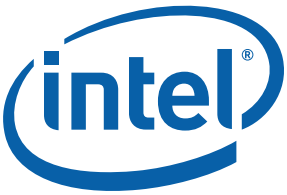


Intel® 7300 Chipset Memory Controller Hub (MCH)

Datasheet

September 2007

Document Number: 318082-001

Notice: This document contains preliminary information on new products in production. The specifications are subject to change without notice. Verify with your local Intel sales office that you have the latest datasheet before finalizing a design

The Intel® 7300 Chipset Memory Controller Hub (MCH) 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.

64-bit computing on Intel architecture requires a computer system with a processor, chipset, BIOS, operating system, device drivers and applications enabled for Intel® 64 architecture. Performance will vary depending on your hardware and software configurations. Consult with your system vendor for more information.

Intel® Active Management Technology requires the platform to have an Intel® AMT-enabled chipset, network hardware and software, as well as connection with a power source and a corporate network connection. With regard to notebooks, Intel AMT may not be available or certain capabilities may be limited over a host OS-based VPN or when connecting wirelessly, on battery power, sleeping, hibernating or powered off. For more information, see http://www.intel.com/technology/iamt.

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

Intel, Xeon, Pentium, Intel Core, Intel XScale, Intel 6700PXH 64-bit PCI Hub, Intel 6702PXH 64-bit PCI Hub, Intel 64, Intel I/O Acceleration Technology (Intel I/OAT), Intel 631xESB/632xESB I/O Controller Hub, Intel IOP 348 I/O Processor, 82575 Gigabit Ethernet controller, Intel Active Management Technology (Intel AMT), Intel QuickData Technology, and the Intel logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

*Other brands and names are the property of their respective owners.

Contents

Figures

Tables

Revision History

Document Number	Revision Number	Description	Date
xxxxxxxx	-001	Initial release	September 2007

§

1 Intel® 7300 Chipset Introduction

The Intel® 7300 Chipset Memory Controller Hub (MCH) Datasheet documents the features, ballout, and registers of the Intel® 7300 Chipset.

1.1 Reference Documentation

Table 1-1. Reference Documentation

Document	Document Number/ Source
Intel® 7300 Chipset Memory Controller Hub (MCH) Specification Update	318083-001
Intel® 7300 Chipset Memory Controller Hub (MCH) Thermal Mechanical Design Guide	318086-001
Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series Datasheet	318080-001
Intel® 6700PXH 64-bit PCI Hub Datasheet	302628-002
Intel® 6700PXH 64-bit PCI Hub Thermal/Mechanical Design Guidelines	302817-003
Intel® 631xESB/632xESB I/O Controller Hub Datasheet	313082-001
Intel® 631xESB/632xESB I/O Controller Hub Thermal/Mechanical Design Guidelines	313073-001
Intel® 82563EB/82564EB LAN on Motherboard Design Guide Application Note (AP-467) Networking Silicon	www.intel.com
JEDEC FBD Memory Specification	www.jedec.org
mPGA604 Socket Design Guide	254239-002

Table 1-2. Intel® 631xESB/632xESB I/O Controller Hub Reference Documentation (Sheet 1 of 2)

Document	Document Number/Source
PCI Express Base Specification*	http://www.pcisig.com/specifications/pciexpress/
PCI Express Card Electromechanical Specification*	http://www.pcisig.com/specifications/pciexpress/
Low Pin Count Interface Specification (LPC)	http://developer.intel.com/design/chipsets/industry/lpc.htm
Wired for Management Baseline (WfM)	http://developer.intel.com/ial/WfM/wfmspecs.htm
System Management Bus Specification (SMBus)	http://www.smbus.org/specs/
PCI Local Bus Specification (PCI)	http://pcisig.com/specs.htm
PCI Power Management Specification	http://pcisig.com/specs.htm
PCI Standard Hot-Plug Controller and Subsystem Specification	http://www.pcisig.com/specifications/conventional/pci_hot_plug
PCI-X Electrical and Mechanical Addendum to the PCI Local Bus Specification*	http://www.pcisig.com/specifications/pcix_20
Universal Serial Bus Specification (USB)	http://www.usb.org
Advanced Configuration and Power Interface (ACPI)	http://www.acpi.info/spec.htm

Table 1-2. Intel® 631xESB/632xESB I/O Controller Hub Reference Documentation (Sheet 2 of 2)

Document	Document Number/Source
Enhanced Host Controller Interface Specification for Universal Serial Bus (EHCI)	http://developer.intel.com/technology/usb/ehcispec.htm
Serial ATA Specification	http://www.serialata.org/cgi-bin/SerialATA10gold.zip
Alert Standard Format Specification	http://www.dmtf.org/standards/standard_alert.php
IEEE 802.3 Fast Ethernet	http://standards.ieee.org
AT Attachment - 6 with Packet Interface (ATA/ATAPI - 6)	http://T13.org (T13 1410D)
Intel® 631xESB/632xESB I/O Controller Hub Datasheet	313082-001
Front Panel I/O Connectivity Design Guide	http://www.formfactors.org/developer/fpio_design_guideline.pdf
Intel® 631xESB/632xESB I/O Controller Hub Thermal Design Guidelines	http://www.intel.com/design/chipsets/designex/313073.htm
ATX Specification Rev 2.02	ftp://download.intel.com/design/motherbd/atx_202.pdf
SSI Specification	http://www.ssiforum.org/html/adoptedspecs.asp
Small Form Factor Specification SFF-8049	Small Form Factor Committee
82563EB/82564EB Gigabit Platform LAN Connect Datasheet	http://download.intel.com/design/network/datashts/82563_datasheet.pdf
Intel® 82563EB/82564EB LAN on Motherboard Design Guide Application Note (AP-467) Networking Silicon	ftp://download.intel.com/design/network/applnots/ap467.pdf
ESB2 LAN EEPROM Map and Programming Information Application Note (AP-477)	http://download.intel.com/design/network/applnots/ap477.pdf
82563EB/82564EB Gigabit Platform LAN Connect Specification Update	http://www.intel.com/design/network/specupdt/82563_64EB_spupdt.htm

Table 1-3. 82575 Reference Documentation

Document	Document Number/Source
82575EA Gigabit Ethernet Controller Datasheet	http://download.intel.com/design/network/datashts/317697.pdf
Intel® 82575 Gigabit Ethernet Controller Design Guide	http://download.intel.com/design/network/desguides/317698.pdf
82575EA/EB/ES Thermal Design Considerations	http://download.intel.com/design/network/applnots/317699.pdf

Table 1-4. Intel IOP 348 I/O Processor Design References (Sheet 1 of 2)

Design References
Transmission Line Design Handbook, Brian C. Wadell
Microstrip Lines and Slotlines, K. C. Gupta. Et al.
PCI-X Addendum to the PCI Local Bus Specification, Revision 1.0a
*PCI-X Electrical Subgroup Report, Version*1.0
Design, Modeling and Simulation Methodology for High Frequency PCI-X Subsystems, Moises Cases, Nam Pham, Dan Neal www.pcisig.com

Table 1-4. Intel IOP 348 I/O Processor Design References (Sheet 2 of 2)

Design References
High-Speed Digital Design "A Handbook of Black Magic" Howard W. Johnson, Martin Graham
"Terminating Differential Signals on PCBs", Steve Kaufer, Kelee Crisafulli, Printed Circuit Design, March 1999
"Board Design Guidelines for PCI Express Interconnect", http://www.intel.com/technology/pciexpress/downloads/PCI_EI_PCB_Guidelines.pdf

Intel documentation is available from your local Intel Sales Representative or Intel Literature Sales.

To obtain Intel literature call
(1-800-548-4725) or visit the Intel website at http://www.intel.com

Table 1-5. Intel IOP 348 I/O Processor Intel-Related Documentation

Document Title	Order #
Intel® Packaging Databook	240800

1.2 Conventions and Terminology

1.2.1 Terminology

This section provides the definitions of some of the terms used in this document.

