX_DN[7]	VSS	PE_TX_DP[8]	PE_TX_DN[1]	PE_TX_DP[1]	VSS	PE_CFG[1]	PE_GEN2_DISABLE#	VSS	VCC	VCC	VSS
AU	VSS	RSVD_AU42	PE_RBIAS	VSS	PE_TX_DP[9]	PE_TX_DN[8]	RSVD_AU37	VSS	PE_TX_DN[0]	QPI_COMP[1]	NC_AU33	VSS	VCC	VCC	VSS
AT	PE_TX_DP[10]	RSVD_AT42	VSS	PE_CLK_DP	PE_TX_DN[9]	VSS	VSS	DP_SYNCRST#	PE_TX_DP[0]	VSS	PE_NTBXL	VSS	VCC	VCC	VSS
AR	PE_TX_DN[10]	PE_TX_DN[11]	PE_TX_DP[11]	PE_CLK_DN	VSS	PE_TX_DP[14]	RSVD_AR37	SMB_CLK	SMB_DATA	RSVD_AR34	VSS	VSS	VCC	VCC	VSS
AP	VSS	PE_TX_DP[12]	PE_TX_DN[13]	PE_TX_DP[13]	PE_TX_DN[15]	PE_TX_DN[14]	VSS	SYS_ERR_STAT[1]#	RSVD_AP35	PE_HP_DATA	PE_HP_CLK	VSS	VCC	VCC	VSS
AN	PE_ICOMPI	PE_TX_DN[12]	VSS	RSVD_AN40	PE_TX_DP[15]	RSVD_AN38	PE_RX_DN[1]	PM_SYNC	VSS	VSS	RSVD_AN33	VSS	VCC	VCC	VSS
AM	PE_ICOMPO	VSS	RSVD_AM41	RSVD_AM40	VSS	PE_RX_DN[2]	PE_RX_DP[1]	SKTOCC#	SYS_ERR_STAT[0]#	DDR_ADR	EXTSYSTRG	VSS	VCC	VCC	VSS
AL	PE_RCOMPO	PE_RX_DP[6]	PE_RX_DN[4]	PE_RX_DP[4]	VSS	PE_RX_DP[2]	VSS	RSVD_AL36	PROCHOT#	SYS_ERR_STAT[2]#	TDO_M	VSS	VCC	VCC	VSS
AK	VSS	PE_RX_DN[6]	VSS	PE_RX_DN[5]	PE_RX_DN[0]	PE_RX_DN[3]	PE_RX_DP[3]	VSS	PECI_ID#	VSS	VCC	VSS	VCC	VCC	VSS
AJ	PE_RX_DP[9]	PE_RX_DN[7]	PE_RX_DP[7]	PE_RX_DP[5]	VSS	PE_RX_DP[0]	NC_AJ37	RSTIN#	BCLK_DN	VCC	VCC				
AH	PE_RX_DN[9]	VSS	PE_RX_DP[10]	VSS	PE_RX_DP[8]	VSS	VTTPWRGOOD	PECI	BCLK_DP	VSS	TDI_M				
AG	PE_RX_DP[11]	PE_RX_DN[11]	PE_RX_DN[10]	PE_RX_DP[13]	PE_RX_DN[8]	VSS	THERMTRIP#	EKEY_NC	VSS	VTTA	VSS				
AF	VSS	PE_RX_DP[12]	PE_RX_DN[12]	PE_RX_DN[13]	VSS	PE_RX_DP[15]	VTTD	VTTD	VSS	VTTA	VTTA				

Table 150. Physical Layout, Left Side (Sheet 2 of 3)

	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29
AE	DMI_PE_TX_DP [0]	DMI_PE_TX_DN [0]	VSS	PE_RX_DP [14]	PE_RX_DN [14]	PE_RX_DN [15]	VSS_SENSE_VTTD	VTTD_SENSE	VTTD	VTTD	VTTA				
AD	VSS	DMI_PE_TX_DP [2]	DMI_PE_TX_DP [1]	DMI_PE_TX_DN [1]	VSS	CATERR#	VSS	VTTD	VTTD	VTTD	VSS				
AC	DMI_PE_TX_DP [3]	DMI_PE_TX_DN [2]	COMP0	VSS	DMI_PE_RX_DP [2]	DMI_PE_RX_DN [3]	DMI_PE_RX_DP [3]	VSS	VTTD	VTTD	VTTD				
AB	DMI_PE_TX_DN [3]	VSS	DMI_PE_RX_DP [0]	DMI_PE_RX_DP [1]	DMI_PE_RX_DN [1]	DMI_PE_RX_DN [2]	VSS	DDRB_DQ [5]	DDRB_DQ [4]	VTTD	VTTD				
AA			DMI_PE_RX_DN [0]	VSS	VSS	DDRB_DQS_DN [9]	DDRB_DQS_DP [9]	DDRB_DQ [0]	DDRB_DQ [1]	VSS	VTTD				
Y			VSS	DDRB_DQ [6]	DDRB_DQ [7]	DDRB_DQS_DP [0]	DDRB_DQS_DN [0]	VSS	DDRB_DQ [2]	DDRB_DQ [3]	VSS				
W	VSS	DDRA_DQ [5]	DDRA_DQ [0]	DDRA_DQ [4]	DDRC_DQ [12]	VSS	DDRC_DQS_DP [0]	DDRC_DQS_DN [0]	DDRC_DQ [1]	DDRC_DQ [0]	VCCPLL				
V	DDRA_DQS_DP [9]	DDRA_DQS_DN [9]	DDRA_DQ [1]	VSS	DDRC_DQ [13]	DDRC_DQ [7]	DDRC_DQ [6]	DDRC_DQ [2]	VSS	DDRC_DQ [5]	VCCPLL				
U	DDRA_DQS_DN [0]	VSS	DDRA_DQ [6]	DDRC_DQS_DP [10]	DDRC_DQ [9]	DDRC_DQ [8]	VSS	DDRC_DQ [3]	DDRC_DQS_DP [9]	DDRC_DQ [4]	VCCPLL				
T	DDRA_DQS_DP [0]	DDRA_DQ [7]	DDRC_DQ [14]	DDRC_DQS_DN [10]	VSS	DDRC_DQS_DN [1]	DDRC_DQS_DP [1]	DDRC_DQ [11]	DDRC_DQS_DN [9]	VSS	VCC				
R	DDRA_DQ [2]	DDRA_DQ [3]	VSS	DDRC_DQ [15]	DDRC_DQ [10]	DDRB_DQS_DP [1]	DDRB_DQS_DN [1]	VSS	DDRB_DQ [13]	DDRB_DQ [12]	VCC				
P	VSS	DDRA_DQ [12]	DDRA_DQ [13]	DDRC_DQ [20]	DDRB_DQ [10]	VSS	DDRB_DQS_DN [10]	DDRB_DQS_DP [10]	DDRB_DQ [9]	DDRB_DQ [8]	VSS				
N	DDRA_DQ [9]	DDRA_DQS_DP [10]	DDRA_DQ [8]	VSS	DDRB_DQ [11]	DDRB_DQ [15]	DDRB_DQ [14]	DDRC_DQ [21]	VSS	DDRB_DQ [20]	VCC				
M	DDRA_DQS_DN [10]	VSS	DDRA_DQS_DN [1]	DDRC_DQ [17]	DDRC_DQ [16]	DDRC_DQS_DP [11]	VSS	DDRB_DQ [21]	DDRB_DQ [16]	DDRB_DQ [17]	VCC	VSS	VCC	VSS	VCC

Table 150. Physical Layout, Left Side (Sheet 3 of 3)

	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29
L	DDR A_DQ [14]	DDR A_DQ [15]	DDR A_DQS_DP [1]	DDR C_DQ [22]	VSS	DDR C_DQS_DN [11]	DDR B_DQS_DP [11]	DDR B_DQS_DN [2]	DDR B_DQS_DP [2]	VSS	DDR B_DQ [25]	DDR B_DQ [30]	DDR B_DQS_DN [3]	DDR B_DQS_DP [3]	VSS
K	DDR A_DQ [11]	DDR A_DQ [10]	VSS	DDR C_DQS_DP [2]	DDR C_DQS_DN [2]	DDR C_DQ [23]	DDR B_DQS_DN [11]	VSS	DDR B_DQ [18]	DDR B_DQS_DP [12]	DDR B_DQS_DN [12]	DDR B_DQ [26]	VSS	DDR B_DQ [31]	VSS
J	VSS	DDR A_DQ [20]	DDR A_DQ [21]	DDR C_DQ [18]	DDR C_DQ [19]	VSS	DDR C_DQ [26]	DDR B_DQ [22]	DDR B_DQ [19]	DDR B_DQ [28]	VSS	DDR B_DQ [27]	DDR C_ECC[4]	DDR C_ECC[5]	VSS
H	DDR A_DQ [17]	DDR A_DQS_DP [11]	DDR A_DQ [16]	VSS	DDR C_DQ [28]	DDR C_DQS_DP [12]	DDR C_DQ [27]	DDR B_DQ [23]	VSS	DDR B_DQ [29]	DDR B_DQ [24]	DDR C_ECC[0]	DDR C_DQS_DP [17]	VSS	VSS
G	DDR A_DQS_DN [11]	VSS	DDR A_DQS_DN [2]	DDR C_DQ [24]	DDR C_DQ [29]	DDR C_DQS_DN [12]	VSS	DDR B_ECC[3]	DDR B_ECC[7]	DDR B_DQS_DN [8]	DDR B_DQS_DP [8]	VSS	DDR C_DQS_DN [17]	DDR C_DQS_DN [8]	DDR C_DQS_DP [8]
F	DDR A_DQ [22]	DDR A_DQ [23]	DDR A_DQS_DP [2]	DDR C_DQ [25]	VSS	DDR C_DQ [30]	DDR B_ECC[5]	DDR B_ECC[1]	DDR B_DQS_DP [17]	VSS	DDR C_ECC[1]	DDR A_ECC[2]	DDR C_ECC[6]	DDR C_ECC[7]	VSS
E	DDR A_DQ [19]	DDR A_DQ [18]	VSS	DDR C_DQS_DN [3]	DDR C_DQS_DP [3]	DDR C_DQ [31]	DDR B_ECC[4]	VSS	DDR B_DQS_DN [17]	DDR B_ECC[6]	DDR B_ECC[2]	DDR C_RESET#	VDDQ	DDR C_ECC[3]	DDR C_ECC[2]
D	VSS	DDR A_DQ [29]	DDR A_DQ [28]	DDR A_DQ [24]	DDR A_DQS_DP [12]	VSS	DDR A_DQ [27]	DDR B_ECC[0]	DDR A_DQS_DN [8]	DDR A_DQS_DP [8]	VSS	DDR A_RESET#	VSS	VSS	DDR B_RESET#
C	VSS	PREQ #	DDR A_DQ [25]	VSS	DDR A_DQS_DN [12]	DDR A_DQ [30]	DDR A_ECC[4]	DDR A_ECC[0]	VSS	DDR A_ECC[7]	DDR A_ECC[3]	VSS	VSS	VDDQ	DDR A_CKE[0]
B		VSS	PRDY #	DDR A_DQS_DN [3]	DDR A_DQS_DP [3]	DDR A_DQ [31]	VSS	DDR A_DQS_DP [17]	DDR A_DQS_DN [17]	DDR A_ECC[6]	NC_B33	VDDQ	DDR A_CKE[3]	DDR A_CKE[2]	DDR A_MA [15]
A			VSS	RSVD_A40	VSS	DDR A_DQ [26]	DDR A_ECC[5]	DDR A_ECC[1]	VSS				VSS	DDR A_CKE[1]	VDDQ
	43	42	41	40	39	38	37	36	35	34	33	32	31	30	29

Table 151. Physical Layout, Center (Sheet 1 of 3)

	28	27	26	25	24	23	22	21	20	19	18	17	16	15
BA	VCC	VCC	VSS	VCC	VCC				VSS	VCC	VCC	VSS	VCC	VCC
AY	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AW	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AV	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AU	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AT	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AR	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AP	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AN	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AM	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AL	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AK	VCC	VCC	VSS	VCC	VCC	VSS	VSS	VCC	VSS	VCC	VCC	VSS	VCC	VCC
AJ														
AH														
AG														
AF														
AE														
AD														
AC														

Table 151. Physical Layout, Center (Sheet 2 of 3)

	28	27	26	25	24	23	22	21	20	19	18	17	16	15
AB														
AA														
Y														
W														
V														
U														
T														
R														
P														
N														
M	VSS	VDDQ	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VDDQ	VSS	VCC
L	DDRB_MA[3]	DDRC_CKE[3]	DDRC_BA[2]	DDRC_MA[8]	VDDQ	RSVD_L23	DDRC_CLK_DP[3]	DDRC_CLK_DN[3]	DDRC_CLK_DP[1]	VDDQ	DDRB_CLK_DN[2]	DDRC_CS[6]#	DDRC_ODT[0]	RSVD_L15
K	DDRB_MA[4]	VSS	VDDQ	NC_K25	RSVD_K24	DDRC_MA[5]	DDRC_MA[6]	VDDQ	DDRC_CLK_DN[1]	DDRA_CLK_DN[0]	DDRB_CLK_DP[2]	DDRC_MA[1]	VDDQ	RSVD_K15
J	VDDQ	DDRB_MA[6]	DDRC_CKE[0]	DDRC_PAR_ERR[1]#	DDRC_MA[7]	VDDQ	DDRC_CLK_DP[0]	DDRC_CLK_DN[0]	DDRC_MA[3]	DDRA_CLK_DP[0]	VDDQ	DDRB_MA[2]	DDRB_MA[1]	DDRC_CS[7]#
H	DDRB_CKE[0]	DDRB_BA[2]	DDRB_MA[14]	VDDQ	DDRC_MA[14]	DDRC_MA[11]	DDRC_MA[9]	DDRC_CLK_DP[2]	VDDQ	DDRB_CLK_DN[3]	DDRB_CLK_DP[3]	DDRC_MA[10]	DDRC_CS[3]#	VDDQ
G	VSS	VDDQ	DDRC_CKE[1]	DDRC_MA[15]	DDRB_MA[9]	DDRC_MA[12]	VDDQ	DDRC_CLK_DN[2]	DDRB_CLK_DN[1]	DDRB_CLK_DP[1]	DDRC_MA[2]	VDDQ	DDRC_CS[0]#	DDRA_CS[0]#
F	VSS	NC_F27	DDRB_MA[15]	DDRB_PAR_ERR[2]#	VDDQ	DDRC_PAR_ERR[2]#	DDRB_MA[5]	DDRC_PAR_ERR[0]#	DDRC_MA[4]	VDDQ	DDRA_CLK_DP[2]	DDRC_BA[1]	DDRC_CAS#	DDRC_MA[13]

Table 151. Physical Layout, Center (Sheet 3 of 3)

	28	27	26	25	24	23	22	21	20	19	18	17	16	15
E	VSS	DDRB_CKE[1]	VDDQ	DDRB_PAR_ERR[1]#	DDRB_MA[12]	DDRB_MA[11]	DDRB_MA[8]	VDDQ	DDRA_CLK_DP[3]	DDRA_CLK_DN[3]	DDRA_CLK_DN[2]	DDRC_CS[4]#	VDDQ	DDRB_CS[2]#
D	VDDQ	DDRB_CKE[2]	DDRC_CKE[2]	DDRA_PAR_ERR[0]#	DDRA_MA[3]	VDDQ	DDRB_MA[7]	DDRB_CLK_DN[0]	DDRB_MA_PAR	DDRA_CLK_DP[1]	VDDQ	DDRC_RAS#	DDRC_CS[2]#	DDRC_ODT[2]
C	DDRA_BA[2]	DDRB_CKE[3]	DDRA_MA[9]	VDDQ	DDRA_MA[6]	DDRA_MA[2]	DDRB_PAR_ERR[0]#	DDRB_CLK_DP[0]	VDDQ	DDRA_CLK_DN[1]	DDRB_BA[0]	DDRB_CS[4]#	DDRC_WE#	VDDQ
B	DDRA_PAR_ERR[1]#	VDDQ	DDRA_MA[12]	DDRA_MA[8]	DDRA_MA[5]	DDRA_MA[4]	VDDQ	DDRA_MA[1]	DDRA_MA_PAR	DDRA_MA[10]	DDRC_MA_PAR	VDDQ	DDRA_BA[0]	DDRA_CS[4]#
A	DDRA_MA[14]	DDRA_PAR_ERR[2]#	DDRA_MA[11]	DDRA_MA[7]	VDDQ				DDRA_MA[0]	VDDQ	DDRC_MA[0]	DDRC_BA[0]	DDRA_BA[1]	DDRA_RAS#
	28	27	26	25	24	23	22	21	20	19	18	17	16	15

Table 152. Physical Layout, Right (Sheet 1 of 3)

14	13	12	11	10	9	8	7	6	5	4	3	2	1	
VSS	VCC	VCC	VSS	VCC	VCC	QPI_RX_DN[2]	QPI_RX_DP[4]	QPI_RX_DN[4]	VSS	RSVD_BA4	VSS			BA
VSS	VCC	VCC	VSS	VCC	VCC	QPI_RX_DP[2]	VSS	QPI_RX_DP[5]	QPI_RX_DN[5]	RSVD_AY4	RSVD_AY3	VSS		AY
VSS	VCC	VCC	VSS	VCC	VCC	VSS	QPI_RX_DN[1]	VSS	QPI_RX_DN[3]	QPI_RX_DP[7]	QPI_RX_DN[7]	RSVD_AW2	VSS	AW
VSS	VCC	VCC	VSS	VCC	VCC	QPI_RX_DN[0]	QPI_RX_DP[1]	VTT_VID[4]	QPI_RX_DP[3]	VSS	VTT_VID[2]	RSVD_AV2	RSVD_AV1	AV
VSS	VCC	VCC	VSS	VCC	VCC	QPI_RX_DP[0]	QPI_RX_DP[6]	QPI_RX_DN[6]	VSS	QPI_RX_DP[8]	QPI_RX_DN[8]	RSVD_AU2	VSS	AU
VSS	VCC	VCC	VSS	VCC	VCC	VSS	VSS	QPI_CLKRX_DP	RSVD_AT5	RSVD_AT4	QPI_RX_DP[9]	QPI_RX_DN[9]	QPI_RX_DP[10]	AT
VSS	VCC	VCC	VSS	VCC	VCC_SENSE	VSS_SENSE	VCCPWRGOOD	QPI_CLKRX_DN	QPI_RX_DN[11]	QPI_RX_DP[11]	VSS	VSS	QPI_RX_DN[10]	AR
VSS	VCC	VCC	VSS	VSS	VID[5]	VID[6]	PSI#	VSS	VSS	QPI_RX_DN[15]	QPI_RX_DP[15]	QPI_RX_DP[12]	VSS	AP
VSS	VCC	VCC	VSS	VID[4]	VID[2]	VID[7]	VSS	QPI_RX_DN[17]	QPI_RX_DP[17]	QPI_RX_DN[16]	VSS	QPI_RX_DN[12]	QPI_RX_DP[13]	AN
VSS	VCC	VCC	VSS	VID[3]	VSS	QPI_RX_DP[19]	QPI_RX_DN[18]	QPI_RX_DP[18]	VSS	QPI_RX_DP[16]	QPI_RX_DN[14]	QPI_RX_DP[14]	QPI_RX_DN[13]	AM
VSS	VCC	VCC	VSS	VID[0]	VID[1]	QPI_RX_DN[19]	VSS	QPI_COMP[0]	RSVD_AL5	RSVD_AL4	NC_AL3	VSS	VSS	AL
VSS	VCC	VCC	VCC	VSS	VSS	ISENSE	VSS	QPI_TX_DP[3]	QPI_TX_DN[3]	QPI_TX_DN[4]	VSS	RSVD_AK2	QPI_TX_DP[7]	AK
			VCC	TDO	TDI	QPI_TX_DP[1]	QPI_TX_DN[1]	QPI_TX_DN[2]	VSS	QPI_TX_DP[4]	QPI_TX_DP[6]	QPI_TX_DN[6]	QPI_TX_DN[7]	AJ
			VCC	TCLK	TRST#	QPI_TX_DN[0]	VSS	QPI_TX_DP[2]	RSVD_AH5	QPI_TX_DN[8]	QPI_TX_DP[8]	QPI_TX_DP[9]	VSS	AH
			VSS	TMS	VSS	QPI_TX_DP[0]	QPI_TX_DP[5]	QPI_TX_DN[5]	RSVD_AG5	RSVD_AG4	VSS	QPI_TX_DN[9]	BCLK_BUF_DP	AG
			VTTA	NC_AF10	VTTD	VTTD	VTT_VID[3]	QPI_CLKTX_DP	VSS	RSVD_AF4	QPI_TX_DN[10]	QPI_TX_DP[10]	BCLK_BUF_DN	AF
			VTTA	VTTA	VTTD	VTTD	VSS	QPI_CLKTX_DN	QPI_TX_DN[18]	QPI_TX_DN[14]	QPI_TX_DP[14]	VSS	QPI_TX_DP[11]	AE
			VSS	VTTA	VTTD	QPI_TX_DN[19]	QPI_TX_DN[17]	QPI_TX_DP[17]	QPI_TX_DP[18]	QPI_TX_DN[15]	QPI_TX_DN[12]	QPI_TX_DP[12]	QPI_TX_DN[11]	AD

Table 152. Physical Layout, Right (Sheet 2 of 3)

14	13	12	11	10	9	8	7	6	5	4	3	2	1	
			VTTD	VTTD	VSS	QPI_TX_DP[19]	VSS	QPI_TX_DN[16]	VSS	QPI_TX_DP[15]	QPI_TX_DP[13]	VSS	DDR_COMP[2]	AC
			VTTD	VTTD	VTTD	VTTD	VSS	QPI_TX_DP[16]	DDR_THERM#	VSS	QPI_TX_DN[13]			AB
			VTTD	VTTD	VSS	DDR_COMP[0]	DDRB_DQ[62]	DDR_DRAMPWROK	BCLK_ITP_DP	BCLK_ITP_DN	VSS			AA
			VSS	DDRB_DQ[58]	DDRB_DQS_DN[7]	DDRB_DQS_DP[7]	DDR_COMP[1]	VSS	DDRB_DQS_DN[16]	DDRB_DQS_DP[16]	DDRA_DQ[59]	DDRA_DQ[58]	VSS	Y
			VCC	DDRB_DQ[59]	DDRB_DQ[63]	VSS	DDRB_DQ[57]	DDRB_DQ[56]	DDRB_DQ[61]	DDRA_DQ[63]	VSS	DDRA_DQS_DP[7]	DDRA_DQS_DN[7]	W
			NC_V11	VSS	DDRB_DQ[60]	DDRC_DQ[62]	DDRC_DQS_DN[16]	DDRC_DQS_DP[16]	VSS	DDRA_DQ[62]	DDRA_DQS_DN[16]	DDRA_DQS_DP[16]	DDRA_DQ[57]	V
			NC_U11	DDRC_DQ[59]	DDRC_DQ[63]	DDRC_DQS_DP[7]	VSS	DDRC_DQ[57]	DDRC_DQ[56]	DDRA_DQ[56]	DDRA_DQ[61]	VSS	DDRA_DQ[60]	U
			VCC	DDRC_DQ[58]	VSS	DDRC_DQS_DN[7]	DDRC_DQ[61]	DDRC_DQ[60]	DDRB_DQ[51]	VSS	DDRA_DQ[55]	DDRA_DQ[51]	DDRA_DQ[50]	T
			VCC	DDRC_DQ[54]	DDRC_DQ[55]	DDRB_DQ[54]	DDRB_DQ[55]	VSS	DDRB_DQ[50]	DDRA_DQ[54]	DDRA_DQS_DN[6]	DDRA_DQS_DP[6]	VSS	R
			VSS	DDRC_DQ[51]	DDRC_DQ[50]	VSS	DDRC_DQ[48]	DDRC_DQS_DP[6]	DDRC_DQS_DN[6]	DDRC_DQS_DN[15]	VSS	DDRA_DQS_DP[15]	DDRA_DQS_DN[15]	P
			VCC	VSS	DDRC_DQ[43]	DDRC_DQ[52]	DDRC_DQ[53]	DDRC_DQ[49]	VSS	DDRC_DQS_DP[15]	DDRA_DQ[53]	DDRA_DQ[49]	DDRA_DQ[48]	N
VSS	VCC	VSS	VCC	DDRC_DQ[45]	DDRC_DQ[42]	DDRC_DQ[47]	VSS	DDRB_DQ[53]	DDRB_DQS_DP[15]	DDRB_DQS_DN[15]	DDRA_DQ[52]	VSS	DDRA_DQ[43]	M
VDDQ	DDRC_DQ[35]	DDRC_DQ[39]	DDRC_DQ[44]	DDRC_DQ[40]	VSS	DDRC_DQ[46]	DDRC_DQS_DP[5]	DDRB_DQS_DP[6]	DDRB_DQS_DN[6]	VSS	DDRA_DQ[46]	DDRA_DQ[47]	DDRA_DQ[42]	L
DDRC_CS[1]#	DDRB_BA[1]	DDRC_DQ[32]	VSS	DDRC_DQ[41]	DDRC_DQS_DP[14]	DDRC_DQS_DN[14]	DDRC_DQS_DN[5]	VSS	DDRB_DQ[49]	DDRB_DQ[48]	DDRA_DQS_DN[5]	DDRA_DQS_DP[5]	VSS	K
DDRB_MA[0]	VSS	DDRC_DQ[33]	DDRC_DQS_DN[13]	DDRC_DQS_DP[4]	DDRC_DQS_DN[4]	VSS	DDRB_DQS_DN[14]	DDRB_DQ[41]	DDRB_DQ[47]	DDRB_DQ[52]	VSS	DDRA_DQS_DP[14]	DDRA_DQS_DN[14]	J

Table 152. Physical Layout, Right (Sheet 3 of 3)

14	13	12	11	10	9	8	7	6	5	4	3	2	1	
DDRB_MA[10]	DDRC_DQ[34]	DDRC_DQ[38]	DDRC_DQS_DP[13]	VSS	DDRB_DQ[45]	DDRB_DQ[40]	DDRB_DQS_DP[14]	DDRB_DQS_DP[5]	VSS	DDRB_DQ[43]	DDRA_DQ[45]	DDRA_DQ[40]	DDRA_DQ[41]	H
DDRB_RAS#	DDRB_WE#	VSS	DDRC_DQ[36]	DDRC_DQ[37]	DDRB_DQ[44]	DDRB_DQ[37]	VSS	DDRB_DQS_DN[5]	DDRB_DQ[46]	DDRB_DQ[42]	DDRA_DQ[35]	VSS	DDRA_DQ[44]	G
VDDQ	DDRC_ODT[1]	DDRA_ODT[0]	DDRB_ODT[3]	DDRB_DQ[36]	VSS	DDRB_DQS_DP[13]	DDRB_DQS_DN[13]	DDRB_DQ[39]	DDRB_DQ[35]	VSS	DDRA_DQ[38]	DDRA_DQ[39]	DDRA_DQ[34]	F
DDRB_CAS#	DDRB_CS[3]#	DDRB_CS[7]#	VDDQ	DDRB_CS[5]#	DDRB_DQ[32]	DDRB_DQ[33]	DDRB_DQS_DP[4]	VSS	DDRB_DQ[34]	DDRA_DQS_DN[4]	DDRA_DQS_DP[4]	BPM[7]#	VSS	E
DDRB_ODT[2]	VDDQ	DDRB_CS[0]#	DDRB_ODT[0]	DDRC_ODT[3]	DDRC_CS[5]#	VSS	DDRB_DQS_DN[4]	DDRB_DQ[38]	DDRA_DQS_DP[13]	DDRA_DQS_DN[13]	VSS	BPM[6]#	BPM[4]#	D
DDRB_CS[6]#	DDRA_CS[2]#	DDRA_CAS#	DDRA_CS[6]#	VDDQ	DDRA_ODT[1]	DDRB_ODT[1]	DDRA_ODT[3]	DDRA_DQ[37]	VSS	DDRA_DQ[33]	BPM[5]#	BPM[2]#		C
DDRB_MA[13]	DDRA_WE#	VDDQ	DDRA_ODT[2]	DDRA_CS[1]#	DDRA_CS[3]#	DDRA_CS[7]#	VDDQ	DDRA_DQ[36]	DDRA_DQ[32]	BPM[3]#	BPM[0]#	VSS		B
VDDQ				DDRA_MA[13]	VDDQ	DDRB_CS[1]#	DDRA_CS[5]#	VSS	BPM[1]#	VSS				A
14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Table 153. Alphabetical Listing by X and Y Coordinate

XY Coord	Signal
A4	VSS
A5	BPM[1]#
A6	VSS
A7	DDRA_CS[5]#
A8	DDRB_CS[1]#
A9	VDDQ
A10	DDRA_MA[13]
A14	VDDQ
A15	DDRA_RAS#
A16	DDRA_BA[1]
A17	DDRC_BA[0]
A18	DDRC_MA[0]
A19	VDDQ
A20	DDRA_MA[0]
A24	VDDQ
A25	DDRA_MA[7]
A26	DDRA_MA[11]
A27	DDRA_PAR_ERR[2]#
A28	DDRA_MA[14]
A29	VDDQ
A30	DDRA_CKE[1]
A31	VSS
A35	VSS
A36	DDRA_ECC[1]
A37	DDRA_ECC[5]
A38	DDRA_DQ[26]
A39	VSS
A40	RSVD_A40
A41	VSS
B2	VSS
B3	BPM[0]#
B4	BPM[3]#
B5	DDRA_DQ[32]
B6	DDRA_DQ[36]
B7	VDDQ
B8	DDRA_CS[7]#
B9	DDRA_CS[3]#
B10	DDRA_CS[1]#
B11	DDRA_ODT[2]

XY Coord	Signal
B12	VDDQ
B13	DDRA_WE#
B14	DDRB_MA[13]
B15	DDRA_CS[4]#
B16	DDRA_BA[0]
B17	VDDQ
B18	DDRC_MA_PAR
B19	DDRA_MA[10]
B20	DDRA_MA_PAR
B21	DDRA_MA[1]
B22	VDDQ
B23	DDRA_MA[4]
B24	DDRA_MA[5]
B25	DDRA_MA[8]
B26	DDRA_MA[12]
B27	VDDQ
B28	DDRA_PAR_ERR[1]#
B29	DDRA_MA[15]
B30	DDRA_CKE[2]
B31	DDRA_CKE[3]
B32	VDDQ
B33	NC_B33
B34	DDRA_ECC[6]
B35	DDRA_DQS_DN[17]
B36	DDRA_DQS_DP[17]
B37	VSS
B38	DDRA_DQ[31]
B39	DDRA_DQS_DP[3]
B40	DDRA_DQS_DN[3]
B41	PRDY#
B42	VSS
C2	BPM[2]#
C3	BPM[5]#
C4	DDRA_DQ[33]
C5	VSS
C6	DDRA_DQ[37]
C7	DDRA_ODT[3]
C8	DDRB_ODT[1]
C9	DDRA_ODT[1]
C10	VDDQ
C11	DDRA_CS[6]#

XY Coord	Signal
C12	DDRA_CAS#
C13	DDRA_CS[2]#
C14	DDRB_CS[6]#
C15	VDDQ
C16	DDRC_WE#
C17	DDRB_CS[4]#
C18	DDRB_BA[0]
C19	DDRA_CLK_DN[1]
C20	VDDQ
C21	DDRB_CLK_DP[0]
C22	DDRB_PAR_ERR[0]#
C23	DDRA_MA[2]
C24	DDRA_MA[6]
C25	VDDQ
C26	DDRA_MA[9]
C27	DDRB_CKE[3]
C28	DDRA_BA[2]
C29	DDRA_CKE[0]
C30	VDDQ
C31	VSS
C32	VSS
C33	DDRA_ECC[3]
C34	DDRA_ECC[7]
C35	VSS
C36	DDRA_ECC[0]
C37	DDRA_ECC[4]
C38	DDRA_DQ[30]
C39	DDRA_DQS_DN[12]
C40	VSS
C41	DDRA_DQ[25]
C42	PREQ#
C43	VSS
D1	BPM[4]#
D2	BPM[6]#
D3	VSS
D4	DDRA_DQS_DN[13]
D5	DDRA_DQS_DP[13]
D6	DDRB_DQ[38]
D7	DDRB_DQS_DN[4]
D8	VSS
D9	DDRC_CS[5]#

XY Coord	Signal
D10	DDRC_ODT[3]
D11	DDRB_ODT[0]
D12	DDRB_CS[0]#
D13	VDDQ
D14	DDRB_ODT[2]
D15	DDRC_ODT[2]
D16	DDRC_CS[2]#
D17	DDRC_RAS#
D18	VDDQ
D19	DDRA_CLK_DP[1]
D20	DDRB_MA_PAR
D21	DDRB_CLK_DN[0]
D22	DDRB_MA[7]
D23	VDDQ
D24	DDRA_MA[3]
D25	DDRA_PAR_ERR[0]#
D26	DDRC_CKE[2]
D27	DDRB_CKE[2]
D28	VDDQ
D29	DDRB_RESET#
D30	VSS
D31	VSS
D32	DDRA_RESET#
D33	VSS
D34	DDRA_DQS_DP[8]
D35	DDRA_DQS_DN[8]
D36	DDRB_ECC[0]
D37	DDRA_DQ[27]
D38	VSS
D39	DDRA_DQS_DP[12]
D40	DDRA_DQ[24]
D41	DDRA_DQ[28]
D42	DDRA_DQ[29]
D43	VSS
E1	VSS
E2	BPM[7]#
E3	DDRA_DQS_DP[4]
E4	DDRA_DQS_DN[4]
E5	DDRB_DQ[34]
E6	VSS
E7	DDRB_DQS_DP[4]

XY Coord	Signal
E8	DDRB_DQ[33]
E9	DDRB_DQ[32]
E10	DDRB_CS[5]#
E11	VDDQ
E12	DDRB_CS[7]#
E13	DDRB_CS[3]#
E14	DDRB_CAS#
E15	DDRB_CS[2]#
E16	VDDQ
E17	DDRC_CS[4]#
E18	DDRA_CLK_DN[2]
E19	DDRA_CLK_DN[3]
E20	DDRA_CLK_DP[3]
E21	VDDQ
E22	DDRB_MA[8]
E23	DDRB_MA[11]
E24	DDRB_MA[12]
E25	DDRB_PAR_ERR[1]#
E26	VDDQ
E27	DDRB_CKE[1]
E28	VSS
E29	DDRC_ECC[2]
E30	DDRC_ECC[3]
E31	VDDQ
E32	DDRC_RESET#
E33	DDRB_ECC[2]
E34	DDRB_ECC[6]
E35	DDRB_DQS_DN[17]
E36	VSS
E37	DDRB_ECC[4]
E38	DDRC_DQ[31]
E39	DDRC_DQS_DP[3]
E40	DDRC_DQS_DN[3]
E41	VSS
E42	DDRA_DQ[18]
E43	DDRA_DQ[19]
F1	DDRA_DQ[34]
F2	DDRA_DQ[39]
F3	DDRA_DQ[38]
F4	VSS
F5	DDRB_DQ[35]

XY Coord	Signal
F6	DDRB_DQ[39]
F7	DDRB_DQS_DN[13]
F8	DDRB_DQS_DP[13]
F9	VSS
F10	DDRB_DQ[36]
F11	DDRB_ODT[3]
F12	DDRA_ODT[0]
F13	DDRC_ODT[1]
F14	VDDQ
F15	DDRC_MA[13]
F16	DDRC_CAS#
F17	DDRC_BA[1]
F18	DDRA_CLK_DP[2]
F19	VDDQ
F20	DDRC_MA[4]
F21	DDRC_PAR_ERR[0]#
F22	DDRB_MA[5]
F23	DDRC_PAR_ERR[2]#
F24	VDDQ
F25	DDRB_PAR_ERR[2]#
F26	DDRB_MA[15]
F27	NC_F27
F28	VSS
F29	VSS
F30	DDRC_ECC[7]
F31	DDRC_ECC[6]
F32	DDRA_ECC[2]
F33	DDRC_ECC[1]
F34	VSS
F35	DDRB_DQS_DP[17]
F36	DDRB_ECC[1]
F37	DDRB_ECC[5]
F38	DDRC_DQ[30]
F39	VSS
F40	DDRC_DQ[25]
F41	DDRA_DQS_DP[2]
F42	DDRA_DQ[23]
F43	DDRA_DQ[22]
G1	DDRA_DQ[44]
G2	VSS
G3	DDRA_DQ[35]

XY Coord	Signal
G4	DDRB_DQ[42]
G5	DDRB_DQ[46]
G6	DDRB_DQS_DN[5]
G7	VSS
G8	DDRB_DQ[37]
G9	DDRB_DQ[44]
G10	DDRC_DQ[37]
G11	DDRC_DQ[36]
G12	VSS
G13	DDRB_WE#
G14	DDRB_RAS#
G15	DDRA_CS[0]#
G16	DDRC_CS[0]#
G17	VDDQ
G18	DDRC_MA[2]
G19	DDRB_CLK_DP[1]
G20	DDRB_CLK_DN[1]
G21	DDRC_CLK_DN[2]
G22	VDDQ
G23	DDRC_MA[12]
G24	DDRB_MA[9]
G25	DDRC_MA[15]
G26	DDRC_CKE[1]
G27	VDDQ
G28	VSS
G29	DDRC_DQS_DP[8]
G30	DDRC_DQS_DN[8]
G31	DDRC_DQS_DN[17]
G32	VSS
G33	DDRB_DQS_DP[8]
G34	DDRB_DQS_DN[8]
G35	DDRB_ECC[7]
G36	DDRB_ECC[3]
G37	VSS
G38	DDRC_DQS_DN[12]
G39	DDRC_DQ[29]
G40	DDRC_DQ[24]
G41	DDRA_DQS_DN[2]
G42	VSS
G43	DDRA_DQS_DN[11]
H1	DDRA_DQ[41]

XY Coord	Signal
H2	DDRA_DQ[40]
H3	DDRA_DQ[45]
H4	DDRB_DQ[43]
H5	VSS
H6	DDRB_DQS_DP[5]
H7	DDRB_DQS_DP[14]
H8	DDRB_DQ[40]
H9	DDRB_DQ[45]
H10	VSS
H11	DDRC_DQS_DP[13]
H12	DDRC_DQ[38]
H13	DDRC_DQ[34]
H14	DDRB_MA[10]
H15	VDDQ
H16	DDRC_CS[3]#
H17	DDRC_MA[10]
H18	DDRB_CLK_DP[3]
H19	DDRB_CLK_DN[3]
H20	VDDQ
H21	DDRC_CLK_DP[2]
H22	DDRC_MA[9]
H23	DDRC_MA[11]
H24	DDRC_MA[14]
H25	VDDQ
H26	DDRB_MA[14]
H27	DDRB_BA[2]
H28	DDRB_CKE[0]
H29	VSS
H30	VSS
H31	DDRC_DQS_DP[17]
H32	DDRC_ECC[0]
H33	DDRB_DQ[24]
H34	DDRB_DQ[29]
H35	VSS
H36	DDRB_DQ[23]
H37	DDRC_DQ[27]
H38	DDRC_DQS_DP[12]
H39	DDRC_DQ[28]
H40	VSS
H41	DDRA_DQ[16]
H42	DDRA_DQS_DP[11]

XY Coord	Signal
H43	DDRA_DQ[17]
J1	DDRA_DQS_DN[14]
J2	DDRA_DQS_DP[14]
J3	VSS
J4	DDRB_DQ[52]
J5	DDRB_DQ[47]
J6	DDRB_DQ[41]
J7	DDRB_DQS_DN[14]
J8	VSS
J9	DDRC_DQS_DN[4]
J10	DDRC_DQS_DP[4]
J11	DDRC_DQS_DN[13]
J12	DDRC_DQ[33]
J13	VSS
J14	DDRB_MA[0]
J15	DDRC_CS[7]#
J16	DDRB_MA[1]
J17	DDRB_MA[2]
J18	VDDQ
J19	DDRA_CLK_DP[0]
J20	DDRC_MA[3]
J21	DDRC_CLK_DN[0]
J22	DDRC_CLK_DP[0]
J23	VDDQ
J24	DDRC_MA[7]
J25	DDRC_PAR_ERR[1]#
J26	DDRC_CKE[0]
J27	DDRB_MA[6]
J28	VDDQ
J29	VSS
J30	DDRC_ECC[5]
J31	DDRC_ECC[4]
J32	DDRB_DQ[27]
J33	VSS
J34	DDRB_DQ[28]
J35	DDRB_DQ[19]
J36	DDRB_DQ[22]
J37	DDRC_DQ[26]
J38	VSS
J39	DDRC_DQ[19]
J40	DDRC_DQ[18]

XY Coord	Signal
J41	DDRA_DQ[21]
J42	DDRA_DQ[20]
J43	VSS
K1	VSS
K2	DDRA_DQS_DP[5]
K3	DDRA_DQS_DN[5]
K4	DDRB_DQ[48]
K5	DDRB_DQ[49]
K6	VSS
K7	DDRC_DQS_DN[5]
K8	DDRC_DQS_DN[14]
K9	DDRC_DQS_DP[14]
K10	DDRC_DQ[41]
K11	VSS
K12	DDRC_DQ[32]
K13	DDRB_BA[1]
K14	DDRC_CS[1]#
K15	RSVD_K15
K16	VDDQ
K17	DDRC_MA[1]
K18	DDRB_CLK_DP[2]
K19	DDRA_CLK_DN[0]
K20	DDRC_CLK_DN[1]
K21	VDDQ
K22	DDRC_MA[6]
K23	DDRC_MA[5]
K24	RSVD_K24
K25	NC_K25
K26	VDDQ
K27	VSS
K28	DDRB_MA[4]
K29	VSS
K30	DDRB_DQ[31]
K31	VSS
K32	DDRB_DQ[26]
K33	DDRB_DQS_DN[12]
K34	DDRB_DQS_DP[12]
K35	DDRB_DQ[18]
K36	VSS
K37	DDRB_DQS_DN[11]
K38	DDRC_DQ[23]

XY Coord	Signal
K39	DDRC_DQS_DN[2]
K40	DDRC_DQS_DP[2]
K41	VSS
K42	DDRA_DQ[10]
K43	DDRA_DQ[11]
L1	DDRA_DQ[42]
L2	DDRA_DQ[47]
L3	DDRA_DQ[46]
L4	VSS
L5	DDRB_DQS_DN[6]
L6	DDRB_DQS_DP[6]
L7	DDRC_DQS_DP[5]
L8	DDRC_DQ[46]
L9	VSS
L10	DDRC_DQ[40]
L11	DDRC_DQ[44]
L12	DDRC_DQ[39]
L13	DDRC_DQ[35]
L14	VDDQ
L15	RSVD_L15
L16	DDRC_ODT[0]
L17	DDRC_CS[6]#
L18	DDRB_CLK_DN[2]
L19	VDDQ
L20	DDRC_CLK_DP[1]
L21	DDRC_CLK_DN[3]
L22	DDRC_CLK_DP[3]
L23	RSVD_L23
L24	VDDQ
L25	DDRC_MA[8]
L26	DDRC_BA[2]
L27	DDRC_CKE[3]
L28	DDRB_MA[3]
L29	VSS
L30	DDRB_DQS_DP[3]
L31	DDRB_DQS_DN[3]
L32	DDRB_DQ[30]
L33	DDRB_DQ[25]
L34	VSS
L35	DDRB_DQS_DP[2]
L36	DDRB_DQS_DN[2]

XY Coord	Signal
L37	DDRB_DQS_DP[11]
L38	DDRC_DQS_DN[11]
L39	VSS
L40	DDRC_DQ[22]
L41	DDRA_DQS_DP[1]
L42	DDRA_DQ[15]
L43	DDRA_DQ[14]
M1	DDRA_DQ[43]
M2	VSS
M3	DDRA_DQ[52]
M4	DDRB_DQS_DN[15]
M5	DDRB_DQS_DP[15]
M6	DDRB_DQ[53]
M7	VSS
M8	DDRC_DQ[47]
M9	DDRC_DQ[42]
M10	DDRC_DQ[45]
M11	VCC
M12	VSS
M13	VCC
M14	VSS
M15	VCC
M16	VSS
M17	VDDQ
M18	VSS
M19	VCC
M20	VSS
M21	VCC
M22	VSS
M23	VCC
M24	VSS
M25	VCC
M26	VSS
M27	VDDQ
M28	VSS
M29	VCC
M30	VSS
M31	VCC
M32	VSS
M33	VCC
M34	DDRB_DQ[17]

XY Coord	Signal
M35	DDRB_DQ[16]
M36	DDRB_DQ[21]
M37	VSS
M38	DDRC_DQS_DP[11]
M39	DDRC_DQ[16]
M40	DDRC_DQ[17]
M41	DDRA_DQS_DN[1]
M42	VSS
M43	DDRA_DQS_DN[10]
N1	DDRA_DQ[48]
N2	DDRA_DQ[49]
N3	DDRA_DQ[53]
N4	DDRC_DQS_DP[15]
N5	VSS
N6	DDRC_DQ[49]
N7	DDRC_DQ[53]
N8	DDRC_DQ[52]
N9	DDRC_DQ[43]
N10	VSS
N11	VCC
N33	VCC
N34	DDRB_DQ[20]
N35	VSS
N36	DDRC_DQ[21]
N37	DDRB_DQ[14]
N38	DDRB_DQ[15]
N39	DDRB_DQ[11]
N40	VSS
N41	DDRA_DQ[8]
N42	DDRA_DQS_DP[10]
N43	DDRA_DQ[9]
P1	DDRA_DQS_DN[15]
P2	DDRA_DQS_DP[15]
P3	VSS
P4	DDRC_DQS_DN[15]
P5	DDRC_DQS_DN[6]
P6	DDRC_DQS_DP[6]
P7	DDRC_DQ[48]
P8	VSS
P9	DDRC_DQ[50]
P10	DDRC_DQ[51]

XY Coord	Signal
P11	VSS
P33	VSS
P34	DDRB_DQ[8]
P35	DDRB_DQ[9]
P36	DDRB_DQS_DP[10]
P37	DDRB_DQS_DN[10]
P38	VSS
P39	DDRB_DQ[10]
P40	DDRC_DQ[20]
P41	DDRA_DQ[13]
P42	DDRA_DQ[12]
P43	VSS
R1	VSS
R2	DDRA_DQS_DP[6]
R3	DDRA_DQS_DN[6]
R4	DDRA_DQ[54]
R5	DDRB_DQ[50]
R6	VSS
R7	DDRB_DQ[55]
R8	DDRB_DQ[54]
R9	DDRC_DQ[55]
R10	DDRC_DQ[54]
R11	VCC
R33	VCC
R34	DDRB_DQ[12]
R35	DDRB_DQ[13]
R36	VSS
R37	DDRB_DQS_DN[1]
R38	DDRB_DQS_DP[1]
R39	DDRC_DQ[10]
R40	DDRC_DQ[15]
R41	VSS
R42	DDRA_DQ[3]
R43	DDRA_DQ[2]
T1	DDRA_DQ[50]
T2	DDRA_DQ[51]
T3	DDRA_DQ[55]
T4	VSS
T5	DDRB_DQ[51]
T6	DDRC_DQ[60]
T7	DDRC_DQ[61]
T8	DDRC_DQS_DN[7]
T9	VSS
T10	DDRC_DQ[58]
T11	VCC
T33	VCC
T34	VSS
T35	DDRC_DQS_DN[9]
T36	DDRC_DQ[11]
T37	DDRC_DQS_DP[1]
T38	DDRC_DQS_DN[1]
T39	VSS
T40	DDRC_DQS_DN[10]
T41	DDRC_DQ[14]
T42	DDRA_DQ[7]
T43	DDRA_DQS_DP[0]
U1	DDRA_DQ[60]
U2	VSS
U3	DDRA_DQ[61]
U4	DDRA_DQ[56]
U5	DDRC_DQ[56]
U6	DDRC_DQ[57]
U7	VSS
U8	DDRC_DQS_DP[7]
U9	DDRC_DQ[63]
U10	DDRC_DQ[59]
U11	NC_U11
U33	VCCPLL
U34	DDRC_DQ[4]
U35	DDRC_DQS_DP[9]
U36	DDRC_DQ[3]
U37	VSS
U38	DDRC_DQ[8]
U39	DDRC_DQ[9]
U40	DDRC_DQS_DP[10]
U41	DDRA_DQ[6]
U42	VSS
U43	DDRA_DQS_DN[0]
V1	DDRA_DQ[57]
V2	DDRA_DQS_DP[16]
V3	DDRA_DQS_DN[16]
V4	DDRA_DQ[62]
V5	VSS
V6	DDRC_DQS_DP[16]
V7	DDRC_DQS_DN[16]
V8	DDRC_DQ[62]
V9	DDRB_DQ[60]
V10	VSS
V11	NC_V11
V33	VCCPLL
V34	DDRC_DQ[5]
V35	VSS
V36	DDRC_DQ[2]
V37	DDRC_DQ[6]
V38	DDRC_DQ[7]
V39	DDRC_DQ[13]
V40	VSS
V41	DDRA_DQ[1]
V42	DDRA_DQS_DN[9]
V43	DDRA_DQS_DP[9]
W1	DDRA_DQS_DN[7]
W2	DDRA_DQS_DP[7]
W3	VSS
W4	DDRA_DQ[63]
W5	DDRB_DQ[61]
W6	DDRB_DQ[56]
W7	DDRB_DQ[57]
W8	VSS
W9	DDRB_DQ[63]
W10	DDRB_DQ[59]
W11	VCC
W33	VCCPLL
W34	DDRC_DQ[0]
W35	DDRC_DQ[1]
W36	DDRC_DQS_DN[0]
W37	DDRC_DQS_DP[0]
W38	VSS
W39	DDRC_DQ[12]
W40	DDRA_DQ[4]
W41	DDRA_DQ[0]
W42	DDRA_DQ[5]
W43	VSS
Y1	VSS

XY Coord	Signal
Y2	DDRA_DQ[58]
Y3	DDRA_DQ[59]
Y4	DDRB_DQS_DP[16]
Y5	DDRB_DQS_DN[16]
Y6	VSS
Y7	DDR_COMP[1]
Y8	DDRB_DQS_DP[7]
Y9	DDRB_DQS_DN[7]
Y10	DDRB_DQ[58]
Y11	VSS
Y33	VSS
Y34	DDRB_DQ[3]
Y35	DDRB_DQ[2]
Y36	VSS
Y37	DDRB_DQS_DN[0]
Y38	DDRB_DQS_DP[0]
Y39	DDRB_DQ[7]
Y40	DDRB_DQ[6]
Y41	VSS
AA3	VSS
AA4	BCLK_ITP_DN
AA5	BCLK_ITP_DP
AA6	DDR_DRAMPWROK
AA7	DDRB_DQ[62]
AA8	DDR_COMP[0]
AA9	VSS
AA10	VTTD
AA11	VTTD
AA33	VTTD
AA34	VSS
AA35	DDRB_DQ[1]
AA36	DDRB_DQ[0]
AA37	DDRB_DQS_DP[9]
AA38	DDRB_DQS_DN[9]
AA39	VSS
AA40	VSS
AA41	DMI_PE_RX_DN[0]
AB3	QPI_TX_DN[13]
AB4	VSS
AB5	DDR_THERM#
AB6	QPI_TX_DP[16]

XY Coord	Signal
AB7	VSS
AB8	VTTD
AB9	VTTD
AB10	VTTD
AB11	VTTD
AB33	VTTD
AB34	VTTD
AB35	DDRB_DQ[4]
AB36	DDRB_DQ[5]
AB37	VSS
AB38	DMI_PE_RX_DN[2]
AB39	DMI_PE_RX_DN[1]
AB40	DMI_PE_RX_DP[1]
AB41	DMI_PE_RX_DP[0]
AB42	VSS
AB43	DMI_PE_TX_DN[3]
AC1	DDR_COMP[2]
AC2	VSS
AC3	QPI_TX_DP[13]
AC4	QPI_TX_DP[15]
AC5	VSS
AC6	QPI_TX_DN[16]
AC7	VSS
AC8	QPI_TX_DP[19]
AC9	VSS
AC10	VTTD
AC11	VTTD
AC33	VTTD
AC34	VTTD
AC35	VTTD
AC36	VSS
AC37	DMI_PE_RX_DP[3]
AC38	DMI_PE_RX_DN[3]
AC39	DMI_PE_RX_DP[2]
AC40	VSS
AC41	COMP0
AC42	DMI_PE_TX_DN[2]
AC43	DMI_PE_TX_DP[3]
AD1	QPI_TX_DN[11]
AD2	QPI_TX_DP[12]
AD3	QPI_TX_DN[12]