Table 1-6. General Terminology (Sheet 1 of 6)

Terminology	Description
Agent	A logical device connected to a bus or shared interconnect that can either initiate accesses or be the target of accesses. Each thread executing within a processor is a unique agent.
Aka	Also known as.
Asserted	Signal is set to a level that represents logical true. For signals that end with "#", this means driving a low voltage. For other signals, it means driving a high voltage.
Atomic operation	A series of operations, any one of which cannot be observed to complete unless all are observed to complete.
Bank	DRAM chips are divided into multiple banks internally. Commodity parts are all 4 bank, which are the only type the MCH supports. Each bank acts somewhat like a separate DRAM, opening and closing pages independently, allowing different pages to be open in each. Most commands to a DRAM target a specific bank, but some commands (i.e., Precharge All) are targeted at all banks. Multiple banks allow higher performance by interleaving the banks and reducing page miss cycles.
Buffer	1. A random access memory structure. 2. The term I/O buffer is also used to describe a low level input receiver and output driver combination.
Cache Line	The unit of memory that is copied to and individually tracked in a cache. Specifically, 64 bytes of data or instructions aligned on a 64-byte physical address boundary.
CDM	Central Data Manager. A custom array within the MCH that acts as a temporary repository for system data in flight between the various ports: FSBs, FBDs, ESI, and PCI Express*.
Cfg, Config	Abbreviation for "Configuration".
Channel	In the MCH a FBD channel is the set of signals that connects to one set of FBD DIMMs. The MCH has up to four FBD channels.
Character	The raw data byte in an encoded system (e.g. the 8b value in a 8b/10b encoding scheme). This is the meaningful quantum of information to be transmitted or that is received across an encoded transmission path.
Chipset Core	The MCH internal base logic.
Coherent	Transactions that ensure that the processor's view of memory through the cache is consistent with that obtained through the I/O subsystem.

Table 1-6. General Terminology (Sheet 2 of 6)

Terminology	Description
Command	The distinct phases, cycles, or packets that make up a transaction. Requests and Completions are referred to generically as Commands.
Completion	A packet, phase, or cycle used to terminate a Transaction on a interface, or within a component. A Completion will always refer to a preceding Request and may or may not include data and/or other information.
Core	The internal base logic in the MCH.
CRC	Cyclic Redundancy Check; A number derived from, and stored or transmitted with, a block of data in order to detect corruption. By recalculating the CRC and comparing it to the value originally transmitted, the receiver can detect some types of transmission errors.
Critical Word First	On the DRAM, processor, and memory interfaces, the requestor may specify a particular word to be delivered first. This is done using Address bits of lower significance than those required to specify the cache line to be accessed. The remaining data is then returned in a standardized specified order.
DDR	Double Data Rate SDRAM. DDR describes the type of DRAMs that transfers two data items per clock on each pin. This is the only type of DRAM supported by the MCH.
Deasserted	Signal is set to a level that represents logical false.
Deferred Transaction	A processor bus Split Transaction. On the processor bus, the requesting agent receives a Deferred Response which allows other transactions to occur on the bus. Later, the response agent completes the original request with a separate Deferred Reply transaction or by Deferred Phase.
Delayed Transaction	A transaction where the target retries an initial request, but without notification to the initiator, forwards or services the request on behalf of the initiator and stores the completion or the result of the request. The original initiator subsequently re-issues the request and receives the stored completion
DFx (DFD, DFM, DFT, DFV)	DFD = Design for Debug DFM = Design for Manufacturing DFT = Design for Testability DFV = Design for Validation
DIMM	Dual-in-Line Memory Module. A packaging arrangement of memory devices on a socketable substrate.
Double-Sided DIMM	Terminology often used to describe a DIMM that contain two DRAM rows. Generally a Double-sided DIMM contains two rows, with the exception noted above. This terminology is not used within this document.
Downstream	See Terminology entry of "Inbound (IB)/Outbound (OB), AKA Upstream/DownStream, Northbound/Southbound, Upbound/Downbound"
DRAM Page (Row)	The DRAM cells selected by the Row Address.
Dword	A reference to 32 bits of data on a naturally aligned four-byte boundary (i.e. the least significant two bits of the address are 00b).
ECC	Error Correcting Code
"ESB2"	Intel® 631xESB/632xESB I/O Controller Hub (code named ESB2 - Enterprise South Bridge 2)
EM64T	Intel® Extended Memory 64-bit Technology
FBD	Fully Buffered DDR2
FBD Channel	One electrical interface to one or more Fully Buffered DDR2 DIMM
FSB	Processor Front-Side Bus. This is the bus that connects the processor to the MCH.
Full Duplex	A connection or channel that allows data or messages to be transmitted in opposite directions simultaneously.
GB/s	Gigabytes per second (10 9 bytes per second)
Gb/s	Gigabits per second (10 9 bits per second)
Hardwired	A parameter that has a fixed value.
Half Duplex	A connection or channel that allows data or messages to be transmitted in either direction, but not simultaneously.
Host	This term is used synonymously with processor.

Table 1-6. General Terminology (Sheet 3 of 6)

Terminology	Description
I/O	1. Input/Output. 2. When used as a qualifier to a transaction type, specifies that transaction targets Intel® architecture-specific I/O space. (e.g., I/O read)
Implicit Writeback	A snoop initiated data transfer from the bus agent with the modified Cache Line to the memory controller due to an access to that line.
Inband	Communication that is multiplexed on the standard lines of an interface, rather than requiring a dedicated signal.
Inbound	See Terminology entry of "Inbound (IB)/Outbound (OB), AKA Upstream/DownStream, Northbound/Southbound, Upbound/Downbound."
Incoming	A transaction or data that enters the MCH.
Inbound (IB)/Outbound (OB), AKA Upstream/ DownStream, Northbound/ Southbound, Upbound/Downbound	Up, North, or Inbound is in the direction of the processor, Down, South, or Outbound is in the direction of I/O (SDRAM, SMBus).
Initiator	The source of requests. An agent sending a request packet on PCI Express is referred to as the Initiator for that Transaction. The Initiator may receive a completion for the Request.
Isochronous	A classification of transactions or a stream of transactions that require service within a fixed time interval.
Layer	A level of abstraction commonly used in interface specifications as a tool to group elements related to a basic function of the interface within a layer and to identify key interactions between layers.
Legacy	Functional requirements handed down from previous chipsets or PC compatibility requirements from the past.
Line	Cache line
Link	The layer of an interface that handles flow control and often error correction by retry.
Lock	A sequence of transactions that must be completed atomically.
LSb	Least Significant Bit
LSB	Least Significant Byte
Master	A device or logical entity that is capable of initiating transactions. A Master is any potential Initiator.
Master Abort	A response to an illegal request. Reads receive all ones. Writes have no effect.
MB/s	Megabytes per second (10 6 bytes per second)
MCH	The Memory Controller Hub component that contains the processor interface, FBD interface and memory controller, and PCI Express interfaces. It communicates with the I/O controller hub (Intel® 631xESB/632xESB I/O Controller Hub) over a proprietary interconnect called the Enterprise South Bridge Interface (ESI).
Mem	Used as a qualifier for transactions that target memory space. (e.g. A mem read to I/O)
Memory Issue	Committing a request to DDR or, in the case of a read, returning the read header.
Mesochronous	Distributed or common referenced clock
Metastability	A characteristic of flip flops that describes the state where the output becomes non-deterministic. Most commonly caused by a setup or hold time violation.
Mirroring	RAID-1. Please see RAID for detail descriptions.
MMIO	Memory Mapped I/O. Any memory access to PCI Express.
MMCFG	Memory Mapped Configuration. A memory transaction that accesses configuration space.
MSb	Most Significant Bit
MSB	Most Significant Byte
MTBF	Mean Time Between Failure
Multi Independent Bus (MIB)	A front side bus architecture with one processor on each bus, rather than a FSB shared between multiple processor agents. This is the bus that connects a processor to the MCH. The terms MIB and FSB are used interchangeably and reflect the same meaning. The MIB architecture provides improved performance by allowing increased FSB speeds and bandwidth.