XY Coord	Signal
AD4	QPI_TX_DN[15]
AD5	QPI_TX_DP[18]
AD6	QPI_TX_DP[17]
AD7	QPI_TX_DN[17]
AD8	QPI_TX_DN[19]
AD9	VTTD
AD10	VTTA
AD11	VSS
AD33	VSS
AD34	VTTD
AD35	VTTD
AD36	VTTD
AD37	VSS
AD38	CATERR#
AD39	VSS
AD40	DMI_PE_TX_DN[1]
AD41	DMI_PE_TX_DP[1]
AD42	DMI_PE_TX_DP[2]
AD43	VSS
AE1	QPI_TX_DP[11]
AE2	VSS
AE3	QPI_TX_DP[14]
AE4	QPI_TX_DN[14]
AE5	QPI_TX_DN[18]
AE6	QPI_CLKTX_DN
AE7	VSS
AE8	VTTD
AE9	VTTD
AE10	VTTA
AE11	VTTA
AE33	VTTA
AE34	VTTD
AE35	VTTD
AE36	VTTD_SENSE
AE37	VSS_SENSE_VTT
AE38	PE_RX_DN[15]
AE39	PE_RX_DN[14]
AE40	PE_RX_DP[14]
AE41	VSS
AE42	DMI_PE_TX_DN[0]
AE43	DMI_PE_TX_DP[0]

XY Coord	Signal
AF1	BCLK_BUF_DN
AF2	QPI_TX_DP[10]
AF3	QPI_TX_DN[10]
AF4	RSVD_AF4
AF5	VSS
AF6	QPI_CLKTX_DP
AF7	VTT_VID[3]
AF8	VTTD
AF9	VTTD
AF10	NC_AF10
AF11	VTTA
AF33	VTTA
AF34	VTTA
AF35	VSS
AF36	VTTD
AF37	VTTD
AF38	PE_RX_DP[15]
AF39	VSS
AF40	PE_RX_DN[13]
AF41	PE_RX_DN[12]
AF42	PE_RX_DP[12]
AF43	VSS
AG1	BCLK_BUF_DP
AG2	QPI_TX_DN[9]
AG3	VSS
AG4	RSVD_AG4
AG5	RSVD_AG5
AG6	QPI_TX_DN[5]
AG7	QPI_TX_DP[5]
AG8	QPI_TX_DP[0]
AG9	VSS
AG10	TMS
AG11	VSS
AG33	VSS
AG34	VTTA
AG35	VSS
AG36	EKEY_NC
AG37	THERMTRIP#
AG38	VSS
AG39	PE_RX_DN[8]
AG40	PE_RX_DP[13]

XY Coord	Signal
AG41	PE_RX_DN[10]
AG42	PE_RX_DN[11]
AG43	PE_RX_DP[11]
AH1	VSS
AH2	QPI_TX_DP[9]
AH3	QPI_TX_DP[8]
AH4	QPI_TX_DN[8]
AH5	RSVD_AH5
AH6	QPI_TX_DP[2]
AH7	VSS
AH8	QPI_TX_DN[0]
AH9	TRST#
AH10	TCLK
AH11	VCC
AH33	TDI_M
AH34	VSS
AH35	BCLK_DP
AH36	PECI
AH37	VTTPWRGOOD
AH38	VSS
AH39	PE_RX_DP[8]
AH40	VSS
AH41	PE_RX_DP[10]
AH42	VSS
AH43	PE_RX_DN[9]
AJ1	QPI_TX_DN[7]
AJ2	QPI_TX_DN[6]
AJ3	QPI_TX_DP[6]
AJ4	QPI_TX_DP[4]
AJ5	VSS
AJ6	QPI_TX_DN[2]
AJ7	QPI_TX_DN[1]
AJ8	QPI_TX_DP[1]
AJ9	TDI
AJ10	TDO
AJ11	VCC
AJ33	VCC
AJ34	VCC
AJ35	BCLK_DN
AJ36	RSTIN#
AJ37	NC_AJ37

XY Coord	Signal
AJ38	PE_RX_DP[0]
AJ39	VSS
AJ40	PE_RX_DP[5]
AJ41	PE_RX_DP[7]
AJ42	PE_RX_DN[7]
AJ43	PE_RX_DP[9]
AK1	QPI_TX_DP[7]
AK2	RSVD_AK2
AK3	VSS
AK4	QPI_TX_DN[4]
AK5	QPI_TX_DN[3]
AK6	QPI_TX_DP[3]
AK7	VSS
AK8	ISENSE
AK9	VSS
AK10	VSS
AK11	VCC
AK12	VCC
AK13	VCC
AK14	VSS
AK15	VCC
AK16	VCC
AK17	VSS
AK18	VCC
AK19	VCC
AK20	VSS
AK21	VCC
AK22	VSS
AK23	VSS
AK24	VCC
AK25	VCC
AK26	VSS
AK27	VCC
AK28	VCC
AK29	VSS
AK30	VCC
AK31	VCC
AK32	VSS
AK33	VCC
AK34	VSS
AK35	PECI_ID#

XY Coord	Signal
AK36	VSS
AK37	PE_RX_DP[3]
AK38	PE_RX_DN[3]
AK39	PE_RX_DN[0]
AK40	PE_RX_DN[5]
AK41	VSS
AK42	PE_RX_DN[6]
AK43	VSS
AL1	VSS
AL2	VSS
AL3	NC_AL3
AL4	RSVD_AL4
AL5	RSVD_AL5
AL6	QPI_COMP[0]
AL7	VSS
AL8	QPI_RX_DN[19]
AL9	VID[1]
AL10	VID[0]
AL11	VSS
AL12	VCC
AL13	VCC
AL14	VSS
AL15	VCC
AL16	VCC
AL17	VSS
AL18	VCC
AL19	VCC
AL20	VSS
AL21	VCC
AL22	VSS
AL23	VSS
AL24	VCC
AL25	VCC
AL26	VSS
AL27	VCC
AL28	VCC
AL29	VSS
AL30	VCC
AL31	VCC
AL32	VSS
AL33	TDO_M

XY Coord	Signal
AL34	SYS_ERR_STAT[2]#
AL35	PROCHOT#
AL36	RSVD_AL36
AL37	VSS
AL38	PE_RX_DP[2]
AL39	VSS
AL40	PE_RX_DP[4]
AL41	PE_RX_DN[4]
AL42	PE_RX_DP[6]
AL43	PE_RCOMPO
AM1	QPI_RX_DN[13]
AM2	QPI_RX_DP[14]
AM3	QPI_RX_DN[14]
AM4	QPI_RX_DP[16]
AM5	VSS
AM6	QPI_RX_DP[18]
AM7	QPI_RX_DN[18]
AM8	QPI_RX_DP[19]
AM9	VSS
AM10	VID[3]
AM11	VSS
AM12	VCC
AM13	VCC
AM14	VSS
AM15	VCC
AM16	VCC
AM17	VSS
AM18	VCC
AM19	VCC
AM20	VSS
AM21	VCC
AM22	VSS
AM23	VSS
AM24	VCC
AM25	VCC
AM26	VSS
AM27	VCC
AM28	VCC
AM29	VSS
AM30	VCC
AM31	VCC

XY Coord	Signal
AM32	VSS
AM33	EXTSYSTRG
AM34	DDR_ADR
AM35	SYS_ERR_STAT[0]#
AM36	SKTOCC#
AM37	PE_RX_DP[1]
AM38	PE_RX_DN[2]
AM39	VSS
AM40	RSVD_AM40
AM41	RSVD_AM41
AM42	VSS
AM43	PE_ICOMPO
AN1	QPI_RX_DP[13]
AN2	QPI_RX_DN[12]
AN3	VSS
AN4	QPI_RX_DN[16]
AN5	QPI_RX_DP[17]
AN6	QPI_RX_DN[17]
AN7	VSS
AN8	VID[7]
AN9	VID[2]
AN10	VID[4]
AN11	VSS
AN12	VCC
AN13	VCC
AN14	VSS
AN15	VCC
AN16	VCC
AN17	VSS
AN18	VCC
AN19	VCC
AN20	VSS
AN21	VCC
AN22	VSS
AN23	VSS
AN24	VCC
AN25	VCC
AN26	VSS
AN27	VCC
AN28	VCC
AN29	VSS

XY Coord	Signal
AN30	VCC
AN31	VCC
AN32	VSS
AN33	RSVD_AN33
AN34	VSS
AN35	VSS
AN36	PM_SYNC
AN37	PE_RX_DN[1]
AN38	RSVD_AN38
AN39	PE_TX_DP[15]
AN40	RSVD_AN40
AN41	VSS
AN42	PE_TX_DN[12]
AN43	PE_ICOMPI
AP1	VSS
AP2	QPI_RX_DP[12]
AP3	QPI_RX_DP[15]
AP4	QPI_RX_DN[15]
AP5	VSS
AP6	VSS
AP7	PSI#
AP8	VID[6]
AP9	VID[5]
AP10	VSS
AP11	VSS
AP12	VCC
AP13	VCC
AP14	VSS
AP15	VCC
AP16	VCC
AP17	VSS
AP18	VCC
AP19	VCC
AP20	VSS
AP21	VCC
AP22	VSS
AP23	VSS
AP24	VCC
AP25	VCC
AP26	VSS
AP27	VCC

XY Coord	Signal
AP28	VCC
AP29	VSS
AP30	VCC
AP31	VCC
AP32	VSS
AP33	PE_HP_CLK
AP34	PE_HP_DATA
AP35	RSVD_AP35
AP36	SYS_ERR_STAT[1]#
AP37	VSS
AP38	PE_TX_DN[14]
AP39	PE_TX_DN[15]
AP40	PE_TX_DP[13]
AP41	PE_TX_DN[13]
AP42	PE_TX_DP[12]
AP43	VSS
AR1	QPI_RX_DN[10]
AR2	VSS
AR3	VSS
AR4	QPI_RX_DP[11]
AR5	QPI_RX_DN[11]
AR6	QPI_CLKRX_DN
AR7	VCCPWRGOOD
AR8	VSS_SENSE
AR9	VCC_SENSE
AR10	VCC
AR11	VSS
AR12	VCC
AR13	VCC
AR14	VSS
AR15	VCC
AR16	VCC
AR17	VSS
AR18	VCC
AR19	VCC
AR20	VSS
AR21	VCC
AR22	VSS
AR23	VSS
AR24	VCC
AR25	VCC

XY Coord	Signal
AR26	VSS
AR27	VCC
AR28	VCC
AR29	VSS
AR30	VCC
AR31	VCC
AR32	VSS
AR33	VSS
AR34	RSVD_AR34
AR35	SMB_DATA
AR36	SMB_CLK
AR37	RSVD_AR37
AR38	PE_TX_DP[14]
AR39	VSS
AR40	PE_CLK_DN
AR41	PE_TX_DP[11]
AR42	PE_TX_DN[11]
AR43	PE_TX_DN[10]
AT1	QPI_RX_DP[10]
AT2	QPI_RX_DN[9]
AT3	QPI_RX_DP[9]
AT4	RSVD_AT4
AT5	RSVD_AT5
AT6	QPI_CLKRX_DP
AT7	VSS
AT8	VSS
AT9	VCC
AT10	VCC
AT11	VSS
AT12	VCC
AT13	VCC
AT14	VSS
AT15	VCC
AT16	VCC
AT17	VSS
AT18	VCC
AT19	VCC
AT20	VSS
AT21	VCC
AT22	VSS
AT23	VSS

XY Coord	Signal
AT24	VCC
AT25	VCC
AT26	VSS
AT27	VCC
AT28	VCC
AT29	VSS
AT30	VCC
AT31	VCC
AT32	VSS
AT33	PE_NTBXL
AT34	VSS
AT35	PE_TX_DP[0]
AT36	DP_SYNCRST#
AT37	VSS
AT38	VSS
AT39	PE_TX_DN[9]
AT40	PE_CLK_DP
AT41	VSS
AT42	RSVD_AT42
AT43	PE_TX_DP[10]
AU1	VSS
AU2	RSVD_AU2
AU3	QPI_RX_DN[8]
AU4	QPI_RX_DP[8]
AU5	VSS
AU6	QPI_RX_DN[6]
AU7	QPI_RX_DP[6]
AU8	QPI_RX_DP[0]
AU9	VCC
AU10	VCC
AU11	VSS
AU12	VCC
AU13	VCC
AU14	VSS
AU15	VCC
AU16	VCC
AU17	VSS
AU18	VCC
AU19	VCC
AU20	VSS
AU21	VCC

XY Coord	Signal
AU22	VSS
AU23	VSS
AU24	VCC
AU25	VCC
AU26	VSS
AU27	VCC
AU28	VCC
AU29	VSS
AU30	VCC
AU31	VCC
AU32	VSS
AU33	NC_AU33
AU34	QPI_COMP[1]
AU35	PE_TX_DN[0]
AU36	VSS
AU37	RSVD_AU37
AU38	PE_TX_DN[8]
AU39	PE_TX_DP[9]
AU40	VSS
AU41	PE_RBIAS
AU42	RSVD_AU42
AU43	VSS
AV1	RSVD_AV1
AV2	RSVD_AV2
AV3	VTT_VID[2]
AV4	VSS
AV5	QPI_RX_DP[3]
AV6	VTT_VID[4]
AV7	QPI_RX_DP[1]
AV8	QPI_RX_DN[0]
AV9	VCC
AV10	VCC
AV11	VSS
AV12	VCC
AV13	VCC
AV14	VSS
AV15	VCC
AV16	VCC
AV17	VSS
AV18	VCC
AV19	VCC

XY Coord	Signal
AV20	VSS
AV21	VCC
AV22	VSS
AV23	VSS
AV24	VCC
AV25	VCC
AV26	VSS
AV27	VCC
AV28	VCC
AV29	VSS
AV30	VCC
AV31	VCC
AV32	VSS
AV33	PE_GEN2_DISABLE#
AV34	PE_CFG[1]
AV35	VSS
AV36	PE_TX_DP[1]
AV37	PE_TX_DN[1]
AV38	PE_TX_DP[8]
AV39	VSS
AV40	PE_TX_DN[7]
AV41	VSS
AV42	RSVD_AV42
AV43	RSVD_AV43
AW1	VSS
AW2	RSVD_AW2
AW3	QPI_RX_DN[7]
AW4	QPI_RX_DP[7]
AW5	QPI_RX_DN[3]
AW6	VSS
AW7	QPI_RX_DN[1]
AW8	VSS
AW9	VCC
AW10	VCC
AW11	VSS
AW12	VCC
AW13	VCC
AW14	VSS
AW15	VCC
AW16	VCC
AW17	VSS

XY Coord	Signal
AW18	VCC
AW19	VCC
AW20	VSS
AW21	VCC
AW22	VSS
AW23	VSS
AW24	VCC
AW25	VCC
AW26	VSS
AW27	VCC
AW28	VCC
AW29	VSS
AW30	VCC
AW31	VCC
AW32	VSS
AW33	DMI_COMP
AW34	NC_AW34
AW35	VSS
AW36	PE_TX_DP[2]
AW37	PE_TX_DP[4]
AW38	PE_TX_DN[4]
AW39	PE_TX_DN[6]
AW40	PE_TX_DP[7]
AW41	RSVD_AW41
AW42	RSVD_AW42
AY2	VSS
AY3	RSVD_AY3
AY4	RSVD_AY4
AY5	QPI_RX_DN[5]
AY6	QPI_RX_DP[5]
AY7	VSS
AY8	QPI_RX_DP[2]
AY9	VCC
AY10	VCC
AY11	VSS
AY12	VCC
AY13	VCC
AY14	VSS
AY15	VCC
AY16	VCC
AY17	VSS

XY Coord	Signal
AY18	VCC
AY19	VCC
AY20	VSS
AY21	VCC
AY22	VSS
AY23	VSS
AY24	VCC
AY25	VCC
AY26	VSS
AY27	VCC
AY28	VCC
AY29	VSS
AY30	VCC
AY31	VCC
AY32	VSS
AY33	PE_CFG[0]
AY34	PE_CFG[2]
AY35	NC_AY35
AY36	PE_TX_DN[2]
AY37	VSS
AY38	PE_TX_DN[5]
AY39	PE_TX_DP[6]
AY40	RSVD_AY40
AY41	RSVD_AY41
AY42	VSS
BA3	VSS
BA4	RSVD_BA4
BA5	VSS
BA6	QPI_RX_DN[4]
BA7	QPI_RX_DP[4]
BA8	QPI_RX_DN[2]
BA9	VCC
BA10	VCC
BA11	VSS
BA12	VCC
BA13	VCC
BA14	VSS
BA15	VCC
BA16	VCC
BA17	VSS
BA18	VCC

XY Coord	Signal
BA19	VCC
BA20	VSS
BA24	VCC
BA25	VCC
BA26	VSS
BA27	VCC
BA28	VCC
BA29	VSS
BA30	VCC
BA35	DMI_PE_CFG#
BA36	PE_TX_DP[3]
BA37	PE_TX_DN[3]
BA38	PE_TX_DP[5]
BA39	VSS
BA40	RSVD_BA40

Table 154. Alphabetical Signal Listing

Signal	XY Coord
BCLK_BUF_DN	AF1
BCLK_BUF_DP	AG1
BCLK_DN	AJ35
BCLK_DP	AH35
BCLK_ITP_DN	AA4
BCLK_ITP_DP	AA5
BPM[0]#	B3
BPM[1]#	A5
BPM[2]#	C2
BPM[3]#	B4
BPM[4]#	D1
BPM[5]#	C3
BPM[6]#	D2
BPM[7]#	E2
CATERR#	AD38
COMP0	AC41
DDR_COMP[0]	AA8
DDR_COMP[1]	Y7
DDR_COMP[2]	AC1
DDR_ADR	AM34
DDR_DRAMPWROK	AA6
DDRA_BA[0]	B16
DDRA_BA[1]	A16
DDRA_BA[2]	C28
DDRA_CAS#	C12
DDRA_CKE[0]	C29
DDRA_CKE[1]	A30
DDRA_CKE[2]	B30
DDRA_CKE[3]	B31
DDRA_CLK_DN[0]	K19
DDRA_CLK_DN[1]	C19
DDRA_CLK_DN[2]	E18
DDRA_CLK_DN[3]	E19
DDRA_CLK_DP[0]	J19
DDRA_CLK_DP[1]	D19
DDRA_CLK_DP[2]	F18
DDRA_CLK_DP[3]	E20
DDRA_CS[0]#	G15
DDRA_CS[1]#	B10

Signal	XY Coord
DDRA_CS[2]#	C13
DDRA_CS[3]#	B9
DDRA_CS[4]#	B15
DDRA_CS[5]#	A7
DDRA_CS[6]#	C11
DDRA_CS[7]#	B8
DDRA_DQ[0]	W41
DDRA_DQ[1]	V41
DDRA_DQ[2]	R43
DDRA_DQ[3]	R42
DDRA_DQ[4]	W40
DDRA_DQ[5]	W42
DDRA_DQ[6]	U41
DDRA_DQ[7]	T42
DDRA_DQ[8]	N41
DDRA_DQ[9]	N43
DDRA_DQ[10]	K42
DDRA_DQ[11]	K43
DDRA_DQ[12]	P42
DDRA_DQ[13]	P41
DDRA_DQ[14]	L43
DDRA_DQ[15]	L42
DDRA_DQ[16]	H41
DDRA_DQ[17]	H43
DDRA_DQ[18]	E42
DDRA_DQ[19]	E43
DDRA_DQ[20]	J42
DDRA_DQ[21]	J41
DDRA_DQ[22]	F43
DDRA_DQ[23]	F42
DDRA_DQ[24]	D40
DDRA_DQ[25]	C41
DDRA_DQ[26]	A38
DDRA_DQ[27]	D37
DDRA_DQ[28]	D41
DDRA_DQ[29]	D42
DDRA_DQ[30]	C38
DDRA_DQ[31]	B38
DDRA_DQ[32]	B5
DDRA_DQ[33]	C4
DDRA_DQ[34]	F1

Signal	XY Coord
DDRA_DQ[35]	G3
DDRA_DQ[36]	B6
DDRA_DQ[37]	C6
DDRA_DQ[38]	F3
DDRA_DQ[39]	F2
DDRA_DQ[40]	H2
DDRA_DQ[41]	H1
DDRA_DQ[42]	L1
DDRA_DQ[43]	M1
DDRA_DQ[44]	G1
DDRA_DQ[45]	H3
DDRA_DQ[46]	L3
DDRA_DQ[47]	L2
DDRA_DQ[48]	N1
DDRA_DQ[49]	N2
DDRA_DQ[50]	T1
DDRA_DQ[51]	T2
DDRA_DQ[52]	M3
DDRA_DQ[53]	N3
DDRA_DQ[54]	R4
DDRA_DQ[55]	T3
DDRA_DQ[56]	U4
DDRA_DQ[57]	V1
DDRA_DQ[58]	Y2
DDRA_DQ[59]	Y3
DDRA_DQ[60]	U1
DDRA_DQ[61]	U3
DDRA_DQ[62]	V4
DDRA_DQ[63]	W4
DDRA_DQS_DN[0]	U43
DDRA_DQS_DN[1]	M41
DDRA_DQS_DN[2]	G41
DDRA_DQS_DN[3]	B40
DDRA_DQS_DN[4]	E4
DDRA_DQS_DN[5]	K3
DDRA_DQS_DN[6]	R3
DDRA_DQS_DN[7]	W1
DDRA_DQS_DN[8]	D35
DDRA_DQS_DN[9]	V42
DDRA_DQS_DN[10]	M43
DDRA_DQS_DN[11]	G43

Signal	XY Coord
DDRA_DQS_DN[12]	C39
DDRA_DQS_DN[13]	D4
DDRA_DQS_DN[14]	J1
DDRA_DQS_DN[15]	P1
DDRA_DQS_DN[16]	V3
DDRA_DQS_DN[17]	B35
DDRA_DQS_DP[0]	T43
DDRA_DQS_DP[1]	L41
DDRA_DQS_DP[2]	F41
DDRA_DQS_DP[3]	B39
DDRA_DQS_DP[4]	E3
DDRA_DQS_DP[5]	K2
DDRA_DQS_DP[6]	R2
DDRA_DQS_DP[7]	W2
DDRA_DQS_DP[8]	D34
DDRA_DQS_DP[9]	V43
DDRA_DQS_DP[10]	N42
DDRA_DQS_DP[11]	H42
DDRA_DQS_DP[12]	D39
DDRA_DQS_DP[13]	D5
DDRA_DQS_DP[14]	J2
DDRA_DQS_DP[15]	P2
DDRA_DQS_DP[16]	V2
DDRA_DQS_DP[17]	B36
DDRA_ECC[0]	C36
DDRA_ECC[1]	A36
DDRA_ECC[2]	F32
DDRA_ECC[3]	C33
DDRA_ECC[4]	C37
DDRA_ECC[5]	A37
DDRA_ECC[6]	B34
DDRA_ECC[7]	C34
DDRA_MA[0]	A20
DDRA_MA[1]	B21
DDRA_MA[2]	C23
DDRA_MA[3]	D24
DDRA_MA[4]	B23
DDRA_MA[5]	B24
DDRA_MA[6]	C24
DDRA_MA[7]	A25
DDRA_MA[8]	B25

Signal	XY Coord
DDRA_MA[9]	C26
DDRA_MA[10]	B19
DDRA_MA[11]	A26
DDRA_MA[12]	B26
DDRA_MA[13]	A10
DDRA_MA[14]	A28
DDRA_MA[15]	B29
DDRA_MA_PAR	B20
DDRA_ODT[0]	F12
DDRA_ODT[1]	C9
DDRA_ODT[2]	B11
DDRA_ODT[3]	C7
DDRA_PAR_ERR[0]#	D25
DDRA_PAR_ERR[1]#	B28
DDRA_PAR_ERR[2]#	A27
DDRA_RAS#	A15
DDRA_RESET#	D32
DDRA_WE#	B13
DDRB_BA[0]	C18
DDRB_BA[1]	K13
DDRB_BA[2]	H27
DDRB_CAS#	E14
DDRB_CKE[0]	H28
DDRB_CKE[1]	E27
DDRB_CKE[2]	D27
DDRB_CKE[3]	C27
DDRB_CLK_DN[0]	D21
DDRB_CLK_DN[1]	G20
DDRB_CLK_DN[2]	L18
DDRB_CLK_DN[3]	H19
DDRB_CLK_DP[0]	C21
DDRB_CLK_DP[1]	G19
DDRB_CLK_DP[2]	K18
DDRB_CLK_DP[3]	H18
DDRB_CS[0]#	D12
DDRB_CS[1]#	A8
DDRB_CS[2]#	E15
DDRB_CS[3]#	E13
DDRB_CS[4]#	C17
DDRB_CS[5]#	E10
DDRB_CS[6]#	C14

Signal	XY Coord
DDRB_CS[7]#	E12
DDRB_DQ[0]	AA36
DDRB_DQ[1]	AA35
DDRB_DQ[2]	Y35
DDRB_DQ[3]	Y34
DDRB_DQ[4]	AB35
DDRB_DQ[5]	AB36
DDRB_DQ[6]	Y40
DDRB_DQ[7]	Y39
DDRB_DQ[8]	P34
DDRB_DQ[9]	P35
DDRB_DQ[10]	P39
DDRB_DQ[11]	N39
DDRB_DQ[12]	R34
DDRB_DQ[13]	R35
DDRB_DQ[14]	N37
DDRB_DQ[15]	N38
DDRB_DQ[16]	M35
DDRB_DQ[17]	M34
DDRB_DQ[18]	K35
DDRB_DQ[19]	J35
DDRB_DQ[20]	N34
DDRB_DQ[21]	M36
DDRB_DQ[22]	J36
DDRB_DQ[23]	H36
DDRB_DQ[24]	H33
DDRB_DQ[25]	L33
DDRB_DQ[26]	K32
DDRB_DQ[27]	J32
DDRB_DQ[28]	J34
DDRB_DQ[29]	H34
DDRB_DQ[30]	L32
DDRB_DQ[31]	K30
DDRB_DQ[32]	E9
DDRB_DQ[33]	E8
DDRB_DQ[34]	E5
DDRB_DQ[35]	F5
DDRB_DQ[36]	F10
DDRB_DQ[37]	G8
DDRB_DQ[38]	D6
DDRB_DQ[39]	F6

Signal	XY Coord
DDRB_DQ[40]	H8
DDRB_DQ[41]	J6
DDRB_DQ[42]	G4
DDRB_DQ[43]	H4
DDRB_DQ[44]	G9
DDRB_DQ[45]	H9
DDRB_DQ[46]	G5
DDRB_DQ[47]	J5
DDRB_DQ[48]	K4
DDRB_DQ[49]	K5
DDRB_DQ[50]	R5
DDRB_DQ[51]	T5
DDRB_DQ[52]	J4
DDRB_DQ[53]	M6
DDRB_DQ[54]	R8
DDRB_DQ[55]	R7
DDRB_DQ[56]	W6
DDRB_DQ[57]	W7
DDRB_DQ[58]	Y10
DDRB_DQ[59]	W10
DDRB_DQ[60]	V9
DDRB_DQ[61]	W5
DDRB_DQ[62]	AA7
DDRB_DQ[63]	W9
DDRB_DQS_DN[0]	Y37
DDRB_DQS_DN[1]	R37
DDRB_DQS_DN[2]	L36
DDRB_DQS_DN[3]	L31
DDRB_DQS_DN[4]	D7
DDRB_DQS_DN[5]	G6
DDRB_DQS_DN[6]	L5
DDRB_DQS_DN[7]	Y9
DDRB_DQS_DN[8]	G34
DDRB_DQS_DN[9]	AA38
DDRB_DQS_DN[10]	P37
DDRB_DQS_DN[11]	K37
DDRB_DQS_DN[12]	K33
DDRB_DQS_DN[13]	F7
DDRB_DQS_DN[14]	J7
DDRB_DQS_DN[15]	M4
DDRB_DQS_DN[16]	Y5

Signal	XY Coord
DDRB_DQS_DN[17]	E35
DDRB_DQS_DP[0]	Y38
DDRB_DQS_DP[1]	R38
DDRB_DQS_DP[2]	L35
DDRB_DQS_DP[3]	L30
DDRB_DQS_DP[4]	E7
DDRB_DQS_DP[5]	H6
DDRB_DQS_DP[6]	L6
DDRB_DQS_DP[7]	Y8
DDRB_DQS_DP[8]	G33
DDRB_DQS_DP[9]	AA37
DDRB_DQS_DP[10]	P36
DDRB_DQS_DP[11]	L37
DDRB_DQS_DP[12]	K34
DDRB_DQS_DP[13]	F8
DDRB_DQS_DP[14]	H7
DDRB_DQS_DP[15]	M5
DDRB_DQS_DP[16]	Y4
DDRB_DQS_DP[17]	F35
DDRB_ECC[0]	D36
DDRB_ECC[1]	F36
DDRB_ECC[2]	E33
DDRB_ECC[3]	G36
DDRB_ECC[4]	E37
DDRB_ECC[5]	F37
DDRB_ECC[6]	E34
DDRB_ECC[7]	G35
DDRB_MA[0]	J14
DDRB_MA[1]	J16
DDRB_MA[2]	J17
DDRB_MA[3]	L28
DDRB_MA[4]	K28
DDRB_MA[5]	F22
DDRB_MA[6]	J27
DDRB_MA[7]	D22
DDRB_MA[8]	E22
DDRB_MA[9]	G24
DDRB_MA[10]	H14
DDRB_MA[11]	E23
DDRB_MA[12]	E24
DDRB_MA[13]	B14

Signal	XY Coord
DDRB_MA[14]	H26
DDRB_MA[15]	F26
DDRB_MA_PAR	D20
DDRB_ODT[0]	D11
DDRB_ODT[1]	C8
DDRB_ODT[2]	D14
DDRB_ODT[3]	F11
DDRB_PAR_ERR[0]#	C22
DDRB_PAR_ERR[1]#	E25
DDRB_PAR_ERR[2]#	F25
DDRB_RAS#	G14
DDRB_RESET#	D29
DDRB_WE#	G13
DDRC_BA[0]	A17
DDRC_BA[1]	F17
DDRC_BA[2]	L26
DDRC_CAS#	F16
DDRC_CKE[0]	J26
DDRC_CKE[1]	G26
DDRC_CKE[2]	D26
DDRC_CKE[3]	L27
DDRC_CLK_DN[0]	J21
DDRC_CLK_DN[1]	K20
DDRC_CLK_DN[2]	G21
DDRC_CLK_DN[3]	L21
DDRC_CLK_DP[0]	J22
DDRC_CLK_DP[1]	L20
DDRC_CLK_DP[2]	H21
DDRC_CLK_DP[3]	L22
DDRC_CS[0]#	G16
DDRC_CS[1]#	K14
DDRC_CS[2]#	D16
DDRC_CS[3]#	H16
DDRC_CS[4]#	E17
DDRC_CS[5]#	D9
DDRC_CS[6]#	L17
DDRC_CS[7]#	J15
DDRC_DQ[0]	W34
DDRC_DQ[1]	W35
DDRC_DQ[2]	V36
DDRC_DQ[3]	U36

Signal	XY Coord
DDRC_DQ[4]	U34
DDRC_DQ[5]	V34
DDRC_DQ[6]	V37
DDRC_DQ[7]	V38
DDRC_DQ[8]	U38
DDRC_DQ[9]	U39
DDRC_DQ[10]	R39
DDRC_DQ[11]	T36
DDRC_DQ[12]	W39
DDRC_DQ[13]	V39
DDRC_DQ[14]	T41
DDRC_DQ[15]	R40
DDRC_DQ[16]	M39
DDRC_DQ[17]	M40
DDRC_DQ[18]	J40
DDRC_DQ[19]	J39
DDRC_DQ[20]	P40
DDRC_DQ[21]	N36
DDRC_DQ[22]	L40
DDRC_DQ[23]	K38
DDRC_DQ[24]	G40
DDRC_DQ[25]	F40
DDRC_DQ[26]	J37
DDRC_DQ[27]	H37
DDRC_DQ[28]	H39
DDRC_DQ[29]	G39
DDRC_DQ[30]	F38
DDRC_DQ[31]	E38
DDRC_DQ[32]	K12
DDRC_DQ[33]	J12
DDRC_DQ[34]	H13
DDRC_DQ[35]	L13
DDRC_DQ[36]	G11
DDRC_DQ[37]	G10
DDRC_DQ[38]	H12
DDRC_DQ[39]	L12
DDRC_DQ[40]	L10
DDRC_DQ[41]	K10
DDRC_DQ[42]	M9
DDRC_DQ[43]	N9
DDRC_DQ[44]	L11

Signal	XY Coord
DDRC_DQ[45]	M10
DDRC_DQ[46]	L8
DDRC_DQ[47]	M8
DDRC_DQ[48]	P7
DDRC_DQ[49]	N6
DDRC_DQ[50]	P9
DDRC_DQ[51]	P10
DDRC_DQ[52]	N8
DDRC_DQ[53]	N7
DDRC_DQ[54]	R10
DDRC_DQ[55]	R9
DDRC_DQ[56]	U5
DDRC_DQ[57]	U6
DDRC_DQ[58]	T10
DDRC_DQ[59]	U10
DDRC_DQ[60]	T6
DDRC_DQ[61]	T7
DDRC_DQ[62]	V8
DDRC_DQ[63]	U9
DDRC_DQS_DN[0]	W36
DDRC_DQS_DN[1]	T38
DDRC_DQS_DN[2]	K39
DDRC_DQS_DN[3]	E40
DDRC_DQS_DN[4]	J9
DDRC_DQS_DN[5]	K7
DDRC_DQS_DN[6]	P5
DDRC_DQS_DN[7]	T8
DDRC_DQS_DN[8]	G30
DDRC_DQS_DN[9]	T35
DDRC_DQS_DN[10]	T40
DDRC_DQS_DN[11]	L38
DDRC_DQS_DN[12]	G38
DDRC_DQS_DN[13]	J11
DDRC_DQS_DN[14]	K8
DDRC_DQS_DN[15]	P4
DDRC_DQS_DN[16]	V7
DDRC_DQS_DN[17]	G31
DDRC_DQS_DP[0]	W37
DDRC_DQS_DP[1]	T37
DDRC_DQS_DP[2]	K40
DDRC_DQS_DP[3]	E39

Signal	XY Coord
DDRC_DQS_DP[4]	J10
DDRC_DQS_DP[5]	L7
DDRC_DQS_DP[6]	P6
DDRC_DQS_DP[7]	U8
DDRC_DQS_DP[8]	G29
DDRC_DQS_DP[9]	U35
DDRC_DQS_DP[10]	U40
DDRC_DQS_DP[11]	M38
DDRC_DQS_DP[12]	H38
DDRC_DQS_DP[13]	H11
DDRC_DQS_DP[14]	K9
DDRC_DQS_DP[15]	N4
DDRC_DQS_DP[16]	V6
DDRC_DQS_DP[17]	H31
DDRC_ECC[0]	H32
DDRC_ECC[1]	F33
DDRC_ECC[2]	E29
DDRC_ECC[3]	E30
DDRC_ECC[4]	J31
DDRC_ECC[5]	J30
DDRC_ECC[6]	F31
DDRC_ECC[7]	F30
DDRC_MA[0]	A18
DDRC_MA[1]	K17
DDRC_MA[2]	G18
DDRC_MA[3]	J20
DDRC_MA[4]	F20
DDRC_MA[5]	K23
DDRC_MA[6]	K22
DDRC_MA[7]	J24
DDRC_MA[8]	L25
DDRC_MA[9]	H22
DDRC_MA[10]	H17
DDRC_MA[11]	H23
DDRC_MA[12]	G23
DDRC_MA[13]	F15
DDRC_MA[14]	H24
DDRC_MA[15]	G25
DDRC_MA_PAR	B18
DDRC_ODT[0]	L16
DDRC_ODT[1]	F13

Signal	XY Coord
DDRC_ODT[2]	D15
DDRC_ODT[3]	D10
DDRC_PAR_ERR[0]#	F21
DDRC_PAR_ERR[1]#	J25
DDRC_PAR_ERR[2]#	F23
DDRC_RAS#	D17
DDRC_RESET#	E32
DDRC_WE#	C16
DMI_COMP	AW33
DMI_PE_CFG#	BA35
DMI_PE_RX_DN[0]	AA41
DMI_PE_RX_DN[1]	AB39
DMI_PE_RX_DN[2]	AB38
DMI_PE_RX_DN[3]	AC38
DMI_PE_RX_DP[0]	AB41
DMI_PE_RX_DP[1]	AB40
DMI_PE_RX_DP[2]	AC39
DMI_PE_RX_DP[3]	AC37
DMI_PE_TX_DN[0]	AE42
DMI_PE_TX_DN[1]	AD40
DMI_PE_TX_DN[2]	AC42
DMI_PE_TX_DN[3]	AB43
DMI_PE_TX_DP[0]	AE43
DMI_PE_TX_DP[1]	AD41
DMI_PE_TX_DP[2]	AD42
DMI_PE_TX_DP[3]	AC43
DP_SYNCRST#	AT36
EKEY_NC	AG36
EXTSYSTRG	AM33
ISENSE	AK8
NC_AF10	AF10
NC_AJ37	AJ37
NC_AL3	AL3
NC_AU33	AU33
NC_AW34	AW34
NC_AY35	AY35
NC_B33	B33
NC_F27	F27
NC_K25	K25
NC_U11	U11
NC_V11	V11

Signal	XY Coord
PE_CFG[0]	AY33
PE_CFG[1]	AV34
PE_CFG[2]	AY34
PE_CLK_DN	AR40
PE_CLK_DP	AT40
PE_GEN2_DISABLE#	AV33
PE_HP_CLK	AP33
PE_HP_DATA	AP34
PE_ICOMPI	AN43
PE_ICOMPO	AM43
PE_NTBXL	AT33
PE_RBIAS	AU41
PE_RCOMPO	AL43
PE_RX_DN[0]	AK39
PE_RX_DN[1]	AN37
PE_RX_DN[2]	AM38
PE_RX_DN[3]	AK38
PE_RX_DN[4]	AL41
PE_RX_DN[5]	AK40
PE_RX_DN[6]	AK42
PE_RX_DN[7]	AJ42
PE_RX_DN[8]	AG39
PE_RX_DN[9]	AH43
PE_RX_DN[10]	AG41
PE_RX_DN[11]	AG42
PE_RX_DN[12]	AF41
PE_RX_DN[13]	AF40
PE_RX_DN[14]	AE39
PE_RX_DN[15]	AE38
PE_RX_DP[0]	AJ38
PE_RX_DP[1]	AM37
PE_RX_DP[2]	AL38
PE_RX_DP[3]	AK37
PE_RX_DP[4]	AL40
PE_RX_DP[5]	AJ40
PE_RX_DP[6]	AL42
PE_RX_DP[7]	AJ41
PE_RX_DP[8]	AH39
PE_RX_DP[9]	AJ43
PE_RX_DP[10]	AH41
PE_RX_DP[11]	AG43

Signal	XY Coord
PE_RX_DP[12]	AF42
PE_RX_DP[13]	AG40
PE_RX_DP[14]	AE40
PE_RX_DP[15]	AF38
PE_TX_DN[0]	AU35
PE_TX_DN[1]	AV37
PE_TX_DN[2]	AY36
PE_TX_DN[3]	BA37
PE_TX_DN[4]	AW38
PE_TX_DN[5]	AY38
PE_TX_DN[6]	AW39
PE_TX_DN[7]	AV40
PE_TX_DN[8]	AU38
PE_TX_DN[9]	AT39
PE_TX_DN[10]	AR43
PE_TX_DN[11]	AR42
PE_TX_DN[12]	AN42
PE_TX_DN[13]	AP41
PE_TX_DN[14]	AP38
PE_TX_DN[15]	AP39
PE_TX_DP[0]	AT35
PE_TX_DP[1]	AV36
PE_TX_DP[2]	AW36
PE_TX_DP[3]	BA36
PE_TX_DP[4]	AW37
PE_TX_DP[5]	BA38
PE_TX_DP[6]	AY39
PE_TX_DP[7]	AW40
PE_TX_DP[8]	AV38
PE_TX_DP[9]	AU39
PE_TX_DP[10]	AT43
PE_TX_DP[11]	AR41
PE_TX_DP[12]	AP42
PE_TX_DP[13]	AP40
PE_TX_DP[14]	AR38
PE_TX_DP[15]	AN39
PECI	AH36
PECI_ID#	AK35
DDR_THERM#	AB5
RSVD_AF4	AF4
PM_SYNC	AN36

Signal	XY Coord
PRDY#	B41
PREQ#	C42
PROCHOT#	AL35
PSI#	AP7
QPI_CLKRX_DN	AR6
QPI_CLKRX_DP	AT6
QPI_CLKTX_DN	AE6
QPI_CLKTX_DP	AF6
QPI_COMP[0]	AL6
QPI_COMP[1]	AU34
QPI_RX_DN[0]	AV8
QPI_RX_DN[1]	AW7
QPI_RX_DN[2]	BA8
QPI_RX_DN[3]	AW5
QPI_RX_DN[4]	BA6
QPI_RX_DN[5]	AY5
QPI_RX_DN[6]	AU6
QPI_RX_DN[7]	AW3
QPI_RX_DN[8]	AU3
QPI_RX_DN[9]	AT2
QPI_RX_DN[10]	AR1
QPI_RX_DN[11]	AR5
QPI_RX_DN[12]	AN2
QPI_RX_DN[13]	AM1
QPI_RX_DN[14]	AM3
QPI_RX_DN[15]	AP4
QPI_RX_DN[16]	AN4
QPI_RX_DN[17]	AN6
QPI_RX_DN[18]	AM7
QPI_RX_DN[19]	AL8
QPI_RX_DP[0]	AU8
QPI_RX_DP[1]	AV7
QPI_RX_DP[2]	AY8
QPI_RX_DP[3]	AV5
QPI_RX_DP[4]	BA7
QPI_RX_DP[5]	AY6
QPI_RX_DP[6]	AU7
QPI_RX_DP[7]	AW4
QPI_RX_DP[8]	AU4
QPI_RX_DP[9]	AT3
QPI_RX_DP[10]	AT1

Signal	XY Coord
QPI_RX_DP[11]	AR4
QPI_RX_DP[12]	AP2
QPI_RX_DP[13]	AN1
QPI_RX_DP[14]	AM2
QPI_RX_DP[15]	AP3
QPI_RX_DP[16]	AM4
QPI_RX_DP[17]	AN5
QPI_RX_DP[18]	AM6
QPI_RX_DP[19]	AM8
QPI_TX_DN[0]	AH8
QPI_TX_DN[1]	AJ7
QPI_TX_DN[2]	AJ6
QPI_TX_DN[3]	AK5
QPI_TX_DN[4]	AK4
QPI_TX_DN[5]	AG6
QPI_TX_DN[6]	AJ2
QPI_TX_DN[7]	AJ1
QPI_TX_DN[8]	AH4
QPI_TX_DN[9]	AG2
QPI_TX_DN[10]	AF3
QPI_TX_DN[11]	AD1
QPI_TX_DN[12]	AD3
QPI_TX_DN[13]	AB3
QPI_TX_DN[14]	AE4
QPI_TX_DN[15]	AD4
QPI_TX_DN[16]	AC6
QPI_TX_DN[17]	AD7
QPI_TX_DN[18]	AE5
QPI_TX_DN[19]	AD8
QPI_TX_DP[0]	AG8
QPI_TX_DP[1]	AJ8
QPI_TX_DP[2]	AH6
QPI_TX_DP[3]	AK6
QPI_TX_DP[4]	AJ4
QPI_TX_DP[5]	AG7
QPI_TX_DP[6]	AJ3
QPI_TX_DP[7]	AK1
QPI_TX_DP[8]	AH3
QPI_TX_DP[9]	AH2
QPI_TX_DP[10]	AF2
QPI_TX_DP[11]	AE1

Signal	XY Coord
QPI_TX_DP[12]	AD2
QPI_TX_DP[13]	AC3
QPI_TX_DP[14]	AE3
QPI_TX_DP[15]	AC4
QPI_TX_DP[16]	AB6
QPI_TX_DP[17]	AD6
QPI_TX_DP[18]	AD5
QPI_TX_DP[19]	AC8
RSTIN#	AJ36
RSVD_A40	A40
RSVD_AG4	AG4
RSVD_AG5	AG5
RSVD_AH5	AH5
RSVD_AK2	AK2
RSVD_AL4	AL4
RSVD_AL5	AL5
RSVD_AL36	AL36
RSVD_AM40	AM40
RSVD_AM41	AM41
RSVD_AN33	AN33
RSVD_AN38	AN38
RSVD_AN40	AN40
RSVD_AP35	AP35
RSVD_AR34	AR34
RSVD_AR37	AR37
RSVD_AT4	AT4
RSVD_AT5	AT5
RSVD_AT42	AT42
RSVD_AU2	AU2
RSVD_AU37	AU37
RSVD_AU42	AU42
RSVD_AV1	AV1
RSVD_AV2	AV2
RSVD_AV42	AV42
RSVD_AV43	AV43
RSVD_AW2	AW2
RSVD_AW41	AW41
RSVD_AW42	AW42
RSVD_AY3	AY3
RSVD_AY4	AY4
RSVD_AY40	AY40

Signal	XY Coord
RSVD_AY41	AY41
RSVD_BA4	BA4
RSVD_BA40	BA40
RSVD_K15	K15
RSVD_K24	K24
RSVD_L15	L15
RSVD_L23	L23
SKTOCC#	AM36
SMB_CLK	AR36
SMB_DATA	AR35
SYS_ERR_STAT[0]#	AM35
SYS_ERR_STAT[1]#	AP36
SYS_ERR_STAT[2]#	AL34
TCLK	AH10
TDI	AJ9
TDI_M	AH33
TDO	AJ10
TDO_M	AL33
THERMTRIP#	AG37
TMS	AG10
TRST#	AH9
VCC	M11
VCC	M13
VCC	M15
VCC	M19
VCC	M21
VCC	M23
VCC	M25
VCC	M29
VCC	M31
VCC	M33
VCC	N11
VCC	N33
VCC	R11
VCC	R33
VCC	T11
VCC	T33
VCC	W11
VCC	AH11
VCC	AJ11
VCC	AJ33

Signal	XY Coord
VCC	AJ34
VCC	AK11
VCC	AK12
VCC	AK13
VCC	AK15
VCC	AK16
VCC	AK18
VCC	AK19
VCC	AK21
VCC	AK24
VCC	AK25
VCC	AK27
VCC	AK28
VCC	AK30
VCC	AK31
VCC	AK33
VCC	AL12
VCC	AL13
VCC	AL15
VCC	AL16
VCC	AL18
VCC	AL19
VCC	AL21
VCC	AL24
VCC	AL25
VCC	AL27
VCC	AL28
VCC	AL30
VCC	AL31
VCC	AM12
VCC	AM13
VCC	AM15
VCC	AM16
VCC	AM18
VCC	AM19
VCC	AM21
VCC	AM24
VCC	AM25
VCC	AM27
VCC	AM28
VCC	AM30

Signal	XY Coord
VCC	AM31
VCC	AN12
VCC	AN13
VCC	AN15
VCC	AN16
VCC	AN18
VCC	AN19
VCC	AN21
VCC	AN24
VCC	AN25
VCC	AN27
VCC	AN28
VCC	AN30
VCC	AN31
VCC	AP12
VCC	AP13
VCC	AP15
VCC	AP16
VCC	AP18
VCC	AP19
VCC	AP21
VCC	AP24
VCC	AP25
VCC	AP27
VCC	AP28
VCC	AP30
VCC	AP31
VCC	AR10
VCC	AR12
VCC	AR13
VCC	AR15
VCC	AR16
VCC	AR18
VCC	AR19
VCC	AR21
VCC	AR24
VCC	AR25
VCC	AR27
VCC	AR28
VCC	AR30
VCC	AR31

Signal	XY Coord
VCC	AT9
VCC	AT10
VCC	AT12
VCC	AT13
VCC	AT15
VCC	AT16
VCC	AT18
VCC	AT19
VCC	AT21
VCC	AT24
VCC	AT25
VCC	AT27
VCC	AT28
VCC	AT30
VCC	AT31
VCC	AU9
VCC	AU10
VCC	AU12
VCC	AU13
VCC	AU15
VCC	AU16
VCC	AU18
VCC	AU19
VCC	AU21
VCC	AU24
VCC	AU25
VCC	AU27
VCC	AU28
VCC	AU30
VCC	AU31
VCC	AV9
VCC	AV10
VCC	AV12
VCC	AV13
VCC	AV15
VCC	AV16
VCC	AV18
VCC	AV19
VCC	AV21
VCC	AV24
VCC	AV25

Signal	XY Coord
VCC	AV27
VCC	AV28
VCC	AV30
VCC	AV31
VCC	AW9
VCC	AW10
VCC	AW12
VCC	AW13
VCC	AW15
VCC	AW16
VCC	AW18
VCC	AW19
VCC	AW21
VCC	AW24
VCC	AW25
VCC	AW27
VCC	AW28
VCC	AW30
VCC	AW31
VCC	AY9
VCC	AY10
VCC	AY12
VCC	AY13
VCC	AY15
VCC	AY16
VCC	AY18
VCC	AY19
VCC	AY21
VCC	AY24
VCC	AY25
VCC	AY27
VCC	AY28
VCC	AY30
VCC	AY31
VCC	BA9
VCC	BA10
VCC	BA12
VCC	BA13
VCC	BA15
VCC	BA16
VCC	BA18

Signal	XY Coord
VCC	BA19
VCC	BA24
VCC	BA25
VCC	BA27
VCC	BA28
VCC	BA30
VCC_SENSE	AR9
VCCPLL	U33
VCCPLL	V33
VCCPLL	W33
VCCPWRGOOD	AR7
VDDQ	A9
VDDQ	A14
VDDQ	A19
VDDQ	A24
VDDQ	A29
VDDQ	B7
VDDQ	B12
VDDQ	B17
VDDQ	B22
VDDQ	B27
VDDQ	B32
VDDQ	C10
VDDQ	C15
VDDQ	C20
VDDQ	C25
VDDQ	C30
VDDQ	D13
VDDQ	D18
VDDQ	D23
VDDQ	D28
VDDQ	E11
VDDQ	E16
VDDQ	E21
VDDQ	E26
VDDQ	E31
VDDQ	F14
VDDQ	F19
VDDQ	F24
VDDQ	G17
VDDQ	G22

Signal	XY Coord
VDDQ	G27
VDDQ	H15
VDDQ	H20
VDDQ	H25
VDDQ	J18
VDDQ	J23
VDDQ	J28
VDDQ	K16
VDDQ	K21
VDDQ	K26
VDDQ	L14
VDDQ	L19
VDDQ	L24
VDDQ	M17
VDDQ	M27
VID[0]	AL10
VID[1]	AL9
VID[2]	AN9
VID[3]	AM10
VID[4]	AN10
VID[5]	AP9
VID[6]	AP8
VID[7]	AN8
VSS	A4
VSS	A6
VSS	A31
VSS	A35
VSS	A39
VSS	A41
VSS	B2
VSS	B37
VSS	B42
VSS	C5
VSS	C31
VSS	C32
VSS	C35
VSS	C40
VSS	C43
VSS	D3
VSS	D8
VSS	D30

Signal	XY Coord
VSS	D31
VSS	D33
VSS	D38
VSS	D43
VSS	E1
VSS	E6
VSS	E28
VSS	E36
VSS	E41
VSS	F4
VSS	F9
VSS	F28
VSS	F29
VSS	F34
VSS	F39
VSS	G2
VSS	G7
VSS	G12
VSS	G28
VSS	G32
VSS	G37
VSS	G42
VSS	H5
VSS	H10
VSS	H29
VSS	H30
VSS	H35
VSS	H40
VSS	J3
VSS	J8
VSS	J13
VSS	J29
VSS	J33
VSS	J38
VSS	J43
VSS	K1
VSS	K6
VSS	K11
VSS	K27
VSS	K29
VSS	K31