Table 1-6. General Terminology (Sheet 4 of 6)

Terminology	Description
Non-Coherent	Transactions that may cause the processor's view of memory through the cache to be different with that obtained through the I/O subsystem.
Outbound	See Terminology entry of "Inbound (IB)/Outbound (OB), AKA Upstream/DownStream, Northbound/Southbound, Upbound/Downbound".
Outgoing	A transaction or completion that exits the MCH. Peer to Peer Transactions that occur between two devices below the PCI Express or ESI ports.
Packet	The indivisible unit of data transfer and routing, consisting of a header, data, and CRC.
Page Hit	An access to an open page, or DRAM row. The data can be supplied from the sense amps at low latency.
Page Miss (Empty Page)	An access to a page that is not buffered in sense amps and must be fetched from DRAM array. Address Bit Permuting Address bits are distributed among channel selects, DRAM selects, bank selects to so that a linear address stream accesses these resources in a certain sequence.
Page Replace Aka Page Miss, Row Hit/Page Miss	An access to a row that has another page open. The page must be transferred back from the sense amps to the array, and the bank must be precharged.
PCI	Peripheral Component Interconnect Local Bus. A 32-bit or 64-bit bus with multiplexed address and data lines that is primarily intended for use as an interconnect mechanism within a system between processor/memory and peripheral components or add-in cards.
PCI 2.3 compliant	Refers to compliance to the *PCI Local Bus Specification*, Revision 2.3
Plesiochronous	Each end of a link uses an independent clock reference. Support of this operational mode places restrictions on the absolute frequency difference, as specified by PCI Express, which can be tolerated between the two independent clock references.
Posted	A Transaction that is considered complete by the initiating agent or source before it actually completes at the Target of the Request or destination. All agents or devices handling the Request on behalf of the original Initiator must then treat the Transaction as being system visible from the initiating interface all the way to the final destination. Commonly refers to memory writes.
Primary PCI	The physical PCI bus that is driven directly by the Intel® 631xESB/632xESB I/O Controller Hub component. Communication between PCI and the MCH occurs over HI. Note that even though the Primary PCI bus is referred to as PCI it is **not** PCI Bus 0 from a configuration standpoint.
Push Model	Method of messaging or data transfer that predominately uses writes instead of reads.
Queue	A storage structure for information. Anything that enters a queue will exit eventually. The most common policy to select an entry to read from the queue is FIFO (First In First Out).

Table 1-6. General Terminology (Sheet 5 of 6)

Terminology	Description
RAID	Redundant Array of Independent Disks. RAID improves performance by disk striping, which interleaves bytes or groups of bytes across multiple drives, so more than one disk is reading and writing simultaneously. Fault tolerance is achieved by mirroring or parity. Mirroring is 100% duplication of the data on two drives (RAID-1), and parity is used (RAID-3 and 5) to calculate the data in two drives and store the results on a third: a bit from drive 1 is XOR'd with a bit from drive 2, and the result bit is stored on drive 3 (see OR for an explanation of XOR). A failed drive can be hot swapped with a new one, and the RAID controller automatically rebuilds the lost data.RAID can be classified into the following categories: RAID-0 RAID-0 is disk striping only, which interleaves data across multiple disks for better performance. It does not provide safeguards against failure. RAID-1 Uses disk mirroring, which provides 100% duplication of data. Offers highest reliability, but doubles storage cost. RAID-2 Bits (rather than bytes or groups of bytes) are interleaved across multiple disks. The Connection Machine used this technique, but this is a rare method. RAID-3 Data are striped across three or more drives. Used to achieve the highest data transfer, because all drives operate in parallel. Parity bits are stored on separate, dedicated drives. RAID-4 Similar to RAID-3, but manages disks independently rather than in unison. Not often used. RAID-5 Most widely used. Data are striped across three or more drives for performance, and parity bits are used for fault tolerance. The parity bits from two drives are stored on a third drive. RAID-6 Highest reliability, but not widely used. Similar to RAID-5, but does two different parity computations or the same computation on overlapping subsets of the data. RAID-10 Actually RAID-1,0. A combination of RAID-1 and RAID-0 (mirroring and striping). Above definitions can be extended to DRAM memory system as well. To avoid confusion, the RAID scheme for memory is referred as **memory-RAID**. **Memory mirroring** scheme is actually **memory-RAID-1**.
RASUM	Reliability, Availability, Serviceability, Usability, and Manageability, which are all important characteristics of servers.
Receiver, Rcvr	1. The Agent that receives a Packet across an interface regardless of whether it is the ultimate destination of the packet. 2. More narrowly, the circuitry required to convert incoming signals from the physical medium to more perceptible forms.
Request	A packet, phase, or cycle used to initiate a Transaction on a interface, or within a component.
Reserved	The contents or undefined states or information are not defined at this time. Using any reserved area is not permitted. Reserved register bits must be set to '0'.
RMW	Read-Modify-Write operation
Row	A group of DRAM chips that fill out the data bus width of the system and are accessed in parallel by each DRAM command.
Row Address	The row address is presented to the DRAMs during an Activate command, and indicates which page to open within the specified bank (the bank number is presented also).
Scalable Bus	Processor-to-MCH interface. The compatible mode of the Scalable Bus is the P6 Bus. The enhanced mode of the Scalable Bus is the P6 Bus plus enhancements primarily consisting of source synchronous transfers for address and data, and FSB interrupt delivery. The Intel® Pentium® 4 processor implements a subset of the enhanced mode.
SDDC	Single Device Disable Code; aka x4 or x8 chip-disable Hamming code to protect single DRAM device (x4 or x8 data width) failure.
SDR	Single Data Rate SDRAM.
SDRAM	Synchronous Dynamic Random Access Memory.
SEC/DED	Single-bit Error Correct / Double-symbol Error Detect
Serial Presence Detect (SPD)	A 2-signal serial bus used to read and write Control registers in the SDRAM's via the SMBus protocol
Single-Sided DIMM	Terminology often used to describe a DIMM that contains one DRAM row. Usually one row fits on a single side of the DIMM allowing the backside to be empty.

Table 1-6. General Terminology (Sheet 6 of 6)

Terminology	Description
Simplex	A connection or channel that allows data or messages to be transmitted in one direction only.
SMBus	System Management Bus. Mastered by a system management controller to read and write configuration registers. Signaling and protocol are loosely based on I^2C*, limited to 100 KHz.
Snooping	A means of ensuring cache coherency by monitoring all coherent accesses on a common multi-drop bus to determine if an access is to information resident within a cache. The Intel® 7300 Chipset Memory Controller Hub (MCH) ensures coherency by initiating snoops on the processor busses with the address of any line that might appear in a cache on that bus.
Split Lock Sequence	A sequence of transactions that occurs when the target of a lock operation is split across a processor bus data alignment or Cache Line boundary, resulting in two read transactions and two write transactions to accomplish a read-modify-write operation.
Split Transaction	A transaction that consists of distinct Request and Completion phases or packets that allow use of bus, or interconnect, by other transactions while the Target is servicing the Request.
SSTL	Stub-Series Terminated Logic
SSTL_2	Stub Series Terminated Logic for 2.6 Volts (DDR)
Symbol	An expanded and encoded representation of a data Byte in an encoded system (e.g. the 10-bit value in a 8-bit/10-bit encoding scheme). This is the value that is transmitted over the physical medium.
Symbol Time	The amount of time required to transmit a symbol.
Target	A device that responds to bus Transactions. [PCI-X*] The agent receiving a request packet is referred to as the Target for that Transaction. [PCI Express]
Tenured Transaction	A transaction that holds the bus, or interconnect, until complete, effectively blocking all other transactions while the Target is servicing the Request.
TID	Transaction Identifier; A multi-bit field used to uniquely identify a transaction. Commonly used to relate a Completion with its originating Request in a Split Transaction system.
Transaction, Txn	An overloaded term that represents an operation between two or more agents that can be comprised of multiple phases, cycles, or packets.
Transmitter	1. The Agent that sends a Packet across an interface regardless of whether it was the original generator of the packet. 2. More narrowly, the circuitry required to drive signals onto the physical medium.
Upstream	See Terminology entry of "Inbound (IB)/Outbound (OB), AKA Upstream/DownStream, Northbound/Southbound, Upbound/Downbound"