Signal	XY Coord
VSS	K36
VSS	K41
VSS	L4
VSS	L9
VSS	L29
VSS	L34
VSS	L39
VSS	M2
VSS	M7
VSS	M12
VSS	M14
VSS	M16
VSS	M18
VSS	M20
VSS	M22
VSS	M24
VSS	M26
VSS	M28
VSS	M30
VSS	M32
VSS	M37
VSS	M42
VSS	N5
VSS	N10
VSS	N35
VSS	N40
VSS	P3
VSS	P8
VSS	P11
VSS	P33
VSS	P38
VSS	P43
VSS	R1
VSS	R6
VSS	R36
VSS	R41
VSS	T4
VSS	T9
VSS	T34
VSS	T39
VSS	U2

Signal	XY Coord
VSS	U7
VSS	U37
VSS	U42
VSS	V5
VSS	V10
VSS	V35
VSS	V40
VSS	W3
VSS	W8
VSS	W38
VSS	W43
VSS	Y1
VSS	Y6
VSS	Y11
VSS	Y33
VSS	Y36
VSS	Y41
VSS	AA3
VSS	AA9
VSS	AA34
VSS	AA39
VSS	AA40
VSS	AB4
VSS	AB7
VSS	AB37
VSS	AB42
VSS	AC2
VSS	AC5
VSS	AC7
VSS	AC9
VSS	AC36
VSS	AC40
VSS	AD11
VSS	AD33
VSS	AD37
VSS	AD39
VSS	AD43
VSS	AE2
VSS	AE7
VSS	AE41
VSS	AF5

Signal	XY Coord
VSS	AF35
VSS	AF39
VSS	AF43
VSS	AG3
VSS	AG9
VSS	AG11
VSS	AG33
VSS	AG35
VSS	AG38
VSS	AH1
VSS	AH7
VSS	AH34
VSS	AH38
VSS	AH40
VSS	AH42
VSS	AJ5
VSS	AJ39
VSS	AK3
VSS	AK7
VSS	AK9
VSS	AK10
VSS	AK14
VSS	AK17
VSS	AK20
VSS	AK22
VSS	AK23
VSS	AK26
VSS	AK29
VSS	AK32
VSS	AK34
VSS	AK36
VSS	AK41
VSS	AK43
VSS	AL1
VSS	AL2
VSS	AL7
VSS	AL11
VSS	AL14
VSS	AL17
VSS	AL20
VSS	AL22

Signal	XY Coord
VSS	AL23
VSS	AL26
VSS	AL29
VSS	AL32
VSS	AL37
VSS	AL39
VSS	AM5
VSS	AM9
VSS	AM11
VSS	AM14
VSS	AM17
VSS	AM20
VSS	AM22
VSS	AM23
VSS	AM26
VSS	AM29
VSS	AM32
VSS	AM39
VSS	AM42
VSS	AN3
VSS	AN7
VSS	AN11
VSS	AN14
VSS	AN17
VSS	AN20
VSS	AN22
VSS	AN23
VSS	AN26
VSS	AN29
VSS	AN32
VSS	AN34
VSS	AN35
VSS	AN41
VSS	AP1
VSS	AP5
VSS	AP6
VSS	AP10
VSS	AP11
VSS	AP14
VSS	AP17
VSS	AP20

Signal	XY Coord
VSS	AP22
VSS	AP23
VSS	AP26
VSS	AP29
VSS	AP32
VSS	AP37
VSS	AP43
VSS	AR2
VSS	AR3
VSS	AR11
VSS	AR14
VSS	AR17
VSS	AR20
VSS	AR22
VSS	AR23
VSS	AR26
VSS	AR29
VSS	AR32
VSS	AR33
VSS	AR39
VSS	AT7
VSS	AT8
VSS	AT11
VSS	AT14
VSS	AT17
VSS	AT20
VSS	AT22
VSS	AT23
VSS	AT26
VSS	AT29
VSS	AT32
VSS	AT34
VSS	AT37
VSS	AT38
VSS	AT41
VSS	AU1
VSS	AU5
VSS	AU11
VSS	AU14
VSS	AU17
VSS	AU20

Signal	XY Coord
VSS	AU22
VSS	AU23
VSS	AU26
VSS	AU29
VSS	AU32
VSS	AU36
VSS	AU40
VSS	AU43
VSS	AV4
VSS	AV11
VSS	AV14
VSS	AV17
VSS	AV20
VSS	AV22
VSS	AV23
VSS	AV26
VSS	AV29
VSS	AV32
VSS	AV35
VSS	AV39
VSS	AV41
VSS	AW1
VSS	AW6
VSS	AW8
VSS	AW11
VSS	AW14
VSS	AW17
VSS	AW20
VSS	AW22
VSS	AW23
VSS	AW26
VSS	AW29
VSS	AW32
VSS	AW35
VSS	AY2
VSS	AY7
VSS	AY11
VSS	AY14
VSS	AY17
VSS	AY20
VSS	AY22

Signal	XY Coord
VSS	AY23
VSS	AY26
VSS	AY29
VSS	AY32
VSS	AY37
VSS	AY42
VSS	BA3
VSS	BA5
VSS	BA11
VSS	BA14
VSS	BA17
VSS	BA20
VSS	BA26
VSS	BA29
VSS	BA39
VSS_SENSE	AR8
VSS_SENSE_VTT	AE37
VTT_VID[2]	AV3
VTT_VID[3]	AF7
VTT_VID[4]	AV6
VTTA	AD10
VTTA	AE10
VTTA	AE11
VTTA	AE33
VTTA	AF11
VTTA	AF33
VTTA	AF34
VTTA	AG34
VTTD	AA10
VTTD	AA11
VTTD	AA33
VTTD	AB8
VTTD	AB9
VTTD	AB10
VTTD	AB11
VTTD	AB33
VTTD	AB34
VTTD	AC10
VTTD	AC11
VTTD	AC33
VTTD	AC34

Signal	XY Coord
VTTD	AC35
VTTD	AD9
VTTD	AD34
VTTD	AD35
VTTD	AD36
VTTD	AE8
VTTD	AE9
VTTD	AE34
VTTD	AE35
VTTD	AF8
VTTD	AF9
VTTD	AF36
VTTD	AF37
VTTD_SENSE	AE36
VTTPWRGOOD	AH37

§ §

13.0 Electrical Specifications

13.1 Processor Signaling

The Intel® Xeon® processor C5500/C3500 series includes 1366 lands that utilize various signaling technologies. Signals are grouped by electrical characteristics and buffer type into various signal groups. These include Intel® QuickPath Interconnect, DDR3 Channel A, DDR3 Channel B, DDR3 Channel C, PCI Express, SMBus, DMI, Platform Environmental Control Interface (PECI), Clock, Reset and Miscellaneous, Thermal, Test Access Port (TAP), and Processor Core Power, Power Sequencing, and No Connect/Reserved signals. See Table 159 for details.

Detailed layout, routing, and termination guidelines corresponding to these signal groups can be found in the applicable platform design guide. See Section 1.9, "Related Documents".

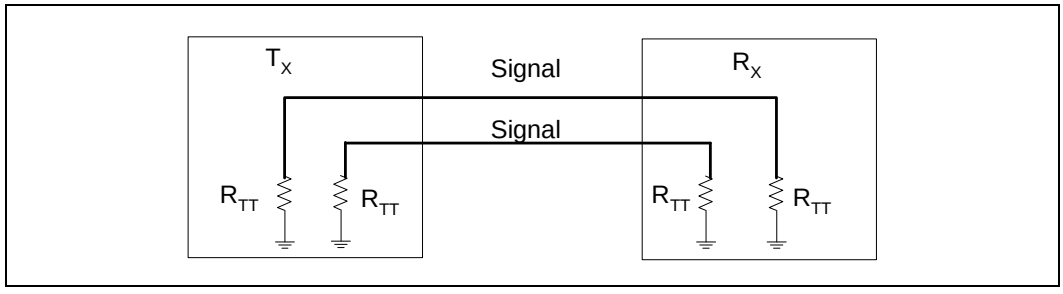
Intel strongly recommends performing analog simulations of all interfaces.

13.1.1 Intel® QuickPath Interconnect

The Intel® Xeon® processor C5500/C3500 series provides one Intel® QuickPath Interconnect port for high-speed serial transfer between other enabled components. Each port consists of two uni-directional links (for transmit and receive). A differential signaling scheme is utilized, which consists of opposite-polarity (D_P, D_N) signal pairs.

On-die termination (ODT) is included on the processor silicon and terminated to V_{SS}. Intel chipsets also provide ODT, thus eliminating the need to terminate on the system board. Figure 83 illustrates the active ODT.

Figure 83. Active ODT for a Differential Link Example



13.1.2 DDR3 Signal Groups

The memory interface utilizes DDR3 technology, which consists of numerous signal groups for each of the three memory channels. Each group consists of multiple signals, which may utilize various signaling technologies. See Table 159 for further details.

On-Die Termination (ODT) is a feature that allows a DRAM device to turn on/off internal

termination resistance for each DQ and DQS/DQS# signal via the ODT control pin. The ODT feature improves signal integrity of the memory channel by allowing the DRAM controller to independently turn on or off the termination resistance for any or all DRAM devices themselves instead of on the motherboard.

13.1.3 Platform Environmental Control Interface (PECI)

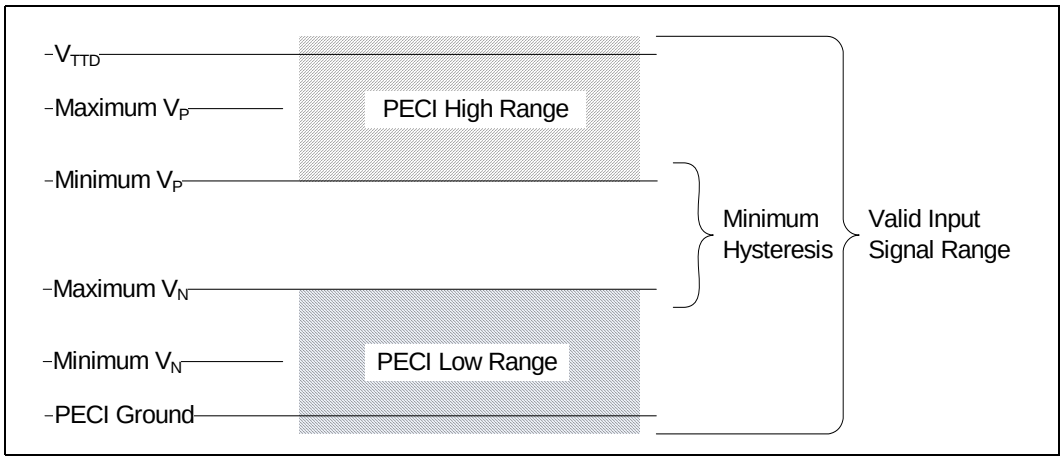
PECI is an Intel proprietary interface that provides a communication channel between Intel processors and chipset components to external thermal monitoring devices. The Intel® Xeon® processor C5500/C3500 series contains a Digital Thermal Sensor (DTS) that reports a relative die temperature as an offset from Thermal Control Circuit (TCC) activation temperature. Temperature sensors located throughout the die are implemented as analog-to-digital converters calibrated at the factory. PECI provides an interface for external devices to read processor temperature, perform processor manageability functions, and manage processor interface tuning and diagnostics. See the *Intel® Xeon® Processor C5500/C3500 Series Thermal / Mechanical Design Guide* for processor-specific implementation details for PECI. Generic PECI specification details are out of the scope of this document.

The PECI interface operates at a nominal voltage set by V_{TTD}. The set of DC electrical specifications shown in Table 170 is used with devices normally operating from a V_{TTD} interface supply.

13.1.3.1 Input Device Hysteresis

The PECI client and host input buffers must use a Schmitt-triggered input design for improved noise immunity. See Figure 84 and Table 170.

Figure 84. Input Device Hysteresis



13.1.4 PCI Express/DMI

The PCI Express* interface signals are driven by transceivers designed specifically for high-speed serial communication. All PCI Express signals are fully differential and operate in a current mode, rather than a voltage mode. These interfaces support A/C coupling to facilitate communication across independent power supply domains and signal at a rate well above the flight time of the interface. (The DMI interface on the legacy processor also supports DC coupling.) The Intel® Xeon® processor C5500/C3500 series supports the *PCI Express Base Specification*, Revision 2.0.

- Point-to-point, serial bi-directional interconnect
- The processor provides up to 16 PCI Express Gen 2 (5 GT/s) lanes

  - — x16 lanes can be bifurcated to support Gen 1/2 combinations of x8 and x4 links
- Intel® Xeon® processor C5500/C3500 series x4 port: 2.5 GT/s.
- Each signal is 8 b/10 b encoded with an embedded clock
- Signaling bit rate of 5 Gbit/sec/lane/direction; for a x4 link, bandwidth is 2 GB/sec in each direction
- Hot Insertion and Removal supported with the addition of Hot-Plug control circuitry
- Boot processor provides one x4 DMI link to the PCH
- Application processor in a dual processor configuration provides one x4 PCI Express Gen 1 (2.5 GT/s) link
- The PCI Express port (muxed with DMI) is only supported in a DP configuration and is not supported on the boot processor.

The Direct Media Interface (DMI2) in the Intel® Xeon® processor C5500/C3500 series IIO is responsible for sending and receiving packets/commands to other components in the system, e.g. PCH. The DMI is an extension of the standard PCI Express specification with special commands and features added to mimic the legacy Hub Interface. DMI2, supported by Intel® Xeon® processor C5500/C3500 series, is the second generation extension of DMI. Further details on DMI2 can be obtained from the DMI Specification, Revision 2.0. DMI connects the processor and the PCH chip-to-chip.

- The DMI is similar to a four-lane PCI Express supporting up to 1 GB/s of bandwidth in each direction.
- Only DMI x4 configuration is supported.
- In DP configurations, the DMI port of the “Non-Legacy” processor may be configured as a single PCIe port, supporting PCIe Gen1 only.

13.1.5 SMBus Interface

SMBus interface consists of two interface pins; one is a clock, and the other is serial data. Multiple initiator and target devices may be electrically present on the same pair of signals. Each target recognizes a start signaling semantic, and recognizes its own 7-bit address to identify pertinent bus traffic.

The Intel® Xeon® processor C5500/C3500 series IO SMBus acts as a slave and may be used to give a BMC out-of-band access to various IO components. The SMBus on processor is SMBus 2.0 compliant. For more details on the SMBus protocol see the System Management Bus Specification 2.0.

- Connected globally to the processors, and to the PCH through a common shared bus hierarchy.
- Low pin count, low speed management interface.
- Provides access to configuration status registers (CSRs).

13.1.6 Clock Signals

The processor core, processor uncore, Intel® QuickPath Interconnect link, and DDR3 memory interface frequencies are generated from BCLK_DP and BCLK_DN signals. There is no direct link between core frequency and Intel® QuickPath Interconnect link frequency (e.g., no core frequency to Intel® QuickPath Interconnect multiplier). The processor maximum core frequency, Intel® QuickPath Interconnect link frequency and DDR3 memory frequency are set during manufacturing. It is possible to override the processor core frequency setting using software. This permits operation at lower core frequencies than the factory set maximum core frequency.

The processor core frequency is configured during reset by using values stored within the device during manufacturing. The stored value sets the lowest core multiplier at which the particular processor can operate. If higher speeds are desired, the appropriate ratio can be configured via the IA32_PERF_CTL MSR (MSR 199h); Bits [15:0].

Clock multiplying within the processor is provided by the internal phase locked loop (PLL), which requires a constant frequency BCLK_DP, BCLK_DN input, with exceptions for spread spectrum clocking. DC specifications for the BCLK_DP, BCLK_DN inputs are provided in Table 171. These specifications must be met while also meeting the associated signal quality specifications.

Details regarding BCLK_DP, BCLK_DN driver specifications are provided in the *CK410B Clock Synthesizer/Driver Design Guidelines*.

13.1.7 Reset and Miscellaneous

The Intel® Xeon® processor C5500/C3500 series includes signals that provide a variety of functions. Details are in Table 159 and in the applicable platform design guide.

See Table 172 and for DC specifications.

13.1.8 Thermal

Intel® Xeon® processor C5500/C3500 series includes signals that support the thermal management feature. These thermal signals serve as indication and protection when the processor reaches a potential overheating condition. Details are in Table 159.

See Table 173 for DC specifications.

13.1.9 Test Access Port (TAP) Signals

Due to the voltage levels supported by other components in the Test Access Port (TAP) logic, it is recommended that the processor(s) be first in the TAP chain and followed by any other components within the system. A translation buffer should be used to connect to the rest of the chain unless one of the other components is capable of accepting an input of the appropriate voltage. Similar considerations must be made for TDI, TDI_M, and TDO_M, TCLK, TDO, TMS, and TRST#. Two copies of each signal may be required with each driving a different voltage level.

Processor TAP signal DC specifications are in Table 172.

13.1.10 Power / Other Signals

Processors also include various other signals including power/ground, sense points, and analog inputs. Details are in Table 159 and in the applicable platform design guide.

Table 155 outlines the required voltage supplies necessary to support Intel® Xeon® processor C5500/C3500 series.

Table 155. Processor Power Supply Voltages[1]

Power Rail	Nominal Voltage	Notes
V_{CC}	See Table 163; Figure 86	Each processor includes a dedicated VR11.1 regulator.
V_{CCPLL}	1.80 V	Each processor includes dedicated V_{CCPLL} and PLL circuits.
V_{DDQ}	1.50 V	Each processor and DDR3 stack shares a dedicated voltage regulator.
V_{TTA}, V_{TTD}	See Table 173	Each processor includes a dedicated VR11.0 regulator. V_{TT} = V_{TTA} = V_{TTD}; P1V1_Vtt is VID[4:2] controlled, VID range is 1.0255-1.2000V. The tolerance is +/- 2% at the processor pin. (This assumes that the filter circuit described in the *Picket Post: Intel® Xeon® Processor C5500/C3500 Series with the Intel® 3420 Chipset Platform Design Guide (PDG)* is used.)

Note:
1. See Table 162 for voltage and current specifications.

Further platform and processor power delivery details are in the *Picket Post: Intel® Xeon® Processor C5500/C3500 Series with the Intel® 3420 Chipset Platform Design Guide (PDG)*.

13.1.10.1 Power and Ground Lands

For clean on-chip power distribution, processors include lands for all required voltage supplies.

The processor has V_{CC}, V_{TT}, V_{DDQ}, V_{CCPLL}, and V_{SS} inputs for on-chip power distribution. All power lands must be connected to their respective processor power planes, while all V_{SS} lands must be connected to the system ground plane. See the *Picket Post: Intel® Xeon® Processor C5500/C3500 Series with the Intel® 3420 Chipset Platform Design Guide (PDG)* for decoupling, voltage plane and routing guidelines for each power supply voltage.

13.1.10.2 Decoupling Guidelines

Due to its large number of transistors and high internal clock speeds, the Intel® Xeon® processor C5500/C3500 series is capable of generating large current swings between low and full power states. This may cause voltages on power planes to sag below their minimum values if bulk decoupling is not adequate. Larger bulk storage (C_{BULK}), such as electrolytic capacitors, supply current during longer lasting changes in current demand, for example coming out of an idle condition. Similarly, they act as a storage well for current when entering an idle condition from a running condition. Care must be taken in the board design to ensure that the voltages provided to the processor remain within the specifications listed in Table 162. Failure to do so can result in timing violations or reduced lifetime of the processor. For further information, see the *Picket Post: Intel® Xeon® Processor C5500/C3500 Series with the Intel® 3420 Chipset Platform Design Guide (PDG)*.

13.1.10.3 Processor V_{CC} Voltage Identification (VID) Signals

The Voltage Identification (VID) specification for the V_{CC} voltage is defined by the *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator-Down (EVRD) 11.1 Design Guidelines*, Revision 1.5. The voltage set by the VID signals is the maximum reference voltage regulator (VR) output to be delivered to the processor V_{CC} lands. VID signals are CMOS push/pull outputs. See Table 172 for the DC specifications for these and other processor signals.

Individual processor VID values may be calibrated during manufacturing such that two processor units with the same core frequency may have different default VID settings.

The processor uses eight voltage identification signals, VID[7:0], to support automatic selection of core power supply voltages. Table 156 specifies the voltage level corresponding to the state of VID[7:0]. A '1' in this table refers to a high voltage level and a '0' refers to a low voltage level. If the processor socket is empty (SKTOCC# high), or the voltage regulation circuit cannot supply the voltage that is requested, the voltage regulator must disable itself. See the *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator-Down (EVRD) 11.1 Design Guidelines*, Revision 1.5 for further details.

The processor provides the ability to operate while transitioning to an adjacent VID and its associated processor core voltage (V_{CC}). This is represented by a DC shift in the loadline. It should be noted that a low-to-high or high-to-low voltage state change may result in as many VID transitions as necessary to reach the target core voltage. Transitions above the maximum specified VID are not permitted. Table 162 includes VID step sizes and DC shift ranges. Minimum and maximum voltages must be maintained as shown in Table 163.

The VRM or EVRD utilized must be capable of regulating its output to the value defined by the new VID. DC specifications for dynamic VID transitions are included in Table 172. See the *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator-Down (EVRD) 11.1 Design Guidelines,* Revision 1.5 for further details.

Power source characteristics must be guaranteed to be stable whenever the supply to the voltage regulator is stable.

Table 156. Voltage Identification Definition (Sheet 1 of 5)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	V_{CC_MAX}
0	0	0	0	0	0	0	0	OFF
0	0	0	0	0	0	0	1	OFF
0	0	0	0	0	0	1	0	1.60000
0	0	0	0	0	0	1	1	1.59375
0	0	0	0	0	1	0	0	1.58750
0	0	0	0	0	1	0	1	1.58125
0	0	0	0	0	1	1	0	1.57500
0	0	0	0	0	1	1	1	1.56875
0	0	0	0	1	0	0	0	1.56250
0	0	0	0	1	0	0	1	1.55625
0	0	0	0	1	0	1	0	1.55000
0	0	0	0	1	0	1	1	1.54375
0	0	0	0	1	1	0	0	1.53750
0	0	0	0	1	1	0	1	1.53125
0	0	0	0	1	1	1	0	1.52500
0	0	0	0	1	1	1	1	1.51875
0	0	0	1	0	0	0	0	1.51250
0	0	0	1	0	0	0	1	1.50625
0	0	0	1	0	0	1	0	1.50000
0	0	0	1	0	0	1	1	1.49375
0	0	0	1	0	1	0	0	1.48750

Table 156. Voltage Identification Definition (Sheet 2 of 5)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	V_{CC_MAX}
0	0	0	1	0	1	0	1	1.48125
0	0	0	1	0	1	1	0	1.47500
0	0	0	1	0	1	1	1	1.46875
0	0	0	1	1	0	0	0	1.46250
0	0	0	1	1	0	0	1	1.45625
0	0	0	1	1	0	1	0	1.45000
0	0	0	1	1	0	1	1	1.44375
0	0	0	1	1	1	0	0	1.43750
0	0	0	1	1	1	0	1	1.43125
0	0	0	1	1	1	1	0	1.42500
0	0	0	1	1	1	1	1	1.41875
0	0	1	0	0	0	0	0	1.41250
0	0	1	0	0	0	0	1	1.40625
0	0	1	0	0	0	1	0	1.40000
0	0	1	0	0	0	1	1	1.39375
0	0	1	0	0	1	0	0	1.38750
0	0	1	0	0	1	0	1	1.38125
0	0	1	0	0	1	1	0	1.37500
0	0	1	0	0	1	1	1	1.36875
0	0	1	0	1	0	0	0	1.36250
0	0	1	0	1	0	0	1	1.35625
0	0	1	0	1	0	1	0	1.35000
0	0	1	0	1	0	1	1	1.34375
0	0	1	0	1	1	0	0	1.33750
0	0	1	0	1	1	0	1	1.33125
0	0	1	0	1	1	1	0	1.32500
0	0	1	0	1	1	1	1	1.31875
0	0	1	1	0	0	0	0	1.31250
0	0	1	1	0	0	0	1	1.30625
0	0	1	1	0	0	1	0	1.30000
0	0	1	1	0	0	1	1	1.29375
0	0	1	1	0	1	0	0	1.28750
0	0	1	1	0	1	0	1	1.28125
0	0	1	1	0	1	1	0	1.27500
0	0	1	1	0	1	1	1	1.26875
0	0	1	1	1	0	0	0	1.26250
0	0	1	1	1	0	0	1	1.25625
0	0	1	1	1	0	1	0	1.25000
0	0	1	1	1	0	1	1	1.24375
0	0	1	1	1	1	0	0	1.23750

Table 156. Voltage Identification Definition (Sheet 3 of 5)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	V_{CC_MAX}
0	0	1	1	1	1	0	1	1.23125
0	0	1	1	1	1	1	0	1.22500
0	0	1	1	1	1	1	1	1.21875
0	1	0	0	0	0	0	0	1.21250
0	1	0	0	0	0	0	1	1.20625
0	1	0	0	0	0	1	0	1.20000
0	1	0	0	0	0	1	1	1.19375
0	1	0	0	0	1	0	0	1.18750
0	1	0	0	0	1	0	1	1.18125
0	1	0	0	0	1	1	0	1.17500
0	1	0	0	0	1	1	1	1.16875
0	1	0	0	1	0	0	0	1.16250
0	1	0	0	1	0	0	1	1.15625
0	1	0	0	1	0	1	0	1.15000
0	1	0	0	1	0	1	1	1.14375
0	1	0	0	1	1	0	0	1.13750
0	1	0	0	1	1	0	1	1.13125
0	1	0	0	1	1	1	0	1.12500
0	1	0	0	1	1	1	1	1.11875
0	1	0	1	0	0	0	0	1.11250
0	1	0	1	0	0	0	1	1.10625
0	1	0	1	0	0	1	0	1.10000
0	1	0	1	0	0	1	1	1.09375
0	1	0	1	0	1	0	0	1.08750
0	1	0	1	0	1	0	1	1.08125
0	1	0	1	0	1	1	0	1.07500
0	1	0	1	0	1	1	1	1.06875
0	1	0	1	1	0	0	0	1.06250
0	1	0	1	1	0	0	1	1.05625
0	1	0	1	1	0	1	0	1.05000
0	1	0	1	1	0	1	1	1.04375
0	1	0	1	1	1	0	0	1.03750
0	1	0	1	1	1	0	1	1.03125
0	1	0	1	1	1	1	0	1.02500
0	1	0	1	1	1	1	1	1.01875
0	1	1	0	0	0	0	0	1.01250
0	1	1	0	0	0	0	1	1.00625
0	1	1	0	0	0	1	0	1.00000
0	1	1	0	0	0	1	1	0.99375
0	1	1	0	0	1	0	0	0.98750

Table 156. Voltage Identification Definition (Sheet 4 of 5)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	V_{CC_MAX}
0	1	1	0	0	1	0	1	0.98125
0	1	1	0	0	1	1	0	0.97500
0	1	1	0	0	1	1	1	0.96875
0	1	1	0	1	0	0	0	0.96250
0	1	1	0	1	0	0	1	0.95625
0	1	1	0	1	0	1	0	0.95000
0	1	1	0	1	0	1	1	0.94375
0	1	1	0	1	1	0	0	0.93750
0	1	1	0	1	1	0	1	0.93125
0	1	1	0	1	1	1	0	0.92500
0	1	1	0	1	1	1	1	0.91875
0	1	1	1	0	0	0	0	0.91250
0	1	1	1	0	0	0	1	0.90625
0	1	1	1	0	0	1	0	0.90000
0	1	1	1	0	0	1	1	0.89375
0	1	1	1	0	1	0	0	0.88750
0	1	1	1	0	1	0	1	0.88125
0	1	1	1	0	1	1	0	0.87500
0	1	1	1	0	1	1	1	0.86875
0	1	1	1	1	0	0	0	0.86250
0	1	1	1	1	0	0	1	0.85625
0	1	1	1	1	0	1	0	0.85000
0	1	1	1	1	0	1	1	0.84375
0	1	1	1	1	1	0	0	0.83750
0	1	1	1	1	1	0	1	0.83125
0	1	1	1	1	1	1	0	0.82500
0	1	1	1	1	1	1	1	0.81875
1	0	0	0	0	0	0	0	0.81250
1	0	0	0	0	0	0	1	0.80625
1	0	0	0	0	0	1	0	0.80000
1	0	0	0	0	0	1	1	0.79375
1	0	0	0	0	1	0	0	0.78750
1	0	0	0	0	1	0	1	0.78125
1	0	0	0	0	1	1	0	0.77500
1	0	0	0	0	1	1	1	0.76875
1	0	0	0	1	0	0	0	0.76250
1	0	0	0	1	0	0	1	0.75625
1	0	0	0	1	0	1	0	0.75000
1	0	0	0	1	0	1	1	0.74375
1	0	0	0	1	1	0	0	0.73750

Table 156. Voltage Identification Definition (Sheet 5 of 5)

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	V_{CC_MAX}
1	0	0	0	1	1	0	1	0.73125
1	0	0	0	1	1	1	0	0.72500
1	0	0	0	1	1	1	1	0.71875
1	0	0	1	0	0	0	0	0.71250
1	0	0	1	0	0	0	1	0.70625
1	0	0	1	0	0	1	0	0.70000
1	0	0	1	0	0	1	1	0.69375
1	0	0	1	0	1	0	0	0.68750
1	0	0	1	0	1	0	1	0.68125
1	0	0	1	0	1	1	0	0.67500
1	0	0	1	0	1	1	1	0.66875
1	0	0	1	1	0	0	0	0.66250
1	0	0	1	1	0	0	1	0.65625
1	0	0	1	1	0	1	0	0.65000
1	0	0	1	1	0	1	1	0.64375
1	0	0	1	1	1	0	0	0.63750
1	0	0	1	1	1	0	1	0.63125
1	0	0	1	1	1	1	0	0.62500
1	0	0	1	1	1	1	1	0.61875
1	0	1	0	0	0	0	0	0.61250
1	0	1	0	0	0	0	1	0.60625
1	0	1	0	0	0	1	0	0.60000
1	0	1	0	0	0	1	1	0.59375
1	0	1	0	0	1	0	0	0.58750
1	0	1	0	0	1	0	1	0.58125
1	0	1	0	0	1	1	0	0.57500
1	0	1	0	0	1	1	1	0.56875
1	0	1	0	1	0	0	0	0.56250
1	0	1	0	1	0	0	1	0.55625
1	0	1	0	1	0	1	0	0.55000
1	0	1	0	1	0	1	1	0.54375
1	0	1	0	1	1	0	0	0.53750
1	0	1	0	1	1	0	1	0.53125
1	0	1	0	1	1	1	0	0.52500
1	0	1	0	1	1	1	1	0.51875
1	0	1	1	0	0	0	0	0.51250
1	0	1	1	0	0	0	1	0.50625
1	0	1	1	0	0	1	0	0.50000
1	1	1	1	1	1	1	0	OFF
1	1	1	1	1	1	1	1	OFF

Note:

- The expected voltage range is 1.35-0.75v.
- When the “11111111” VID pattern is observed, or when the SKTOCC# pin is high, the voltage regulator output should be disabled.
- Shading denotes the expected VID range of the processor.
- The VID range includes VID transitions that may be initiated by thermal events, Extended HALT state transitions (see Section 8.2, “Processor Core Power Management”), higher C-States (see Section 8.2.4, “Core C-States”) or Enhanced Intel SpeedStep® Technology transitions (see Section 8.2.1, “Enhanced Intel SpeedStep® Technology”). **The Extended HALT state must be enabled for the processor to remain within its specifications.**
- Once the VRM/EVRD is operating after power-up, if either the Output Enable signal is de-asserted or a specific VID off code is received, the VRM/EVRD must turn off its output (the output should go to high impedance) within 500 ms and latch off until power is cycled. See the *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator-Down (EVRD) 11.1 Design Guidelines*, Revision 1.5.

13.1.10.3.1 Power-On Configuration (POC) Logic

VID[7:0] signals also serve a second function. During power-up, Power-On Configuration POC[7:0] functionality is multiplexed onto these signals via 1-5 kOhms pull-up or pull down resistors located on the board. These values provide voltage regulator keying (VID[7]), inform the processor of the platforms power delivery capabilities (MSID[2:0]), and program the gain applied to the ISENSE input (CSC[2:0]). Table 157 maps VID signals to the corresponding POC functionality.

Table 157. Power-On Configuration (POC[7:0]) Decode

Function	Bits	POC Settings		Description
VR_Key	VID[7]	0b for VR11.1		Electronic safety key distinguishing VR11.1
Spare	VID[6]	0b (default)		Reserved for future use
CSC[2:0]	VID[5:3]	000 001 010 011 100 101 110 111	Feature Disabled ICC_MAX ≤ 40A 40A ≤ ICC_MAX ≤ 60A 60A ≤ ICC_MAX ≤ 80A 80A ≤ ICC_MAX ≤ 100A 100A ≤ ICC_MAX ≤ 120A 120A ≤ ICC_MAX ≤ 140A 140A ≤ ICC_MAX ≤ 180A	Current Sensor Configuration (CSC) programs the gain applied to the ISENSE A/D output. ISENSE data is then used to dynamically calculate current and power. See the *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator-Down (EVRD) 11.1 Design Guidelines*, Revision 1.5 for further details on the IMON signal.
MSID[2:0]	VID[2:0]	000 001 010 011 100 101 110 111	Undefined Undefined Undefined 60W TDP / 80A ICC_MAX 80W TDP / 100A ICC_MAX 95W TDP / 120A ICC_MAX 130W TDP / 150A ICC_MAX Undefined	MSID[2:0] signals are provided to indicate the Market Segment for the processor and may be used for future processor compatibility or keying. See Figure 85 for platform timing requirements of the MSID[2:0] signals.

Some POC signals include specific timing requirements. See the following section for details.

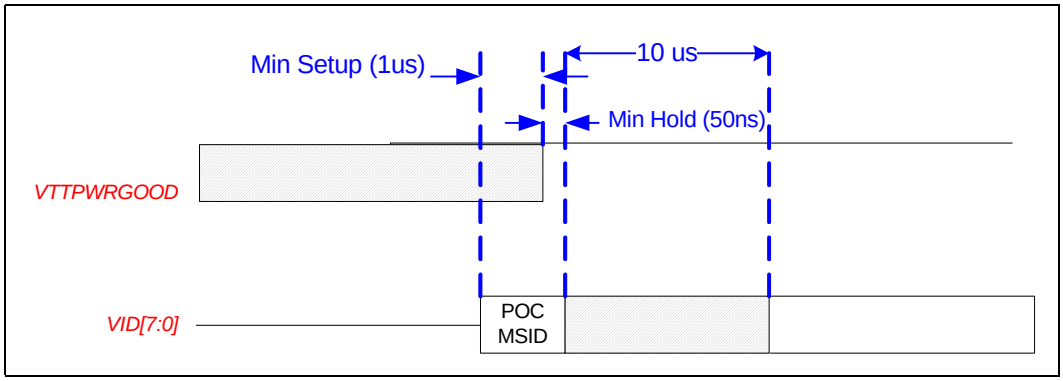
13.1.10.3.2 Power-On Configuration (POC)

Several configuration options can be configured by hardware. Power-on configuration (POC) functionality is either MUx'ed onto VID signals (see Section 13.1.10.3) or sampled on the active-to-inactive transition of RSTIN#. For specifics on these options, See Table 158.

Requests to execute Built-In Self Test (BIST) are not selected by hardware, but rather passed across the Intel® QuickPath Interconnect link during initialization.

Figure 85 outlines the timing associated with VID[2:0]/MSID[2:0] sampling. After OUTEN is asserted, the VID[7:0] CMOS drivers (typically 50 Ohms up/down impedance) over-ride the POC pull-up/down resistors located on the baseboard and drive the necessary VID pattern.

Figure 85. MSID Timing Requirement



13.1.10.4 Processor V_{TT} Voltage Identification (VTT_VID) Signals

The VTT Voltage Identification (VTT_VID) specification is defined by the *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator-Down (EVRD) 11.1 Design Guidelines*, Revision 1.5. The voltage set by the VTT_VID signals is the typical reference voltage regulator (VR) output to be delivered to the processor V_{TTA} and V_{TTD} lands. It is expected that one regulator will supply all V_{TTA} and V_{TTD} lands. VTT_VID signals are CMOS push/pull outputs. See Table 172 for the DC specifications for these signals.

Individual processor VTT_VID values may be calibrated during manufacturing such that two processor units with the same core frequency may have different default VTT_VID settings.

The processor utilizes three voltage identification signals to support automatic selection of power supply voltages. These correspond to VTT_VID[4:2] as defined in the *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator-Down (EVRD) 11.1 Design Guidelines*, Revision 1.5. The V_{TT} voltage level delivered to the processor lands must also encompass a 20 mV offset (See Table 158; V_{TT_TYP}) above the voltage level corresponding to the state of the VTT_VID[7:0] signals (See Table 158; VR 11.0 Voltage).

Power source characteristics must be guaranteed to be stable whenever the supply to the voltage regulator is stable.

Note: For available SKUs, see Table 1, "Available SKUs". For voltage and current specifications, see Table 162.

Table 158. V_{TT} Voltage Identification Definition

VID7	VID6	VID5	VID4	VID3	VID2	VID1	VID0	VR 11.0 Voltage	V_{TT_TYP} (Voltage + Offset)
0	1	0	0	0	0	1	0	1.200V	1.220V
0	1	0	0	0	1	1	0	1.175V	1.195V
0	1	0	0	1	0	1	0	1.150V	1.170V
0	1	0	0	1	1	1	0	1.125V	1.145V
0	1	0	1	0	0	1	0	1.100V	1.120V
0	1	0	1	0	1	1	0	1.075V	1.095V
0	1	0	1	1	0	1	0	1.050V	1.070V
0	1	0	1	1	1	1	0	1.025V	1.045V

13.1.11 Reserved or Unused Signals

Unless otherwise specified, all Reserved (RSVD) signals should be left as No Connect. Connection of these signals to V_{CC}, V_{TTA}, V_{TTD}, V_{DDQ}, V_{SS}, or any other signal (including each other) can result in component malfunction or incompatibility with future processor.

For reliable operation, connect unused input signals to an appropriate signal level. Unused Intel® QuickPath Interconnect input and output pins can be left floating. Unused active high inputs should be connected through a resistor to ground (V_{SS}). Unused outputs can be left unconnected; however, this may interfere with some TAP functions, complicate debug probing, and prevent boundary scan testing. A resistor must be used when tying bidirectional signals to power or ground. When tying any signal to power or ground, including a resistor will also allow for system testability. Resistor values should be within ± 20% of the impedance of the baseboard trace, unless otherwise noted in the appropriate platform design guidelines.

TAP signals do not include on-die termination, however they may include resistors on package (see Section 13.1.9 for details). Inputs and utilized outputs must be terminated on the board. Unused outputs may be terminated on the board or left unconnected. Leaving unused outputs unterminated may interfere with some TAP functions, complicate debug probing, and prevent boundary scan testing. Signal termination requirements are detailed in the *Picket Post: Intel® Xeon® Processor C5500/C3500 Series with the Intel® 3420 Chipset Platform Design Guide (PDG)*.

13.2 Signal Group Summary

Signals are combined in Table 159 by buffer type and characteristics. “Buffer Type” denotes the applicable signaling technology and specifications.

Table 159. Signal Groups (Sheet 1 of 5)

Signal Group	Buffer Type	Signals[1]
PCI Express Signals		
Differential	PCI Express Input	PE_RX_DN[15:0], PE_RX_DP[15:0]
Differential	PCI Express Output	PE_TX_DN[15:0], PE_TX_DP[15:0]
Single ended	Analog Input	PE_ICOMPO, PE_ICOMPI, PE_RCOMPO, PE_RBIAS
Differential	PCI Express Input	PE_CLK_D[P/N]

Table 159. Signal Groups (Sheet 2 of 5)

Signal Group	Buffer Type	Signals[1]
Single ended	CMOS Input/Output	PE_GEN2_DISABLE#, PE_CFG[2:0], PE_NTBXL
Single ended	PCI Express SMB Output	PE_HP_CLK
Single ended	PCI Express SMB Input/Output	PE_HP_DATA
DMI Signals		
Differential	DMI Input	DMI_PE_RX_DN[3:0], DMI_PE_RX_DP[3:0]
Differential	DMI Output	DMI_PE_TX_DN[3:0], DMI_PE_TX_DP[3:0]
Single ended	Analog Input	DMI_COMP
Single ended	CMOS Input	DMI_PE_CFG#
Intel® QuickPath Interconnect Signals		
Differential	Intel® QuickPath Interconnect Input	QPI_RX_DN[19:0], QPI_RX_DP[19:0], QPI_CLKRX_DP, QPI_CLKRX_DN
Differential	Intel® QuickPath Interconnect Output	QPI_TX_DN[19:0], QPI_TX_DP[19:0], QPI_CLKTX_DP, QPI_CLKTX_DN
Single ended	Analog Input	QPI1_COMP[0], QPI1_COMP[1]
DDR Channel A Signals[2]		
Single ended	CMOS Output	DDRA_BA[2:0]
Single ended	CMOS Output	DDRA_CAS#
Single ended	CMOS Output	DDRA_CKE[3:0]
Differential	Output	DDRA_CLK_DN[3:0]
Differential	Output	DDRA_CLK_DP[3:0]
Single ended	CMOS Output	DDRA_CS[7:0]#
Single ended	CMOS Input/Output	DDRA_DQ[63:0]
Differential	CMOS Input/Output	DDRA_DQS_DN[17:0]
Differential	CMOS Input/Output	DDRA_DQS_DP[17:0]
Single ended	CMOS Input/Output	DDRA_ECC[7:0]
Single ended	CMOS Output	DDRA_MA_[15:0]
Single ended	CMOS Output	DDRA_MA_PAR
Single ended	CMOS Output	DDRA_ODT[3:0]
Single ended	Asynchronous Input	DDRA_PAR_ERR[2:0]
Single ended	CMOS Output	DDRA_RAS#
Single ended	Asynchronous Output	DDRA_Reset#
Single ended	CMOS Output	DDRA_WE#
DDR Channel B Signals[2]		
Single ended	CMOS Output	DDRB_BA[2:0]
Single ended	CMOS Output	DDRB_CAS#
Single ended	CMOS Output	DDRB_CKE[3:0]
Differential	Output	DDRB_CLK_DN[3:0]
Differential	Output	DDRB_CLK_DP[3:0]
Single ended	CMOS Output	DDRB_CS[7:0]#
Single ended	CMOS Input/Output	DDRB_DQ[63:0]

Table 159. Signal Groups (Sheet 3 of 5)

Signal Group	Buffer Type	Signals[1]
Differential	CMOS Input/Output	DDRB_DQS_DN[17:0]
Differential	CMOS Input/Output	DDRB_DQS_DP[17:0]
Single ended	CMOS Input/Output	DDRB_ECC[7:0]
Single ended	CMOS Output	DDRB_MA_[15:0]
Single ended	CMOS Output	DDRB_MA_PAR
Single ended	CMOS Output	DDRB_ODT[3:0]
Single ended	Asynchronous Input	DDRB_PAR_ERR[2:0]
Single ended	CMOS Output	DDRB_RAS#
Single ended	Asynchronous Output	DDRB_Reset#
Single ended	CMOS Output	DDRB_WE#
DDR Channel C Signals[2]		
Single ended	CMOS Output	DDRC_BA[2:0]
Single ended	CMOS Output	DDRC_CAS#
Single ended	CMOS Output	DDRC_CKE[3:0]
Differential	Output	DDRC_CLK_DN[3:0]
Differential	Output	DDRC_CLK_DP[3:0]
Single ended	CMOS Output	DDRC_CS[7:0]#
Single ended	CMOS Input/Output	DDRC_DQ[63:0]
Differential	CMOS Input/Output	DDRC_DQS_DN[17:0]
Differential	CMOS Input/Output	DDRC_DQS_DP[17:0]
Single ended	CMOS Input/Output	DDRC_ECC[7:0]
Single ended	CMOS Output	DDRC_MA_[15:0]
Single ended	CMOS Output	DDRC_MA_PAR
Single ended	CMOS Output	DDRC_ODT[3:0]
Single ended	Asynchronous Input	DDRC_PAR_ERR[2:0]
Single ended	CMOS Output	DDRC_RAS#
Single ended	Asynchronous Output	DDRC_Reset#
Single ended	CMOS Output	DDRC_WE#
DDR Compensation Signals[2]		
Single ended	Analog Input	DDR_COMP[2:0]
System Management Bus (SMBus)		
Single ended	SMB Input/Output	SMB_CLK
Single ended	SMB Input/Output	SMB_DATA
Platform Environmental Control Interface (PECI)		
Single ended	Asynchronous Input/Output	PECI
Reset and Miscellaneous Signals		
Single Ended	Asynchronous Input	RSTIN#
Single Ended	CMOS Input/Output	DP_SYNCRST#
Single Ended	Analog	COMP0

Table 159. Signal Groups (Sheet 4 of 5)

Signal Group	Buffer Type	Signals[1]
Single ended	Asynchronous CMOS Input/Output	EXTSYSTRG
Single ended	CMOS Input	PM_SYNC
Single ended	GTL Input/Output	BPM#[7:0]
Single ended	Asynchronous GTL Output	PRDY#
Single ended	Asynchronous GTL Input	PREQ#
Single ended	Asynchronous CMOS Input	DDR_ADR
Thermal Signals		
Single ended	Asynchronous GTL Input/Output	CAT_ERR#
Single ended	Asynchronous Input	PECI_ID#
Single ended	Asynchronous GTL Output	THERMTRIP#
Single ended	Asynchronous GTL Input/Output	PROCHOT#
Single ended	Asynchronous CMOS Output	PSI#
Single ended	Asynchronous Input	DDR_THERM
Single ended	Asynchronous Output	SYS_ERR_STAT[2:0]
Power Sequencing Signals		
Single ended	Asynchronous Output	SKTOCC#
Single ended	Asynchronous Input	DBR#
Single ended	Asynchronous Input	VCCPWRGOOD
Single ended	Asynchronous Input	VTTPWRGOOD
Single ended	Asynchronous Input	DDR_DRAMPWROK
System Reference Clock		
Differential	Input	BCLK_DN
Differential	Input	BCLK_DP
Differential	Output	BCLK_BUF_DN
Differential	Output	BCLK_BUF_DP
Test Access Port (TAP) Signals		
Single ended	GTL Input	TCLK
Single ended	GTL Input	TDI
Single ended	CMOS Input	TDI_M
Single ended	GTL Input	TMS
Single ended	GTL Input	TRST#
Single ended	CMOS Output	TDO
Single ended	CMOS Output	TDO_M
Processor Core Power Signals		
Power / Ground	Analog	V_{CC}
Power / Ground	Analog	V_{CCPLL}
Power / Ground	Analog	V_{DDQ}
Power / Ground	Analog	V_{TTA}
Power / Ground	Analog	V_{TTD}

Table 159. Signal Groups (Sheet 5 of 5)

Signal Group	Buffer Type	Signals[1]
Power / Ground	Analog	V_{SS}
	Analog Input	ISENSE
	Analog	VCCSENSE
	Analog	VSSSENSE
	Analog	VSS_SENSE_VTTD
	Analog	VTTD_SENSE
Single ended	CMOS (Push-Pull) Output (CMOS Input during power up for POC straps)	VID[7:6], VID[5:3]/CSC[2:0], VID[2:0]/MSID[2:0]
Single ended	CMOS (Push-Pull) Output	VTT_VID[4:2]
No Connect & Reserved Signals		
		NC_x
		RSV_x

1. See Section 6.0, "System Address Map" for signal definitions.
2. DDR3A refers to DDR3 Channel A, DDR3B refers to ChannelB, and DDR3C refers to Channel C.

Signals that include on-die termination (ODT) are listed in Table 160.

Table 160. Signals With On-Die Termination (ODT)

Intel® QuickPath Interface Signal Group[1]
QPI_RX_DP[19:0], QPI_RX_DN[19:0], QPI_TX_DP[19:0], QPI_TX_DN[19:0], QPI_CLKRX_DN, QPI_CLKRX_DP, QPI_CLKTX_DN, QPI_CLKTX_DP
PCI Express Signals
PE_RX_DN[15:0], PE_RX_DP[15:0], PE_TX_DN[15:0], PE_TX_DP[15:0]
DDR3 Signal Group[2]
DDRA_DQ[63:0], DDRB_DQ[63:0], DDDRC_DQ[63:0], DDRA_DQS_N[17:0], DDRA_DQS_P[17:0], DDRB_DQS_N[17:0], DDRB_DQS_P[17:0], DDRC_DQS_N[17:0], DDRC_DQS_P[17:0], DDRA_ECC[7:0], DDRB_ECC[7:0], DDRC_ECC[7:0], DDRA_PAR_ERR#[2:0], DDRB_PAR_ERR#[2:0], DDRC_PAR_ERR#[2:0][3]
Reset and Miscellanous Signal Group and Thermal Signal Group[1]
BPM#[7:0][6], PECI_ID#[7], PREQ#[6], DP_SYNCRST#[9], EXTSYSTRG[9], PMSYNC[9], DDR_ADR[9]
Test Access Port (TAP) Signal Group
TCK[4], TDI[5], TMS[5], TRST#[5], TDI_M[9]
Power/Other Signal Group[8]
VCCPWRGOOD, VTTPWRGOOD, DDR_DRAMPWROK

Notes:
1. Unless otherwise specified, signals have ODT in the package with a 50 Ohm pull-down to V_{SS}.
2. Unless otherwise specified, all DDR3 signals are terminated to $V_{DDQ}/2$.
3. DDRA_PAR_ERR#[2:0], DDRB_PAR_ERR#[2:0], and DDRC_PAR_ERR#[2:0] are terminated to V_{DDQ}.
4. TCK does not include ODT, this signal is weakly pulled-down via a 1-5 kOhm resistor to V_{SS}.
5. TDI, TMS, TRST# do not include ODT, these signals are weakly pulled-up via ~ 10 kOhm resistor to V_{TT}.
6. BPM[7:0]# and PREQ# signals have ODT in package with 35 Ohm pull-ups to V_{TT}.
7. PECI_ID# has ODT in package with a 1-5 kOhm pull-up to V_{TT}.
8. VCCPWRGOOD, VTTPWRGOOD, and DDR_DRAMPWROK have ODT in package with a 5-20 kOhm pull-down to V_{SS}.
9. DP_SYNCRST, EXTSYSTRG, PMSYNC, and TDI_M have a 50 ohm ODT to Vtt.

13.3 Mixing Processors

Intel supports and validates dual processor configurations only in which both processors operate with the same Intel® QuickPath Interconnect frequency, core frequency, power segment, and have the same internal cache sizes. Mixing components operating at different internal clock frequencies is not supported and will not be validated by Intel. Combining processors from different power segments is also not supported.

Note: Processors within a system must operate at the same frequency per bits [15:8] of the FLEX_RATIO MSR (Address: 194h); however this does not apply to frequency transitions initiated due to thermal events, Extended HALT, Enhanced Intel SpeedStep® Technology transitions signal (See Section 8.0, "Power Management").

Not all operating systems can support dual processors with mixed frequencies. Mixing processors of different steppings but the same model (as per CPUID instruction) is supported. Details regarding the CPUID instruction are provided in the *AP-485, Intel® Processor Identification and the CPUID Instruction* application note. See also the *Intel® Xeon® Processor C5500/C3500 Series Specification Update* for details.

13.4 Flexible Motherboard Guidelines (FMB)

The Flexible Motherboard (FMB) guidelines are estimates of the maximum values the Intel® Xeon® processor C5500/C3500 series will have over certain time periods. The values are only estimates and actual specifications for future processors may differ. Processors may or may not have specifications equal to the FMB value in the foreseeable future. System designers should meet the FMB values to ensure their systems will be compatible with future Bloomfield processor.