Table 1-7. Intel® 631xESB/632xESB I/O Controller Hub Conventions and Terminology

Acronym, Convention/ Terminology	Definition
ASF	Alert Standard Format
Anti-Etch	Any plane-split, void or cutout in a Vcc or GND plane is referred to as an anti-etch
BER	Bit Error Rate
BGA	Ball Grid Array
BMC	Baseboard Management Controller
CMC	Common Mode Choke
ESI	Enterprise South Bridge Interface
EHCI	Enhanced Host Controller Interface
EMI	Electromagnetic Interference
"ESB2"	Intel® 631xESB/632xESB I/O Controller Hub (code named ESB2 - Enterprise South Bridge Second Generation)
ESD	Electrostatic Discharge
FS	Full-speed. Refers to USB
HS	High Speed. Refers to USB
LCI	LAN Connect Interface
LOM	LAN on Motherboard
LPC	Low Pin Count
LS	Low-speed. Refers to USB
MC	Modem Codec
PCM	Pulse Code Modulation
PLC	Platform LAN Connect
RTC	Real Time Clock
SATA	Serial ATA
SMBus	System Management Bus – a two-wire interface through which various system components can communicate
SPD	Serial Presence Detect
S/PDIF	Sony/Philips Digital Interface*
STD	Suspend To Disk
STR	Suspend To RAM
TCO	Total Cost of Ownership
TDM	Time Division Multiplexed
TDR	Time Domain Reflectometry
UHCI	Universal Host Controller Interface
USB	Universal Serial Bus

Table 1-8. Intel IOP 348 I/O Processor Terminology and Definitions (Sheet 1 of 2)

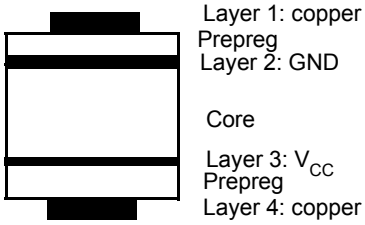
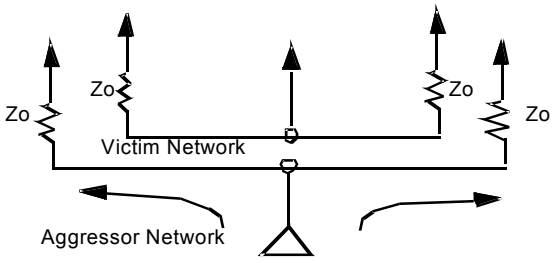
Term	Definition	
Stripline		Stripline in a PCB is composed of the conductor inserted in a dielectric with GND planes to the top and bottom. ***Note:*** An easy way to distinguish stripline from microstrip is that you need to strip away layers of the board to view the trace on stripline.
Microstrip		Microstrip in a PCB is composed of the conductor on the top layer above the dielectric with a ground plane below
Prepreg	Material used for the lamination process of manufacturing PCBs. It consists of a layer of epoxy material that is placed between two cores. This layer melts into epoxy when heated and forms around adjacent traces.	
Core	Material used for the lamination process of manufacturing PCBs. This material is two sided laminate with copper on each side. The core is an internal layer that is etched.	
PCB	 Example of a Four-Layer Stack	Printed circuit board. Example manufacturing process consists of the following steps: • Consists of alternating layers of core and prepreg stacked • The finished PCB is heated and cured. • The via holes are drilled • Plating covers holes and outer surfaces • Etching removes unwanted copper • Board is tinned, coated with solder mask and silk screened
DDR	Double Data Rate Synchronous DRAM. Data clocked on both rising and falling edges of clock.	
DDR2	DDR2 is backward compatible with DDR I. It allows 4.3 GBytes/sec. for a clock rate of 533 MHz and 3.2 GB/sec. for a clock rate of 400 MHz.	
DIMM	Dual Inline Memory Module	
Source Synchronous DDR	With source-synchronous DDR interfaces, data and clock transport from a transmitter to a receiver, and the receiver interface uses the clock to latch the accompanying data.	
SSTL_2	Series Stub Terminated Logic for 2.5 V	
JEDEC	Provides standards for the semiconductor industry.	
DLL	Delay Lock Loop - DDR feature used to provide appropriate strobe delay to clock in data.	
PLL	Phase Lock Loop - A phase-locked loop (PLL) is an electronic circuit with a voltage- or current-driven oscillator that is constantly adjusted to match in phase (and thus lock on) the frequency of an input signal.	
Aggressor	A network that transmits a coupled signal to another network is aggressor network. 	
Victim	A network that receives a coupled cross-talk signal from another network is a victim network.	
Network	The trace of a PCB that completes an electrical connection between two or more components.	
Stub	Branch from a trunk terminating at the pad of an agent.	

Table 1-8. Intel IOP 348 I/O Processor Terminology and Definitions (Sheet 2 of 2)

Term	Definition
ISI	Intersymbol Interference (ISI). This occurs when a transition that has not been completely dissipated, interferes with a signal being transmitted down a transmission line. ISI can impact both the timing and signal integrity. It is dependent on frequency, time delay of the line and the refection coefficient at the driver and receiver. Examples of ISI patterns that could be used in testing at the maximum allowable frequencies are the sequences shown below: 0101010101010101 0011001100110011 0001110001110001111
CRB	Customer Reference Board
PC4300	JEDEC Names for DDR2 533 based on peak data rates. PC4300= clock of 266 MHz * 2 data words/clock * 8 bytes = 4256 MB/sec
PC5300	JEDEC Names for DDR2 667 based on peak data rates. PC5300= clock of 333 MHz * 2 data words/clock * 8 bytes = 5328 MB/sec
Host processor	Processor located upstream from the Intel IOP 348 I/O Processor IOP
Local processor	Intel XScale® core within Intel IOP 348 I/O Processor IOP
Downstream	• PCI Express: At or toward a PCI Express port directed away from root complex (to a bus with a higher number). • PCI-X: At or toward a PCI bus with a higher number (after configuration) away from host processor.
Upstream	• PCI Express: At or toward a PCI Express port directed to the PCI Express root complex (to a bus with a lower number). • PCI-X: At or toward a PCI bus with a higher number (after configuration) toward host processor.
Local memory	Memory subsystem on the Intel® XScale core DDR SDRAM or Peripheral Bus Interface busses.
WORD	16-bits of data.
DWORD	32-bit data word.
QWORD	64-bit data word
Local bus	Internal Bus.
Outbound	At or toward the PCI interface of the ATU from the Internal Bus.
Inbound	At or toward the Internal Bus from the PCI interface of the ATU.
Core processor	Intel XScale core within the part.
Flip Chip	FC-BGA (flip chip-ball grid array) chip packages are designed with core flipped up on the back of the chip, facing away from the PCB. This allows more efficient cooling of the package.
Mode Conversion	Mode Conversions are due to imperfections on the interconnect which transform differential mode voltage to common mode voltage and common mode voltage to differential voltage.
ROMB	Raid on motherboard
ODT	On Die Termination - eliminates the need for termination resistors by placing the termination at the chip.

§

2 System Overview

The Intel® 7300 Chipset is paired with the Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series and is comprised of the Intel® 7300 Chipset Memory Controller Hub (MCH) , the Intel® 631xESB/632xESB I/O Controller Hub , and the I/O subsystem. The platform is targeted for the multi-processor server space. The MCH is configured for symmetric multiprocessing across four independent front side bus interfaces that connect to the Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series. The quad independent front side bus architecture provides improved performance by allowing increased front side bus speeds and bandwidth. Each front side bus on the MCH uses a 64-bit wide, 1066 MHz data bus capable of transferring data at 8.5 GB/s for a total bandwidth of 34 GB/s. The MCH supports a 40-bit wide address bus, capable of addressing up to 512 GB of memory. The MCH is the priority agent for all four front side bus interfaces, and is optimized for one processor on each bus.