13.5 Absolute Maximum and Minimum Ratings

Note: **All specifications are pre-silicon estimates and are subject to change.**

Table 161 specifies absolute maximum and minimum ratings which lie outside the functional limits of the processor. Only within specified operation limits, can functionality and long-term reliability be expected.

At conditions outside functional operation condition limits, but within absolute maximum and minimum ratings, neither functionality nor long-term reliability can be expected. If a device is returned to conditions within functional operation limits after having been subjected to conditions outside these limits, but within the absolute maximum and minimum ratings, the device may be functional, but with its lifetime degraded depending on exposure to conditions exceeding the functional operation condition limits.

At conditions exceeding absolute maximum and minimum ratings, neither functionality nor long-term reliability can be expected. Moreover, if a device is subjected to these conditions for any length of time then, when returned to conditions within the functional operating condition limits, it will either not function or its reliability will be severely degraded.

Although the processor contains protective circuitry to resist damage from static electric discharge, precautions should always be taken to avoid high static voltages or electric fields.

Table 161. Processor Absolute Minimum and Maximum Ratings

Symbol	Parameter	Min	Nominal	Max	Unit	Notes[1,2]
V_{CC}	Processor core voltage with respect to V_{SS}	-0.300		1.350	V	
V_{CCPLL}	Processor PLL voltage with respect to V_{SS}		1.800		V	4
V_{DDQ}	Processor I/O supply voltage for DDR3 with respect to V_{SS}		1.500		V	4
V_{TTA}	Processor uncore analog voltage with respect to V_{SS}			1.155	V	3
V_{TTD}	Processor uncore digital voltage with respect to V_{SS}			1.155	V	3
T_{CASE}	Processor case temperature				°C	8
$T_{STORAGE}$	Storage temperature	-40		85	°C	5,6,7

Notes:
1. For functional operation, all processor electrical, signal quality, mechanical and thermal specifications must be satisfied.
2. Excessive overshoot or undershoot on any signal will likely result in permanent damage to the processor.
3. V_{TTA} and V_{TTD} should be derived from the same voltage regulator (VR).
4. ±5% tolerance.
5. Storage temperature is applicable to storage conditions only. In this scenario, the processor must not receive a clock, and no lands can be connected to a voltage bias. Storage within these limits will not affect the long-term reliability of the device. For functional operation, see the processor case temperature specifications.
6. This rating applies to the processor and does not include any tray or packaging.
7. Failure to adhere to this specification can affect the long-term reliability of the processor.
8. See the *Intel*® Xeon® Processor C5500/C3500 Series *Thermal / Mechanical Design Guide*.

13.6 Processor DC Specifications

Note: **All electrical specifications are pre-silicon estimates and are subject to change.**

Note: DC specifications are defined at the processor pads, unless otherwise noted.

Note: For available SKUs, see Table 1, “Available SKUs”.

DC specifications are only valid while meeting specifications for case temperature (T_{CASE} specified in *Intel*® Xeon® Processor C5500/C3500 Series *Thermal / Mechanical Design Guide*), clock frequency, and input voltages. Care should be taken to read all notes associated with each specification.

This section is divided into groups of signals that have similar characteristics and buffer types. There are eight sections in total, divided as follows:

- Voltage and Current
- DDR3 Channel A, B, & C
- PCI Express
- DMI Interface
- SMBUS Interface
- PECI Interface
- Clock
- Reset and Miscellaneous, Thermal, Power Sequencing, and TAP

Table 162. Voltage and Current Specifications (Sheet 1 of 3)

Symbol	Parameter	Voltage Plane	Min	Typ	Max	Unit	Notes[1]
VID	V_{CC} VID Range	-	0.750		1.350	V	2,3
V_{CC}	Core Voltage (Launch - FMB)	V_{CC}	See Table 163 and Figure 86			V	3,4,6,7, 11
V_{VID_STEP}	VID step size during a transition	-			± 6.250	mV	9
V_{CCPLL}	PLL Voltage (DC + AC specification)	V_{CCPLL}	0.95*V_{CCPLL} (Typ)	1.800	1.05*V_{CCPLL} (Typ)	V	10
V_{DDQ}	I/O Voltage for DDR3 (DC + AC specification)	V_{DDQ}	0.95*V_{DDQ} (Typ)	1.500	1.05*V_{DDQ} (Typ)	V	10
VTT_VID	V_{TT} VID Range	-	1.045		1.220	V	2,3
V_{TT}	Uncore Voltage (Launch - FMB)	V_{TT}	See Table 173			V	3,5,8,11

Notes:

1. Unless otherwise noted, all specifications in this table apply to all processors. These specifications are based on pre-silicon characterization and will be updated as further data becomes available.
2. Individual processor VID and/or VTT_VID values may be calibrated during manufacturing such that two devices at the same speed may have different settings.
3. These voltages are targets only. A variable voltage source should exist on systems in the event that a different voltage is required.
4. The V_{CC} voltage specification requirements are measured across vias on the platform for VCCSENSE and VSSSENSE pins close to the socket with a 100 MHz bandwidth oscilloscope, 1.5 pF maximum probe capacitance, and 1 MΩ minimum impedance. The maximum length of ground wire on the probe should be less than 5 mm. Ensure external noise from the system is not coupled in the scope probe.
5. The V_{TT} voltage specification requirements are measured across platform vias for VTTD_SENSE and VSS_SENSE_VTTD lands close to the socket with a 100 MHz bandwidth oscilloscope, 1.5 pF maximum probe capacitance, and 1 MΩ minimum impedance. The maximum length of ground wire on the probe should be less than 5 mm. Ensure external noise from the system is not coupled in the scope probe.
6. See Table 163 and corresponding Figure 86. The processor should not be subjected to any static V_{CC} level that exceeds the V_{CC_MAX} associated with any particular current. Failure to adhere to this specification can shorten processor lifetime.
7. Minimum V_{CC} and maximum I_{CC} are specified at the maximum processor case temperature (T_{CASE}) shown in the *Intel® Xeon® Processor C5500/C3500 Series Thermal / Mechanical Design Guide*. I_{CC_MAX} is specified at the relative V_{CC_MAX} point on the V_{CC} load line. The processor is capable of drawing I_{CC_MAX} for up to 10 ms.
8. See Table 173. Do not subject processor to any static V_{TT} level exceeding V_{TT_MAX} associated with any particular current. Failure to adhere to this specification can shorten processor lifetime.
9. This specification represents the V_{CC} reduction due to each VID transition. See Section 13.1.10.3. .
10. Baseboard bandwidth is limited to 20 MHz.
11. FMB is the flexible motherboard guidelines. See Section 13.4 for FMB details.

Table 162. Voltage and Current Specifications (Sheet 2 of 3)

Symbol	Parameter	Voltage Plane	Min	Typ	Max	Unit	Notes[1]
I_{CC_MAX} I_{CCPLL_MAX} I_{DDQ_MAX} I_{TT_MAX}	ECC5549: TDP = 85W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			100 1.5 9 6 22	A	11
	ECC5509: TDP = 85W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			75 1.5 7.5 6 27	A	11
	ECC3539: TDP = 65W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			55 1.5 7.5 4 20	A	11
	LC5528: TDP = 60W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			70 1.5 7.5 6 17	A	11
	EC5539: TDP = 65W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			42 1.5 9 6 27	A	11
I_{CC_MAX} I_{CCPLL_MAX} I_{DDQ_MAX} I_{TT_MAX}	LC5518: TDP = 48W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			54 1.5 7.5 6 17	A	11

Notes:

1. Unless otherwise noted, all specifications in this table apply to all processors. These specifications are based on pre-silicon characterization and will be updated as further data becomes available.
2. Individual processor VID and/or VTT_VID values may be calibrated during manufacturing such that two devices at the same speed may have different settings.
3. These voltages are targets only. A variable voltage source should exist on systems in the event that a different voltage is required.
4. The V_{CC} voltage specification requirements are measured across vias on the platform for VCCSENSE and VSSSENSE pins close to the socket with a 100 MHz bandwidth oscilloscope, 1.5 pF maximum probe capacitance, and 1 MΩ minimum impedance. The maximum length of ground wire on the probe should be less than 5 mm. Ensure external noise from the system is not coupled in the scope probe.
5. The V_{TT} voltage specification requirements are measured across platform vias for VTTD_SENSE and VSS_SENSE_VTTD lands close to the socket with a 100 MHz bandwidth oscilloscope, 1.5 pF maximum probe capacitance, and 1 MΩ minimum impedance. The maximum length of ground wire on the probe should be less than 5 mm. Ensure external noise from the system is not coupled in the scope probe.
6. See Table 163 and corresponding Figure 86. The processor should not be subjected to any static V_{CC} level that exceeds the V_{CC_MAX} associated with any particular current. Failure to adhere to this specification can shorten processor lifetime.
7. Minimum V_{CC} and maximum I_{CC} are specified at the maximum processor case temperature (T_{CASE}) shown in the *Intel® Xeon® Processor C5500/C3500 Series Thermal / Mechanical Design Guide*. I_{CC_MAX} is specified at the relative V_{CC_MAX} point on the V_{CC} load line. The processor is capable of drawing I_{CC_MAX} for up to 10 ms.
8. See Table 173. Do not subject processor to any static V_{TT} level exceeding V_{TT_MAX} associated with any particular current. Failure to adhere to this specification can shorten processor lifetime.
9. This specification represents the V_{CC} reduction due to each VID transition. See Section 13.1.10.3. .
10. Baseboard bandwidth is limited to 20 MHz.
11. FMB is the flexible motherboard guidelines. See Section 13.4 for FMB details.

Table 162. Voltage and Current Specifications (Sheet 3 of 3)

Symbol	Parameter	Voltage Plane	Min	Typ	Max	Unit	Notes[1]
I_{CC_MAX} I_{CCPLL_MAX} I_{DDQ_MAX} I_{TT_MAX}							
	P1053: TDP = 50W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			13 1.5 4.5 4.2 22	A	11
	LC3528: TDP = 32W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			24 1.5 5.5 4 15	A	11
	LC3518: TDP = 23W	V_{CC} V_{CCPLL} V_{DDQ} V_{TTA} V_{TTD}			11 1.5 4.5 4 13	A	11
I_{DDQ_S3}	DDR3 System Memory Interface Supply Current in Standby State	V_{DDQ}				A	

Notes:

1. Unless otherwise noted, all specifications in this table apply to all processors. These specifications are based on pre-silicon characterization and will be updated as further data becomes available.
2. Individual processor VID and/or VTT_VID values may be calibrated during manufacturing such that two devices at the same speed may have different settings.
3. These voltages are targets only. A variable voltage source should exist on systems in the event that a different voltage is required.
4. The V_{CC} voltage specification requirements are measured across vias on the platform for VCCSENSE and VSSSENSE pins close to the socket with a 100 MHz bandwidth oscilloscope, 1.5 pF maximum probe capacitance, and 1 MΩ minimum impedance. The maximum length of ground wire on the probe should be less than 5 mm. Ensure external noise from the system is not coupled in the scope probe.
5. The V_{TT} voltage specification requirements are measured across platform vias for VTTD_SENSE and VSS_SENSE_VTTD lands close to the socket with a 100 MHz bandwidth oscilloscope, 1.5 pF maximum probe capacitance, and 1 MΩ minimum impedance. The maximum length of ground wire on the probe should be less than 5 mm. Ensure external noise from the system is not coupled in the scope probe.
6. See Table 163 and corresponding Figure 86. The processor should not be subjected to any static V_{CC} level that exceeds the V_{CC_MAX} associated with any particular current. Failure to adhere to this specification can shorten processor lifetime.
7. Minimum V_{CC} and maximum I_{CC} are specified at the maximum processor case temperature (T_{CASE}) shown in the *Intel® Xeon® Processor C5500/C3500 Series Thermal / Mechanical Design Guide*. I_{CC_MAX} is specified at the relative V_{CC_MAX} point on the V_{CC} load line. The processor is capable of drawing I_{CC_MAX} for up to 10 ms.
8. See Table 173. Do not subject processor to any static V_{TT} level exceeding V_{TT_MAX} associated with any particular current. Failure to adhere to this specification can shorten processor lifetime.
9. This specification represents the V_{CC} reduction due to each VID transition. See Section 13.1.10.3. .
10. Baseboard bandwidth is limited to 20 MHz.
11. FMB is the flexible motherboard guidelines. See Section 13.4 for FMB details.

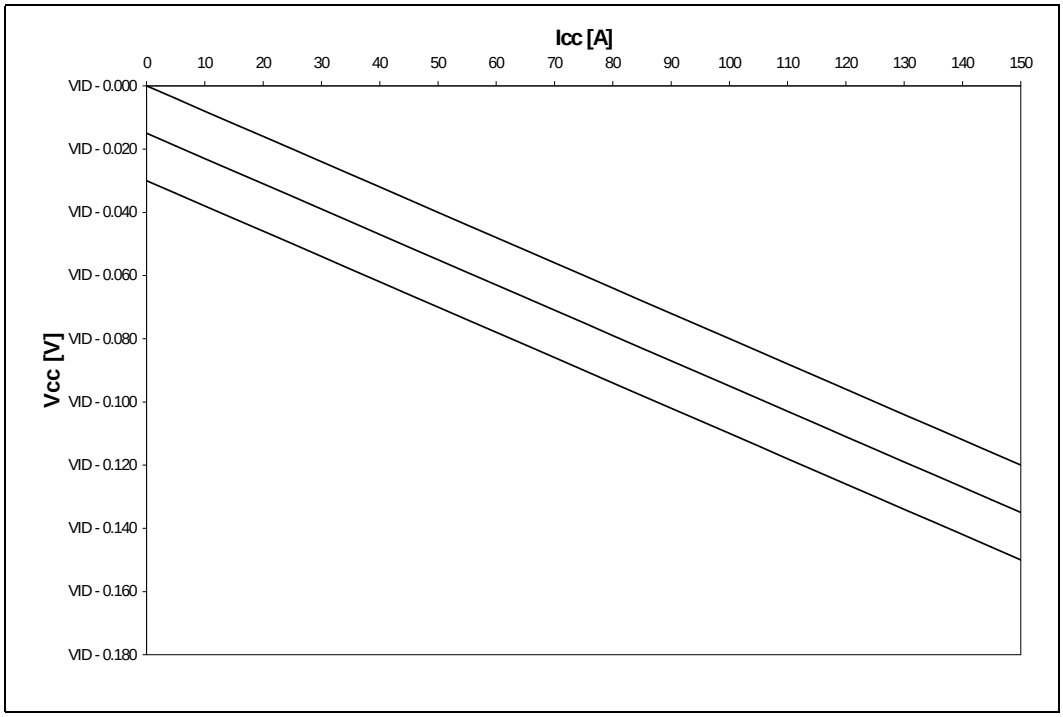
Table 163. V_{CC} Static and Transient Tolerance

I_{CC} (A)	V_{CC_MAX} (V)	V_{CC_TYP} (V)	V_{CC_MIN} (V)	Notes[1,2,3,4]
0	VID - 0.000	VID - 0.015	VID - 0.030	
5	VID - 0.004	VID - 0.019	VID - 0.034	
10	VID - 0.008	VID - 0.023	VID - 0.038	
15	VID - 0.012	VID - 0.027	VID - 0.042	
20	VID - 0.016	VID - 0.031	VID - 0.046	
25	VID - 0.020	VID - 0.035	VID - 0.050	
30	VID - 0.024	VID - 0.039	VID - 0.054	
35	VID - 0.028	VID - 0.043	VID - 0.058	
40	VID - 0.032	VID - 0.047	VID - 0.062	
45	VID - 0.036	VID - 0.051	VID - 0.066	
50	VID - 0.040	VID - 0.055	VID - 0.070	
55	VID - 0.044	VID - 0.059	VID - 0.074	
60	VID - 0.048	VID - 0.063	VID - 0.078	
65	VID - 0.052	VID - 0.067	VID - 0.082	
70	VID - 0.056	VID - 0.071	VID - 0.086	
75	VID - 0.060	VID - 0.075	VID - 0.090	
80	VID - 0.064	VID - 0.079	VID - 0.094	
85	VID - 0.068	VID - 0.083	VID - 0.098	
90	VID - 0.072	VID - 0.087	VID - 0.102	
95	VID - 0.076	VID - 0.091	VID - 0.106	
100	VID - 0.080	VID - 0.095	VID - 0.110	
105	VID - 0.084	VID - 0.099	VID - 0.114	
110	VID - 0.088	VID - 0.103	VID - 0.118	
115	VID - 0.092	VID - 0.107	VID - 0.122	
120	VID - 0.096	VID - 0.111	VID - 0.126	
125	VID - 0.100	VID - 0.115	VID - 0.130	
130	VID - 0.104	VID - 0.119	VID - 0.134	
135	VID - 0.108	VID - 0.123	VID - 0.138	
140	VID - 0.112	VID - 0.127	VID - 0.142	
145	VID - 0.116	VID - 0.131	VID - 0.146	
150	VID - 0.120	VID - 0.135	VID - 0.150	

Notes:

1. The V_{CC_MIN} and V_{CC_MAX} loadlines represent static and transient limits. See Section 13.6.1 for V_{CC} overshoot specifications.
2. This table is intended to aid in reading discrete points on Figure 86.
3. The loadlines specify voltage limits at the die measured at the VCC_SENSE and VSS_SENSE lands. Voltage regulation feedback for voltage regulator circuits must also be taken from processor VCC_SENSE and VSS_SENSE lands. See the *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator Down (EVRD) 11.1 Design Guidelines*, Revision 1.5 for socket loadline guidelines and regulator implementation. See the appropriate platform design guide for further details on regulator and decoupling implementations.
4. Processor core current (I_{CC}) ranges are valid up to I_{CC_MAX} of the processor SKU as defined in Table 162, “Voltage and Current Specifications”.

Figure 86. V_{CC} Static and Transient Tolerance Loadlines[1,2,3,4]



Notes:

1. The V_{CC_MIN} and V_{CC_MAX} loadlines represent static and transient limits. See Section 13.6.1 for V_{CC} overshoot specifications.
2. See Table 163 for V_{CC} Static and Transient Tolerance.
3. The loadlines specify voltage limits at the die measured at the VCC_SENSE and VSS_SENSE lands. Voltage regulation feedback for voltage regulator circuits must also be taken from processor VCC_SENSE and VSS_SENSE lands. See the *Voltage Regulator Module (VRM) and Enterprise Voltage Regulator Down (EVRD) 11.1 Design Guidelines*, Revision 1.5 for socket loadline guidelines and regulator implementation. See the appropriate platform design guide for further details on regulator and decoupling implementations.
4. Processor core current (I_{CC}) ranges are valid up to I_{CC_MAX} of the processor SKU as defined in Table 162, "Voltage and Current Specifications".

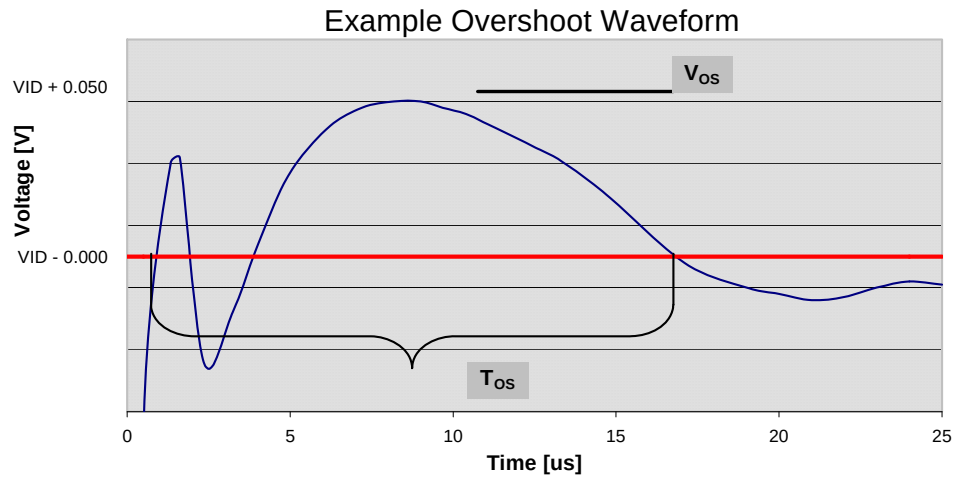
13.6.1 V_{CC} Overshoot Specifications

The processor can tolerate short transient overshoot events where V_{CC} exceeds the VID voltage when transitioning from a high-to-low current load condition. This overshoot cannot exceed VID + V_{OS_MAX} (V_{OS_MAX} is the maximum allowable overshoot above VID). These specifications apply to the processor die voltage as measured across the VCC_SENSE and VSS_SENSE lands.

Table 164. V_{CC} Overshoot Specifications

Symbol	Parameter	Min	Max	Units	Figure	Notes
V_{OS_MAX}	Magnitude of V_{CC} overshoot above VID	-	50	mV	87	
T_{OS_MAX}	Time duration of V_{CC} overshoot above VID	-	25	µs	87	

Figure 87. V_{CC} Overshoot Example Waveform



Notes:
1. V_{OS} is the measured overshoot voltage.
2. T_{OS} is the measured time duration above VID.

13.6.2 Die Voltage Validation

Core voltage (V_{CC}) overshoot events at the processor must meet the specifications in Table 164 when measured across the VCC_SENSE and VSS_SENSE lands. Overshoot events that are < 10 ns in duration may be ignored. These measurements of processor die level overshoot should be taken with a 100 MHz bandwidth limited oscilloscope.

Table 165. ICC Max and ICC TDP by SKU

SKU	ICC_{MAX}	ICC_{TDC}
ECC5549	100	70
ECC5509	75	70
ECC3539	55	50
LC5528	70	50
EC5539	42	40
LC5518	54	39
P1053	13	12
LC3528	24	20
LC3518	11	10

13.6.3 DDR3 Signal DC Specifications

The following table defines the parameters for DDR3.

Table 166. DDR3 Signal Group DC Specifications (Sheet 1 of 2)

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage			$0.43*V_{DDQ}$	V	2,
V_{IH}	Input High Voltage	$0.57*V_{DDQ}$			V	3, 4
V_{OL}	Output Low Voltage		$(V_{DDQ} / 2)* (R_{ON} / (R_{ON}+R_{VTT_TERM}))$		V	6
V_{OH}	Output High Voltage		$V_{DDQ} - ((V_{DDQ} / 2)* (R_{ON}/(R_{ON}+R_{VTT_TERM}))$		V	4,6
R_{ON}	DDR3 Clock Buffer On Resistance	21		31	Ohms	5
R_{ON}	DDR3 Command Buffer On Resistance	16		24	Ohms	5
R_{ON}	DDR3 Reset Buffer On Resistance	25		75	Ohms	5
R_{ON}	DDR3 Control Buffer On Resistance	21		31	Ohms	5

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. V_{IL} is the maximum voltage level at a receiving agent that will be interpreted as a logical low value.
3. V_{IH} is the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
4. V_{IH} and V_{OH} may experience excursions above V_{DDQ}. However, input signal drivers must comply with the signal quality specifications.
5. This is the pull down driver resistance. See the processor signal integrity models for I/V characteristics.
6. R_{VTT_TERM} is the termination on the DIMM and not controlled by the Intel® Xeon® processor C5500/C3500 series. See the applicable DIMM datasheet.
7. The minimum and maximum values for these signals are programmable by BIOS to one of the pairs.
8. COMP resistance must be provided on the system board with 1% resistors. See the *Picket Post: Intel® Xeon® Processor C5500/C3500 Series with the Intel® 3420 Chipset Platform Design Guide (PDG)* for implementation details. DDR_COMP[2:0] resistors are to Vss.

Table 166. DDR3 Signal Group DC Specifications (Sheet 2 of 2)

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
R_{ON}	DDR3 Data Buffer On Resistance	21		31	Ohms	5
Data ODT	On-Die Termination for Data Signals	45 90		55 110	Ohms	7
ParErr ODT	On-Die Termination for Parity Error bits	60		80	Ohms	
I_{LI}	Input Leakage Current			± 500	mA	
DDR_COMP0	COMP Resistance	99	100	101	Ohms	8
DDR_COMP1	COMP Resistance	24.65	24.9	25.15	Ohms	8
DDR_COMP2	COMP Resistance	128.7	130	131.3	Ohms	8

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. V_{IL} is the maximum voltage level at a receiving agent that will be interpreted as a logical low value.
3. V_{IH} is the minimum voltage level at a receiving agent that will be interpreted as a logical high value.
4. V_{IH} and V_{OH} may experience excursions above V_{DDQ}. However, input signal drivers must comply with the signal quality specifications.
5. This is the pull down driver resistance. See the processor signal integrity models for I/V characteristics.
6. R_{VTT_TERM} is the termination on the DIMM and not controlled by the Intel® Xeon® processor C5500/C3500 series. See the applicable DIMM datasheet.
7. The minimum and maximum values for these signals are programmable by BIOS to one of the pairs.
8. COMP resistance must be provided on the system board with 1% resistors. See the *Picket Post: Intel® Xeon® Processor C5500/C3500 Series with the Intel® 3420 Chipset Platform Design Guide (PDG)* for implementation details. DDR_COMP[2:0] resistors are to Vss.

13.6.4 PCI Express Signal DC Specifications

The following table defines the parameters for transmitters and receivers.

Table 167. PCI Express/DMI Interface -- 2.5 and 5.0 GT/s Transmitter DC Specifications

Symbol	Parameter	2.5 GT/s	5.0 GT/s	Units	Comments
$Z_{TX-DIFF-DC}$	DC differential Tx impedance	80 (min) 120 (max)	120 (max)	Ohms	Low impedance defined during signaling. Parameter is captured for 5.0 GHz by $RL_{TX-DIFF}$.
$I_{TX-SHORT}$	Transmitter short-circuit current limit	90 (max)	90 (max)	mA	The total current the Transmitter can supply when shorted to ground.
$V_{TX-DC-CM}$	Transmitter DC common-mode voltage	0 (min) 3.6 (max)	0 (min) 3.6 (max)	V	The allowed DC common-mode voltage at the Transmitter pins under any conditions.
$V_{TX-CM-DC-ACTIVEIDLE-DELTA}$	Absolute Delta of DC Common Mode Voltage during L0 and Electrical Idle	0 (min) 100 (max)	0 (min) 100 (max)	mV	|VTX-CM-DC [during L0] – VTX-CM-Idle-DC [during Electrical Idle]| <= 100 mV VTX-CM-DC = DC(avg) of |VTX-D+ + VTX-D-|/2 VTX-CM-Idle-DC= DC(avg) of |VTX-D+ + VTX-D-|/2 [Electrical Idle]
$V_{TX-CM-DC-LINEDELTA}$	Absolute Delta of DC Common Mode Voltage between D+ and D	0 (min) 25 (max)	0 (min) 25 (max)	mV	$\|V_{TX-CM-DC-D+ [during L0] – } V_{TX-CM-DC-D-[during L0.]}\|$ 25 mV $V_{TX-CM-DC-D+}$ = DC(avg) of |VTX-D+| [during L0] VTX-CM-DC-D- = DC(avg) of |VTX-D-| [during L0]
$V_{TX-IDLE-DIFF-DC}$	DC Electrical Idle Differential Output Voltage	Not specified	0 (min) 5 (max)	mV	VTX-IDLE-DIFF-DC = |VTX-Idle-D+ - VTx-Idle-D-| 5 mV. Voltage must be low pass filtered to remove any AC component. Filter characteristics complementary to above.

Table 168. PCI Express Interface -- 2.5 and 5.0 GT/s Recevier DC Specifications

Symbol	Parameter	2.5 GT/s	5.0 GT/s	Units	Comments
ZRX-DC	Receiver DC common mode impedance	40 (min) 60 (max)	40 (min) 60 (max)	Ohms	DC impedance limits are needed to guarantee Receiver detect.
ZRX-DIFF-DC	DC differential impedance	80 (min) 120 (max)	Not specified	Ohms	For 5.0 GT/s covered under $RL_{RX-DIFF}$ parameter.
ZRX-HIGH-IMP-DCPOS	DC Input CM Input Impedance for V>0 during Reset or power down	50 k (min)	50 k (min)	Ohms	Rx DC CM impedance with the Rx terminations not powered, measured over the range 0 - 200 mV with respect to ground.
ZRX-HIGH-IMP-DCNEG	DC Input CM Input Impedance for V<0 during Reset or power down	1.0 k (min)	1.0 k (min)	Ohms	Rx DC CM impedance with the Rx terminations not powered, measured over the range -150 - 0 mV with respect to ground.
VRX-IDLE-DET-DIFFp-p	Electrical Idle Detect Threshold	65 (min) 175 (max)	65 (min) 175 (max)	mV	$V_{RX-IDLE-DET-DIFFp-p} = 2*\|V_{RX-D+} - V_{RXD-}\|$. Measured at the package pins of the Receiver.

13.6.5 SMBus Signal DC Specifications

The following tables define the parameters for SMBus.

Table 169. SMBus Clock DC Electrical Limits

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage			$0.64 * V_{TTA}$	V	2,3
V_{IH}	Input High Voltage	$0.76 * V_{TTA}$				2
V_{OL}	Output Low Voltage			$V_{TTA} * R_{ON} / (R_{ON} + R_{SYS_TERM})$	V	2,4
V_{OH}	Output High Voltage	V_{TTA}			V	2
R_{ON}	IO Buffer On Resistance	10		18	Ohms	
I_{LI}	Input Leakage Current			± 200	µA	

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. The V_{TTA} referred to in these specifications refers to instantaneous V_{TTA}.
3. Based on a test load of 50 Ohms to V_{TTA}.
4. R_{SYS_TERM} is the termination on the system and is not controlled by the Intel® Xeon® processor C5500/C3500 series.

13.6.6 PECI Signal DC Specifications

The following table defines the parameters for PECI.

Table 170. PECI DC Electrical Limits

Symbol	Definition and Conditions	Min	Max	Units	Notes[1]
V_{In}	Input Voltage Range	-0.150	V_{TTD} + 0.150	V	
$V_{Hysteresis}$	Hysteresis	0.100 * V_{TTD}		V	
V_N	Negative-edge threshold voltage	0.275 * V_{TTD}	0.500 * V_{TTD}	V	2,6
V_P	Positive-edge threshold voltage	0.550 * V_{TTD}	0.725 * V_{TTD}	V	2,6
R_{Pullup}	Pullup Resistance (V_{OH} = 0.75 * V_{TTD})		50	Ohms	
I_{Leak+}	High impedance state leakage to V_{TTD} (V_{leak} = V_{OL})		50	µA	3
I_{Leak-}	High impedance leakage to GND (V_{leak} = V_{OH})		25	µA	3
C_{Bus}	Bus capacitance per node		10	pF	4,5
V_{Noise}	Signal noise immunity above 300 MHz	0.100 * V_{TTD}		V_{p-p}	

Notes:
1. V_{TTD} supplies the PECI interface. PECI behavior does not affect V_{TTD} min/max specifications.
2. It is expected that the PECI driver will take into account, the variance in the receiver input thresholds and consequently, be able to drive its output within safe limits (-0.150 V to 0.275*V_{TTD} for the low level and 0.725*V_{TTD} to V_{TTD}+0.150 for the high level).
3. The leakage specification applies to powered devices on the PECI bus.
4. One node is counted for each client and one node for the system host. Extended trace lengths might appear as additional nodes.
5. Excessive capacitive loading on the PECI line may slow down the signal rise/fall times and consequently limit the maximum bit rate at which the interface can operate.
6. See Figure 84 for further information.

13.6.7 System Reference Clock Signal DC Specifications

The following table defines the parameters for System Reference Clock.

Table 171. System Reference Clock DC Specifications

Symbol	Parameter	Min	Max	Unit	Notes[1]
$V_{Refclk_diff_ih}$	Differential Input High Voltage	0.150		V	
$V_{Refclk_diff_il}$	Differential Input Low Voltage		-0.150	V	
V_{cross} (abs)	Absolute Crossing Point	0.250	0.550	V	2, 4
V_{cross}(rel)	Relative Crossing Point	0.250 + 0.5*(VH_{avg} - 0.700)	0.550 + 0.5*(VH_{avg} - 0.700)	V	3,4,5
ΔV_{cross}	Range of Crossing Points		0.140	V	6

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. Crossing Voltage is defined as the instantaneous voltage value when the rising edge of BCLK0 is equal to the falling edge of BCLK1.
3. V_{Havg} is the statistical average of the VH measured by the oscilloscope.
4. The crossing point must meet the absolute and relative crossing point specifications simultaneously.
5. V_{Havg} can be measured directly using “Vtop” on Agilent* and “High” on Tektronix oscilloscopes.
6. V_{CROSS} is defined as the total variation of all crossing voltages as defined in Note 2.

13.6.8 Reset and Micscellaneous DC Specifications

Table 172. Reset and Miscellaneous Signal Group DC Specifications

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage			0.64 * V_{TTA}	V	2,3
V_{IH}	Input High Voltage	0.76 * V_{TTA}			V	2
V_{OL}	Output Low Voltage			V_{TTA} * R_{ON} / (R_{ON} + R_{SYS_TERM})	V	2,4
V_{OH}	Output High Voltage	V_{TTA}			V	2
ODT	On-Die Termination	45		55		5
R_{ON}	Buffer On Resistance	10		18	Ohms	

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. The V_{TTA} referred to in these specifications refers to instantaneous V_{TTA}.
3. Based on a test load of 50 Ohms to V_{TTA}.
4. R_{SYS_TERM} is the termination on the system and is not controlled by the Intel® Xeon® processor C5500/C3500 series.
5. Applies to all signals, unless otherwise mentioned in Table 159.

13.6.9 Thermal DC Specification

Table 173. Thermal Signal Group DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage			0.64 * V_{TTA}	V	2,3
V_{IH}	Input High Voltage	0.76 * V_{TTA}			V	2
V_{IL}	Input Low Voltage (PECI_ID#)			0.22	V	7
V_{IH}	Input High Voltage (PECI_ID#)	0.6			V	7
V_{OL}	Output Low Voltage			V_{TTA} * R_{ON} / (R_{ON} + R_{SYS_TERM})	V	2,4
V_{OH}	Output High Voltage	V_{TTA}			V	2
ODT	On-Die Termination	45		55		5
R_{ON}	Buffer On Resistance	10		18	Ohms	
I_{LI}	Input Leakage Current			± 200	µA	6

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. The V_{TTA} referred to in these specifications refers to instantaneous V_{TTA}.
3. Based on a test load of 50 Ohms to V_{TTA}.
4. R_{SYS_TERM} is the termination on the system and is not controlled by the Intel® Xeon® processor C5500/C3500 series.
5. Applies to all Thermal signals, unless otherwise mentioned in Table 159.
6. Applies to PROCHOT# signal only. See Section 13.1.10.3.2 and Section 8.1 for information regarding Power-On Configuration options.
7. This specification only applies to PECI_ID#. This specification includes hysteresis values.

13.6.10 Test Access Port (TAP) DC Specification

Table 174. Test Access Port (TAP) Signal Group DC Specification

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage			0.40 * V_{TTA}	V	2,3
V_{IH}	Input High Voltage	0.60 * V_{TTA}			V	2
V_{OL}	Output Low Voltage			V_{TTA} * R_{ON} / (R_{ON} + R_{SYS_TERM})	V	2,4
V_{OH}	Output High Voltage	V_{TTA}			V	2
ODT	On-Die Termination	45		55		5
R_{ON}	Buffer On Resistance	10		18	Ohms	

Notes:
1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. The V_{TTA} referred to in these specifications refers to instantaneous V_{TTA}.
3. Based on a test load of 50 Ohms to V_{TTA}.
4. R_{SYS_TERM} is the termination on the system and is not controlled by the Intel® Xeon® processor C5500/C3500 series.
5. Applies to all TAP signals, unless otherwise mentioned in Table 159.

13.6.11 Power Sequencing Signal DC Specification

Table 175. Power Sequencing Signal Group DC Specifications

Symbol	Parameter	Min	Typ	Max	Units	Notes[1]
V_{IL}	Input Low Voltage			0.25 * V_{TTA}	V	2,3,7
V_{IL}	Input Low Voltage			0.29	V	8
V_{IH}	Input High Voltage	0.75 * V_{TTA}			V	2,7
V_{IH}	Input High Voltage	0.87			V	8
R_{ON}	Buffer On Resistance for VID[7:0]			100	Ohms	

1. Unless otherwise noted, all specifications in this table apply to all processor frequencies.
2. The V_{TTA} referred to in these specifications refers to instantaneous V_{TTA}.
3. Based on a test load of 50 Ohms to V_{TTA}.
4. R_{SYS_TERM} is the termination on the system and is not controlled by the Intel® Core™ i7 processor.
5. Applies to all signals, unless otherwise mentioned in Table 159.
6. Applies to PROCHOT# signal only. See Section 13.1.10.3.2 and Section 8.1 for information regarding Power-On Configuration options.
7. This specification only applies to VCCPWRGOOD and VTTPWRGOOD.
8. This specification only applies to DDR_DRAMPWROK.

§ §

14.0 Testability

The processor includes boundary-scan for board and system level testability.

14.1 Boundary-Scan

The processor is compatible with the IEEE 1149.1-2001 (Standard Test Access Port and Boundary-Scan Architecture) specification. See the specification for functionality.

After applying voltage to the power pins, the following initialization sequence must be completed prior to first TAP (Test Access Port) accesses that apply the boundary-scan test patterns:

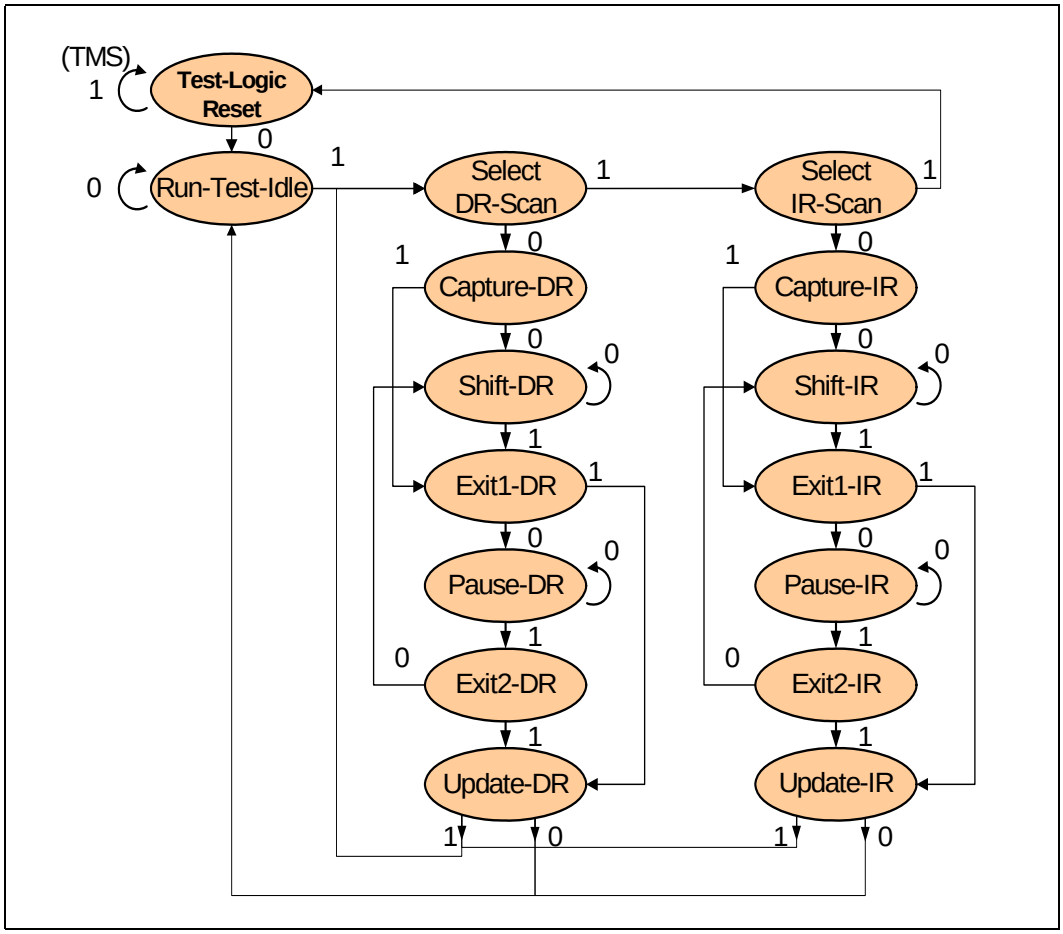
- The BCLK pins must be clocked.
- VCCPWRGOOD_1 pin must be held LOW for a minimum of 1us and then driven HIGH prior to the first TAP access (the other power good pins, VTTPWRGOOD and VCCPWRGOOD_0 must be asserted for correct TAP operation).
- The RSTIN# pin must be driven LOW (active) initially until after VCCPWRGOOD_1 is driven HIGH. RSTIN# may be held low or driven high during application of boundary-scan patterns.

14.2 TAP Controller Operation and State Diagram

Figure 88 shows the state diagram for the TAP controller Finite State Machine. The TAP controllers are asynchronously reset via the hardware TAP reset. Once reset is deasserted, the TAP controllers sample the TMS pin at the rising edge of TCK pin, and sequence through the states under the control of the TMS pin.

Holding the TMS pin high for five or more TCK cycles will take the TAP controllers to the Test-Logic-Reset state regardless of what state they are in. It is recommended that TMS be held high at the desassertion of reset to ensure deterministic operation of the TAP controllers. The TDO pin is output-enabled only during Shift-DR or Shift-IR states.

Figure 88. TAP Controller State Diagram



14.3 TAP Instructions and Opcodes

The TAP controllers support the boundary-scan instructions listed in:

- Table 176, “Processor Core TAP Controller Supported Boundary-Scan Instruction Opcodes” on page 517.
- Table 177, “Processor Un-Core TAP Controller Supported Boundary-Scan Instruction Opcodes” on page 517.
- Table 178, “Processor Integrated I/O TAP Controller Supported Boundary-Scan Instruction Opcodes” on page 518.

14.3.1 Processor Core TAP Controller

The instruction register length is 8 bits, and the capture value is "00000001" binary.

Table 176. Processor Core TAP Controller Supported Boundary-Scan Instruction Opcodes

Opcode (Hex)	Instruction	Selected Test Data Register	TDR Length
0x00h	EXTEST	Bypass	1
0x01h	SAMPLE/ PRELOAD (SAMPRE)	Bypass	1
0x02h	IDCODE	Device Identification	32
0x04h	CLAMP	Bypass	1
0x08h	HIGHZ	Bypass	1
0xFFh	BYPASS	Bypass	1
others	Reserved		

14.3.2 Processor Un-Core TAP Controller

The instruction register length is 8 bits, and the capture value is "00000001" binary.

Table 177. Processor Un-Core TAP Controller Supported Boundary-Scan Instruction Opcodes

Opcode (binary)	Instruction	Selected Test Data Register	TDR Length
0x00h	EXTEST	Boundary-scan	569
0x01h	SAMPLE/ PRELOAD (SAMPRE)	Boundary-scan	134
0x01h	SAMPLE/ PRELOAD (SAMPRE)	Boundary-scan	569
0x02h	IDCODE	Device Identification	32
0x04h	CLAMP	Bypass	1
0x08h	HIGHZ	Bypass	1
0xFFh	BYPASS	Bypass	1
others	Reserved		

14.3.3 Processor Integrated I/O TAP Controller

The instruction register length is 8 bits, and the capture value is "00000001" binary.

Table 178. Processor Integrated I/O TAP Controller Supported Boundary-Scan Instruction Opcodes

Opcode (binary)	Instruction	Selected Test Data Register	TDR Length
0x00h	EXTEST	Boundary-scan	115
0x01h	SAMPLE/ PRELOAD (SAMPRE)	Boundary-scan	115
0x02h	IDCODE	Device Identification	32
0x04h	CLAMP	Bypass	1
0x05h	EXTEST_TOGGLE	Boundary-scan	115
0x08h	HIGHZ	Bypass	1
0xFFh	BYPASS	Bypass	1
others	Reserved		

14.3.4 TAP Interface

This component contains several TAP controllers. The processor's TAP controllers connectivity is as follows: TDI --> Processor Un-Core --> Processor Execution Core 0 --> Processor Execution Core 1 --> Processor Execution Core 2--> Processor Execution Core 3--> Processor Integrated I/O --> TDO.

Figure 89. Processor TAP Controller Connectivity

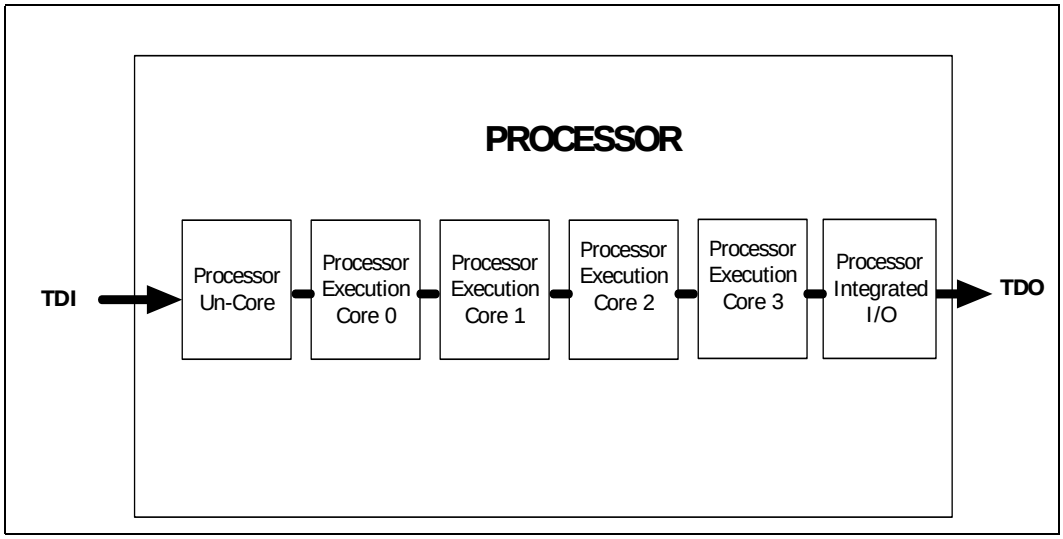
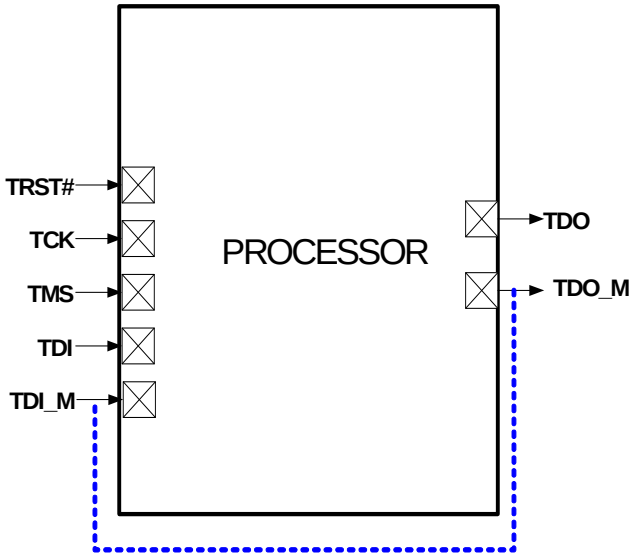


Figure 90. Processor TAP Connections



The processor uses seven dedicated pins to access the TAP as shown in Figure 90 and as described in Table 179.

Power must be applied and VCCPWRGOOD_0, VCCPWRGOOD_1 and VTTPWRGOOD must be driven high prior to using the TAP.

Table 179. Processor Boundary-Scan TAP Pin Interface

Pin	Direction	Description
TRST#	Input	Boundary-scan test reset pin, This signal has an ODT pull-up.
TCK	Input	Test clock pin for Boundary-scan TAP Controller and test logic. This signal has an ODT pull-down.
TMS	Input	Boundary-scan Test Mode Select pin. Sampled by the TAP on the Rising edge of TCK to control the operation of the TAP state machine. It is recommended that TMS is held high when the Boundary-scan reset is driven from low to high, to ensure deterministic operation of the test logic. This signal has an ODT pull-up.
TDI	Input	Boundary-scan Test Data Input pin, sampled on the Rising edge of TCK to provide serial test instructions and data. This signal has an ODT pull-up.
TDO	Output	Boundary-scan Test Data Output pin. In inactive drive state except when instructions or data are being shifted. TDO changes on the falling edge of TCK. This signal has no pull-up or pull-down.
TDI_M	Input	Intermediate Boundary-scan connection. This signal must be connected to TDO_M for correct operation. This signal has an ODT pull-up.
TDO_M	Output	Intermediate Boundary-scan connection. This signal must be connected to TDI_M for correct operation. This signal has no pull-up or pull-down.

14.4 TAP Port Timings

The TAP port timings are shown in Figure 91 and Table 180.

Figure 91. Boundary-Scan Port Timing Waveforms

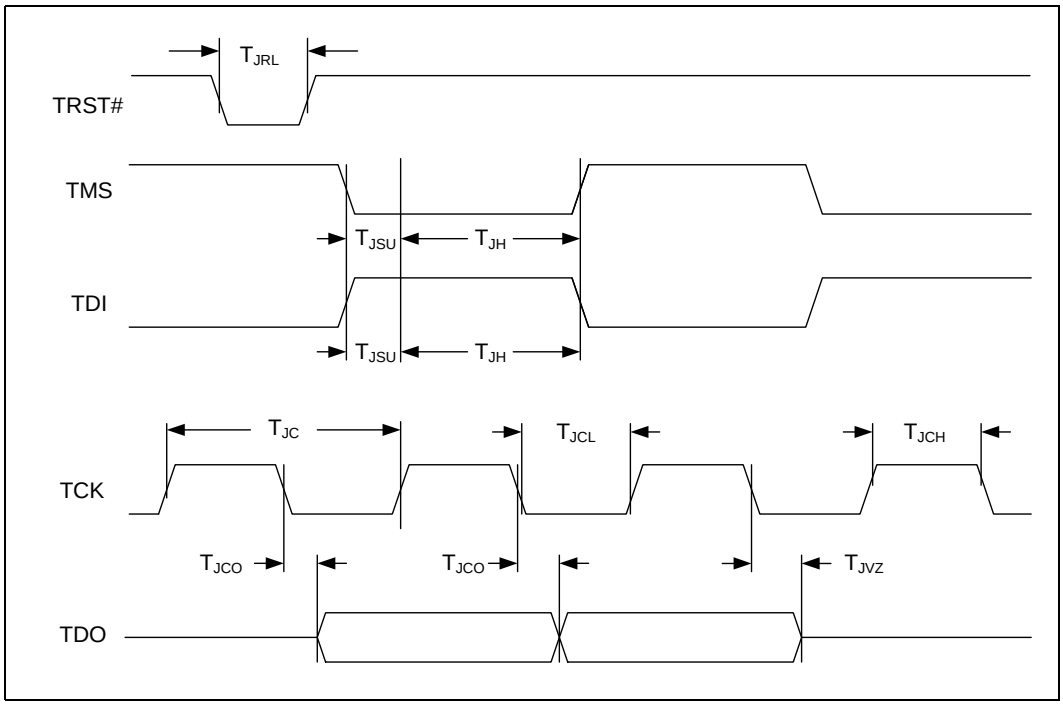


Table 180. Boundary-Scan Signal Timings

Symbol	Parameter	Min	Max	Unit	Notes
T_{JC}	Boundary-scan TCK clock period	31.25		ns	32 MHZ
T_{JCL}	Boundary-scan TCK clock low time	0.4 * T_{JC}			
T_{JCH}	Boundary-scan TCK clock high time	0.4 * T_{JC}			
T_{JSU}	Setup of TMS and TDI before TCK rising	8		ns	
T_{JH}	TMS and TDI hold after TCK rising	5		ns	
T_{JCO}	TCK falling to TDO output valid	0.5	7	ns	
T_{JVZ}	TCK falling to TDO output high-impedance		9	ns	
T_{JRL}	Active Boundary-scan Reset Low time	2		T_{TCK}	

14.5 Boundary-Scan Register Definition

See the Boundary-Scan Description Language (BSDL) file(s) for details about the boundary-scan register structure, application information, and design warnings.

§ §