The MCH connects to up to 32 fully buffered DIMMs (four memory channels with up to eight DIMMs per channel). This provides a maximum theoritical bandwidth of 8.5 GB/s for writes 17 GB/s for reads when DDR2 533 MHz memory is used. When DDR2 667 MHz memory is used, the maximum theoritical write bandwidth is 10.7 GB/s and the maximum theoritical read bandwidth is 21.3 GB/s.

The MCH provides seven x4 PCI Express* interfaces and one x4 ESI interface to the Intel® 631xESB/632xESB I/O Controller Hub. If more bandwidth is desired, it is possible to configure x4 ports 2 & 3, 4 &5, 6 & 7 into single x8 ports. Each PCI Express port on the MCH provides 4 GB/s bidirectional bandwidth when configured as a x8 port or 2 GB/s bidirectional bandwidth when configured as a x4 port.

In addition to these performance features, the platform also provides a wide range of RAS (Reliability, Availability, & Serviceability) features such as memory interface ECC (including internal data paths), x4/x8 Single Device Data Correction, CRC, parity protection, out-of-band register access via SMBus and JTAG, memory mirroring, memory sparing, and Hot-Plug support on the PCI Express interface.

2.1 Processor Features

The Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series processor is intended for high performance workstation and server systems. Several architectural and microarchitectural enhancements have been added to this processor including four processor cores in the processor package (2 dual-core dies), an increased L2 cache size, and a new instruction cache. Table 2-1 provides a feature set overview of the Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series processor.

Table 2-1. Processor Feature Set Overview (Sheet 1 of 2)

Feature	Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series Processor
L1 Cache Sizes	Instruction Cache = 32 KB Data Cache 32 KB
L2 Cache Size	4 MB shared between two cores on one die for a total of 8 MB per processor
Data Bus Transfer Rate	8.5 GB/s

Table 2-1. Processor Feature Set Overview (Sheet 2 of 2)

Feature	Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series Processor
Quad Core Support	4 Cores per Processor
Multi-Processor Support	1 Processor per FSB
Package	604-pin FC-mPGA4
Core Operating Voltage	1.0000 to 1.5000 V
Front Side Bus Operating Voltage	1.20 V
VRM	Version 11.0

2.2 Intel® 7300 Chipset MCH Features

The MCH is in a 2013-ball FC-BGA package with the following functionality:

2.2.1 Front Side Bus

- Quad independent processor buses (1 processor per bus) for improved data bandwidth and frequency
- Each bus supports up to 4 physical processor cores
- Supports Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series and Dunnington processors
- Two load point-to-point operation at 266 MHz (data @ 1066 MT/s) providing a bandwidth of 8.5 GB/s (1066 MT/s)
- 40-bit address ability support
- Double-pumped address buses with a peak address bandwidth of 533 million addresses/second
- Parity protection on address and data signals

2.2.2 System Memory Interface

- Four Fully Buffered DIMM (FBD) channels, organized as two branches each having two channels
  - — Branch 0 contains channel 0 and channel 1
  - — Branch 1 contains channel 2 and channel 3
- Channels within a branch must be populated identically; branches must be populated identically if mirrored mode is used
- Up to eight DIMMs per channel supported
- Supports FBD DDR2 DIMMs using 256 Mb, 512 Mb, 1 Gbit, 2 Gbit or 4 Gbit DDR2 DRAMs in x4/x8 device widths, single- or double-rank DIMMs
- Memory speeds of 533 MHz DDR2/3.2 GHz FBD and 667 DDR2/4.0 GHz FBD are supported
- Supports a maximum of 512 GB of memory (256 GB in mirrored mode)
- x4/x8 Single Device Data Correction (SDDC) supported
- Supports ECC DIMMs

2.2.3 PCI Express Interfaces

The Intel® 7300 Chipset MCH provides multiple PCI Express generation 1 interfaces. Some of the MCH PCI Express features are:

- Point-to-point, serial bi-directional interconnect
- One x4 ESI link to Intel® 631xESB/632xESB I/O Controller Hub (MCH port 0 to Intel® 631xESB/632xESB I/O Controller Hub port 4)
- Seven x4 PCI Express ports (Ports 1-7)
- x4 link pairs can be combined to form x8 links
- Ports 2 and 3 are the recommended ports connected to the Intel® 631xESB/632xESB I/O Controller Hub for a total of three x4 links between the MCH and Intel® 631xESB/632xESB I/O Controller Hub
- Each signal is 8b/10b encoded with an embedded clock
- Signaling bit rate of 2.5 Gbit/sec/lane/direction; for a x4 link, bandwidth is 1 GB/sec in each direction
- Hot Insertion and Removal supported with the addition of Hot-Plug control circuitry

2.2.4 SMBus Interfaces

The MCH provides six fully functional System Management Bus (SMBus) Revision 2.0 compliant target interfaces. These interfaces are used to support the following platform level operations:

- 1 System management bus (CFGSMB, slave device)
- 1 PCI Express Hot-Plug virtual pin port SMBus (VPPSMB, master device)
- 4 FBD serial presence detect SMBuses (SPD[0:3]Bus for FBD channels 0-3, master devices)

2.2.5 MCH Bandwidth Summary

Table 2-2. Intel® 7300 Chipset MCH System Bandwidth Summary

Interface	Bandwidth
Front Side Bus (1066 MHz)	8.5 GB/s per bus (34 GB/s total)
FBD (3.2 GHz)	25.5 GB/s total (17 GB/s read, 8.5 GB/s write)
FBD (4.0 GHz)	32.0 GB/s total (21.3 GB/s read, 10.7 GB/s write)
PCI Express (x8 link)	4 GB/s bidirectional
PCI Express and ESI (x4 link)	2 GB/s bidirectional

2.3 Intel® 631xESB/632xESB I/O Controller Hub Features

The Intel® 631xESB/632xESB I/O Controller Hub provides widely integrated I/O functionality. The Intel® 631xESB/632xESB I/O Controller Hub component provides the data buffering and interface arbitration required to ensure that system interfaces operate efficiently and provide the bandwidth necessary to enable the system to obtain peak performance.

Through the use of the integrated LAN functions, the Intel® 631xESB/632xESB I/O Controller Hub also supports the Alert Standard Format (ASF) for remote management.

The Intel® 631xESB/632xESB I/O Controller Hub provides extensive I/O support. Functions and capabilities include:

- Dual interface to Memory Controller
  - Enterprise South Bridge Interface (ESI): 1 GB/s each direction, full duplex, transparent to software
  - x4/x8 PCI Express interface
- Supports *PCI Express Base Specification*, Revision 1.0a
  - Four PCI Express root ports can be statically configured as four x1 or one x4; Module based Hot-Plug supported on these ports
  - Two x4 PCI Express downstream ports, connector based Hot-Plug supported on these ports
- PCI/PCI-X* Bus Interface
  - *PCI Local Bus Specification, Revision 2.3* compliant
  - Six REQ/GNT pairs
  - *PCI-X Addendum to the PCI Local Bus Specification*, Revision 1.0b compliant
  - Decoupled operation from PCI Express interface
  - 64-bit addressing for inbound and outbound transactions
  - Supports outbound PxLOCK# cycles
  - Fast Back-to-Back capable as target
  - Bus parking on last PCI agent or Intel 6700PXH 64-bit PCI Hub
  - Up to four active and four pending inbound delayed transactions, and two outbound delayed transactions per interface
- Hot-Plug Controller
  - *PCI Standard Hot-Plug Controller and Subsystem Specification*, Revision 1.0 compliant
  - Serial support for three to six slot Hot-Plug systems
  - Parallel support for one- and two-slot Hot-Plug systems
  - One-slot-no-glue Hot-Plug mode
- *PCI Local Bus Specification*, Revision 2.3-compliant PCI bus interface with support for 33 MHz PCI operations
  - Supports up to seven Req/Gnt pairs
- Kumeran integrated Gigabit LAN controller and PHY interface
  - Reduced SGMII interface running at 1.25 Gb/s
- ACPI Power Management Logic Support

- Enhanced DMA controller, interrupt controller, and timer functions
- Integrated IDE controller supports Ultra ATA100/66/33
- USB host interface with support for eight USB ports; four UHCI host controllers; one EHCI high speed USB 2.0 Host controller
- *System Management Bus (SMBus) Specification*, Version 2.0-compliant with additional support for I^2C devices
- Low Pin Count (LPC) interface
- Firmware Hub (FWH) interface support
- Alert On LAN* (AOL) and Alert On LAN 2* (AOL2)
- Support for two IOAPICs
- Test/Debug
  - JTAG TAP access port compliant with *IEEE Standard Test Access Port and Boundary Scan Architecture 1149.1a*
- Full read/write access to all configuration and memory registers
- PCI Express and PCI Power Management

2.4 Additional Platform Components

2.4.1 Intel® 6700PXH/6702PXH 64-bit PCI Hub System Features

The Intel® 6700PXH 64-bit PCI Hub is a peripheral component that performs PCI bridging functions between the PCI Express* interface and the PCI Bus. The Intel 6700PXH 64-bit PCI Hub contains two PCI bus interfaces that can be independently configured to operate in PCI (33 or 66 MHz) or PCI-X Mode 1 (66, 100, or 133 MHz), for either 32 or 64 bit PCI devices. The Intel 6700PXH 64-bit PCI Hub further supports the new PCI Standard Hot-Plug Controller and Subsystem Specification Revision 1.0. Each PCI interface contains an I/OxAPIC with 24 interrupts and a standard hot plug controller.

Design details for the PXH are contained in a separate document titled *Intel® 6700PXH 64-bit PCI Hub Design Guide*.

The Intel 6700PXH/6702PXH 64-bit PCI Hub supports the following features:

- PCI Express Interface support
  - *PCI Express Base Specification*, Revision 1.0a compatible
  - 64-bit addressing support
  - 32-bit CRC (Cyclical Redundancy Checking) covering all transmitted data packets
  - 16-bit CRC on all link message information
- PCI/PCI-X Bus Interfaces (two segments for Intel 6700PXH 64-bit PCI Hub, one segment for Intel® 6702PXH 64-bit PCI Hub,)
  - *PCI Local Bus Specification, Revision 2.3* compliant
  - Full peer-to-peer read/write capability between the two PCI segments
  - Six REQ/GNT pairs per PCI bus segment
  - *PCI-X Addendum to the PCI Local Bus Specification*, Revision 1.0b-compliant

  - Decoupled operation from PCI Express interface
  - 64-bit addressing for inbound and outbound transactions
  - Supports outbound PxLOCK# cycles
  - Fast Back-to-Back capable as target
  - Bus parking on last PCI agent or Intel 6700PXH 64-bit PCI Hub
  - Up to four active and four pending inbound delayed transactions, and two outbound delayed transactions per interface
- Hot-Plug Controllers (two per Intel 6700PXH 64-bit PCI Hub, one per Intel 6702PXH 64-bit PCI Hub)
  - *PCI Standard Hot-Plug Controller and Subsystem Specification*, Revision 1.0 compliant
  - One interface per PCI bus segment
  - Serial support for three to six slot Hot-Plug systems
  - Parallel support for one- and two-slot Hot-Plug systems
  - One-slot-no-glue Hot-Plug mode
- IOapic (2 per Intel 6700PXH 64-bit PCI Hub, one per Intel 6702PXH 64-bit PCI Hub)
  - One interface per PCI bus segment
  - Supports up to twenty-four interrupts (16 pins) per PCI interface in server mode
  - Compatible with both Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series and Dunnington processors
  - Boot interrupt output
- Test/Debug
  - JTAG TAP access port compliant with *IEEE Standard Test Access Port and Boundary Scan Architecture 1149.1a*
- Miscellaneous
  - ACPI support with in-band PCI Express interface messaging
  - Full read/write access to all configuration and memory registers
  - *System Management Bus 2.0* compliant
  - PCI Express and PCI Power Management

2.4.2 Intel® IOP 348 I/O Processor Storage I/O Processor Features

Intel® IOP 348 I/O processor is an I/O storage processor that integrates two Intel XScale® cores with intelligent peripherals including a PCI bus application bridge and eight Serial-Attached SCSI (SAS) Engines. The Intel IOP 348 also supports two internal busses: North XSI bus and South XSI bus. With the two internal busses, transactions can take place simultaneously on each bus. The north XSI bus provides the two Intel XScale cores with low latency access to either the DDR SDRAM Memory Controller, the on-chip SRAM Memory Controller, or the SAS Engines control registers. Peripherals that generate large burst transactions are located on the south XSI bus, thus allowing the two Intel XScale cores exclusive access to the north XSI bus.

The Intel IOP 348 I/O storage processor consolidates the following features into a single system:

- Two Intel XScale cores running at speeds up to 1.2 GHz
- Eight Serial-Attached SCSI Links or Eight Serial ATA links
- PCI Express x8 interface
- PCI - Local Memory Bus Address Translation Unit
- Messaging Unit, function 0 programming interface
- Third Party Messaging Interface (TPMI)
- Application Direct Memory Access (DMA) Controllers
- Transport DMA Controllers
- Peripheral Bus Interface Unit
- Integrated DDR2 Memory Controller
- Integrated SRAM Memory Controller
- Performance Monitor
- Application Accelerator
- Two Programmable Timers per Intel XScale core
- Watchdog Timer per Intel XScale core
- Three I^2C Bus Interface Units
- Two Serial Port Units
- Eight General Purpose Input Output (GPIO) ports
- Sixteen General Purpose Output ports – two per SAS Engine
- Internal North Bus—South Bus Bridge

2.4.3 82563EB/82564EB Dual-Port 10/100/1000BASE-T PHY

The 82563EB/82564EB is a dual-port 10/100/1000BASE-T PHY for use with the MAC integrated into the Intel® 631xESB/632xESB I/O Controller Hub.

Design details for the 82563EB/82564EB Gigabit Ethernet Physical Layer Device are contained in a separate document titled the *82563EB/82564EB LAN on Motherboard Design Guide*

The 82563EB/82564EB supports the following features:

- Kumeran Interface to Intel® 631xESB/632xESB I/O Controller Hub MAC
  - — High-speed serial interface
  - — Two differential pairs - only four pins/port
  - — Embedded Tx/Rx clock
  - — In-band MDIO for register control
- Power: ~1 W per port (~2.0 W total) Typical
- Package: 100-lead, 14x14 mm, Quad Flat Pak, 0.5 mm lead-pitch
- Cable Length: 150 m (standard)
- Other features:
  - — Remote diagnostics & unified SW drivers

— Auto link speed detection

2.4.4 82575 Dual Port 10/100/1000 Gigabit Ethernet Controller

The 82575 is a Dual Port 10/100/1000 x4 PCI Express* Gigabit Ethernet Controller for Server and Embedded Platforms

Design details for the 82575EA/EB/ES Gigabit Ethernet Controller Device are contained in a separate document titled the *82575EA/EB/ES Gigabit Ethernet Controller Design Guide*

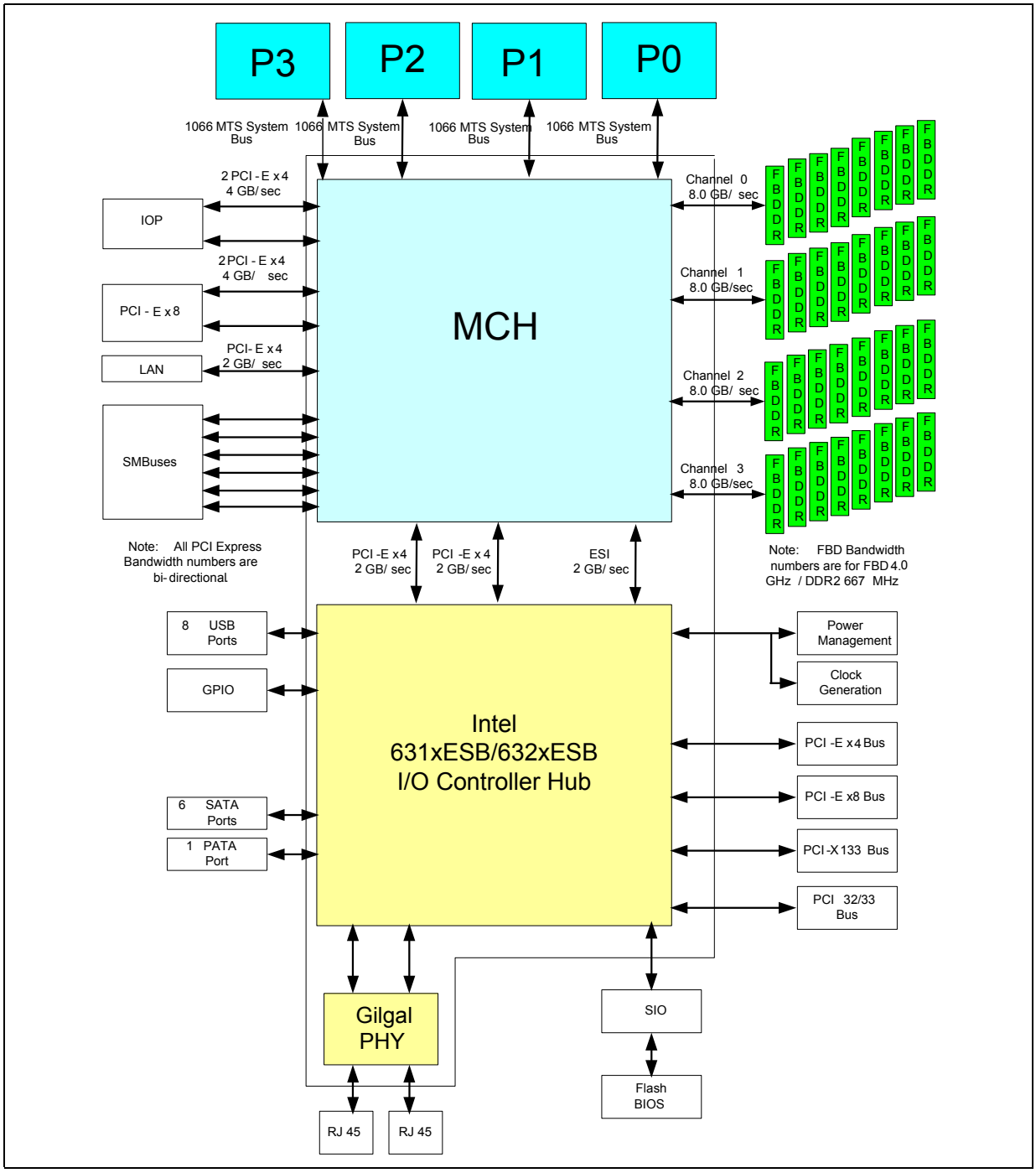
The 82575 supports the following features:

- PCIe* x2 or x4
- Intel® I/O Acceleration Technology
  - — TCP Acceleration
  - — RDMA Acceleration
  - — iSCSI CRC Acceleration
- Advanced Manageability
  - — Intel® Active Management Technology
  - — RMII or SMBus/FML Interfaces
  - — IPMI BMC Pass-thru
- ECC on all memory
- Dual SerDes/SGMII interface
- MSI-X support
- Multiple Tx & Rx Queues
- Offloads compatible with multiple VLAN tags
- 25mm x 25mm FCBGA
- ~3.2W total power consumption (2 port operation)

2.5 System Configuration

Figure 2-1 illustrates a typical system configuration.

Figure 2-1. Typical System Configuration



§

3 Signal Description

This section provides a detailed description of MCH signals. The signals are arranged in functional groups according to their associated interface. Throughout this section the following conventions are used:

The terms *assertion* and *deassertion* are to avoid confusion when working with a mix of active-high and active-low signals. The terms *assert*, or *assertion*, indicates that the signal is active, independent of whether the active level is represented by a high or low voltage. The terms *deassert*, or *deassertion*, indicates that the signal is inactive.

Signal names may or may not have a "_N" appended to them. The "_N" symbol at the end of a signal name indicates that the active, or asserted state occurs when the signal is at a low voltage level. When "_N" is not present after the signal name, the signal is asserted when at the high voltage level.

Differential signal pairs adopt a "{P/N}" suffix to indicate the "positive" (P) or "negative" (N) signal in the pair. If a "_N" is appended, it is appended to the positive and negative signals in a pair.

Typical frequencies of operation for the fastest operating modes are indicated. No frequency is specified for asynchronous or analog signals.

Some signals or groups of signals have multiple versions. These signal groups may represent distinct but similar ports or interfaces, or may represent identical copies of the signal used to reduce loading effects.

Curly-bracketed non-trailing numerical indices, e.g. "{X/Y}", represent replications of major buses. Square-bracketed numerical indices, e.g., "[n:m]" represent functionally similar but logically distinct bus signals; each signal provides an independent control, and may or may not be asserted at the same time as the other signals in the grouping. In contrast, trailing curly-bracketed numerical indices, e.g., "{x/y}" typically represent identical duplicates of a signal; such duplicates are provided for electrical reasons.

The following notations are used to describe the signal type:

I Input pin

O Output pin

I/O Bi-directional Input/Output pin

The signal description also includes the type of buffer used for the particular signal:

Table 3-1. Signal Buffer Types

Buffer Type	Description
AGTL+	Open Drain AGTL+ interface signal. Refer to the AGTL+ I/O Specification for complete details. The MCH integrates AGTL+termination resistors, and supports VTT from 1.15 V to 1.55 V.
PCIEX	Current-mode 2.5 GHz PCI Express signaling
SMBus	3.3V 100 KHz SMBus Open Drain output with Schmidt trigger input
CMOS	1.1V 200 MHz CMOS totem-pole output with Schmidt trigger input
JTAG	1.1V 20 MHz CMOS open-drain output with Schmidt trigger input

Host Interface signals that perform multiple transfers per clock cycle may be marked as either "4X" (for signals that are "quad-pumped") or 2X (for signals that are "doublepumped").

Note: Processor address and data bus signals are logically inverted signals. In other words, the actual values are inverted of what appears on the processor bus. This must be taken into account and the addresses and data bus signals must be inverted inside the MCH host bridge. All processor control signals follow normal convention. A 0 indicates an active level (low voltage) if the signal is followed by _N symbol and a 1 indicates an active level (high voltage) if the signal has no _N suffix.

Table 3-2. Signal Naming Conventions

Convention	Expands to
RR{0/1/2}XX	Expands to: RR0XX, RR1XX, and RR2XX. This denotes similar signals on replicated buses.
RR[2:0]	Expands to: RR[2], RR[1], and RR[0]. This denotes a bus.
RR{0/1/2}	Expands to: RR2, RR1, and RR0. This denotes electrical duplicates.
RR_N or RR_N[2:0]	Denotes an active low signal or bus.

3.1 Processor Front Side Bus Signals

3.1.1 Processor Front Side Bus 0

Signal Name	Type	Description
FSB0A_N[39:3]	AGTL I/O	**Processor 0 Address Bus:** FSB0A_N[39:3] connect to the processor address bus. During processor cycles, FSB0A_N[39:3] are inputs. The MCH drives FSB0A_N[39:3] during snoop cycles on behalf of ESI and Secondary PCI initiators. FSB0A_N[39:3] are transferred at 2X rate. Note that the address is inverted on the processor bus. FSB0A_N[25,12:9,3] are used for Power ON Configuration of the processor during RESET_N deassertion. Refer to Section 4.8.6.3, "POC: Power-On Configuration" on page 107
FSB0ADS_N	AGTL I/O	**Processor 0 Address Strobe:** The processor bus owner asserts FSB0ADS_N to indicate the first of two cycles of a request phase. The MCH can assert this signal for snoop cycles and interrupt messages.
FSB0ADSTB_N[1:0]	AGTL I/O	**Processor 0 Address Strobe:** FSB0ADSTB_N[1:0] are source synchronous strobes used to transfer FSB0A_N[39:3] and FSB0REQ_N[4:0] at the 2X transfer rate. **StrobeAddress Bits** FSB0ADSTB_N[0] FSB0A_N[37:36, 16:3], FSB0REQ_N[4:0] FSB0ADSTB_N[1] FSB0A_N[39:38, 35:17]
FSB0AP_N[1:0]	AGTL I/O	**Processor 0 Address Parity:** FSB0AP_N[1:0] provide parity protection on the address bus
FSB0BINIT_N	AGTL I/O	**Processor 0 Bus Initialization:** This signal causes a reset of the bus state machines.
FSB0BNR_N	AGTL I/O	**Processor 0 Block Next Request:** This signal is used to block the current request bus owner from issuing a new request. This signal is used to dynamically control the processor bus pipeline depth.
FSB0BPM[5:4]	AGTL I/O	**Breakpoint /Debug Bus:** These signals are breakpoint and performance monitor signals. These are output from the processor to indicate the status of breakpoints and programmable counters used for monitoring processor performance.
FSB0BPRI_N	AGTL O	**Processor 0 Priority Agent Bus Request:** The MCH is the only Priority Agent on the processor bus. It asserts this signal to obtain ownership of the address bus. This signal has priority over symmetric bus requests and cause the current symmetric owner to stop issuing new transactions unless the FSB0LOCK_N signal was asserted.

Signal Name	Type	Description
FSB0BREQ_N[1:0]	AGTL I/O	**Processor 0 Bus Requests**: The MCH pulls the FSB0BREQ_N[0] signal low during RESET_N. The signal is sampled by the processor on the active-to-inactive transition of FSB0RESET_N. The minimum setup time for this signal is 4 FSB0CLKs. The minimum hold time is 2 clocks and the maximum hold time is 20 FSB0CLKs.
FSB0D_N[63:0]	AGTL I/O	**Processor 0 Data Bus:** These signals are connected to the processor data bus. Data on FSB0D_N[63:0] is transferred at a 4X rate. Note that the data signals may be inverted on the processor bus, depending on the **P0DBI**[3:0] signals.
FSB0DBI_N[3:0]	AGTL I/O	**Processor 0 Dynamic Bus Inversion:** These signals are driven along with the FSB0D_N[63:0] signals. They indicate if the associated signals are inverted. FSB0DBI_N[3:0] are asserted such that the number of data bits driven electrically low (low voFSB0ltage) within the corresponding 16-bit group never exceeds 8. **FSB0DBI_N[x]Data Bits** FSB0DBI_N[3]FSB0D_N[63:48] FSB0DBI_N[2]FSB0D_N[47:32] FSB0DBI_N[1]FSB0D_N[31:16] FSB0DBI_N[0]FSB0D_N[15:0]
FSB0DBSY_N	AGTL I/O	**Processor 0 Data Bus Busy:** This signal is used by the data bus owner to hold the data bus for transfers requiring more than one cycle.
FSB0DEFER_N	AGTL O	**Processor 0 Data Bus Defer:** Defer indicates that the MCH will terminate the transaction currently being snooped with either a deferred response or with a retry response.
FSB0DP[3:0]_N	AGTL I/O	**Processor 0 Data Bus Parity:** FSB0DP_N[3:0] provide parity protection on the data bus.
FSB0DRDY_N	AGTL I/O	**Processor 0 Data Ready:** This signal is asserted for each cycle that data is transferred.
FSB0HIT_N	AGTL I/O	**Processor 0 Cache Hit:** This signal indicates that a caching agent holds an unmodified version of the requested line. FSB0HIT_N is also driven in conjunction with FSB0HITM_N by the target to extend the snoop window.
FSB0HITM_N	AGTL I/O	**Processor 0 Cache Hit Modified:** This signal indicates that a caching agent holds a modified version of the requested line and that this agent assumes responsibility for providing the line. FSB0HITM_N is also driven in conjunction with FSB0HIT_N to extend the snoop window.
FSB0LOCK_N	AGTL I	**Processor 0 Lock:** This signal indicates to the system that a transaction must occur atomically. This signal must connect the appropriate pins of all processor FSB agents. For a locked sequence of transactions, LOCK_N is asserted from the beginning of the first transaction to the end of the last transaction. When the priority agent asserts BPRI_N to arbitrate for ownership of the processor FSB, it will wait until it observes LOCK_N deasserted. This enables symmetric agents to retain ownership of the processor FSB throughout the bus locked operation and ensure the atomicity of lock.
FSB0MCERR_N	AGTL I/O	**Processor 0 Machine Check Error:** Machine check error
FSB0REQ_N[4:0]	AGTL I/O	**Processor Bus 0 Request Command:** These signals define the attributes of the request. FSB0REQ_N[4:0] are transferred at 2X rate. They are asserted by the requesting agent during both halves of request phase. In the first half the signals define the transaction type to a level of detail that is sufficient to begin a snoop request. In the second half the signals carry additional information to define the complete transaction type.
FSB0RESET_N	AGTL O	**Processor 0 Reset:** The FSB0RESET_N pin is an output from the MCH. The MCH asserts FSB0RESET_N while RESET_N(PCIRST_N from Intel® 631xESB/632xESB I/O Controller Hub) is asserted and for approximately 1 ms after RESET_N is deasserted. The FSB0RESET_N allows the processors to begin execution in a known state.

Signal Name	Type	Description
FSB0RS_N[2:0]	AGTL O	**Processor 0 Response Status Signals:** These signals indicate the type of response according to the following: **Encoding Response Type** 000 Idle state 001 Retry response 010 Deferred response 011 Reserved (not driven by MCH) 100 Hard Failure (not driven by MCH) 101 No data response 110 Implicit Writeback 111 Normal data response
FSB0RSP_N	AGTL O	**Processor 0 Response Status Parity:**
FSB0STBP_N[3:0] FSB0STBN_N[3:0]	AGTL I/O	**Processor 0 Differential Host Data Strobes:** The differential source synchronous strobes used to transfer FSB0D_N[63:0] and FSB0DBI_N[3:0] at the 4X transfer rate. **Strobe Data Bits** FSB0DSTBP_N[3], FSB0DSTBN_N[3] FSB0D_N[63:48], FSB0DBI_N[3] FSB0DSTBP_N[2], FSB0DSTBN_N[2] FSB0D_N[47:32], FSB0DBI_N[2] FSB0DSTBP_N[1], FSB0DSTBN_N[1] FSB0D_N[31:16], FSB0DBI_N[1] FSB0DSTBP_N[0], FSB0DSTBN_N[0] FSB0D_N[15:0], FSB0DBI_N[0]
FSB0TRDY_N	AGTL O	**Processor Bus 0 Target Ready:** This signal indicates that the target of the processor transaction is able to enter the data transfer phase.
FSB0VREF[4,2,0]	Analog	**Processor 0 Voltage Reference:** Processor 0 voltage reference.

3.1.2 Processor Front Side Bus 1

Signal Name	Type	Description
FSB1A_N[39:3]	AGTL I/O	**Processor 1 Address Bus:** FSB1A_N[39:3] connect to the processor address bus. During processor cycles, FSB1A_N[39:3] are inputs. The MCH drives FSB1A_N[39:3] during snoop cycles on behalf of ESI and Secondary PCI initiators. FSB1A_N[39:3] are transferred at 2X rate. Note that the address is inverted on the processor bus. FSB1A_N[25,12:9,3] are used for Power ON Configuration of the processor during RESET_N deassertion. Refer to Section 4.8.6.3, "POC: Power-On Configuration" on page 107
FSB1ADS_N	AGTL I/O	**Processor 1 Address Strobe:** The processor bus owner asserts FSB1ADS_N to indicate the first of two cycles of a request phase. The MCH can assert this signal for snoop cycles and interrupt messages.
FSB1ADSTB_N[1:0]	AGTL I/O	**Processor 1 Address Strobe:** FSB1ADSTB_N[1:0] are source synchronous strobes used to transfer FSB1A_N[39:3] and FSB1REQ_N[4:0] at the 2X transfer rate. **StrobeAddress Bits** FSB1ADSTB_N[0] FSB1A_N[37:36, 16:3], FSB1REQ_N[4:0] FSB1ADSTB_N[1] FSB1A_N[39:38, 35:17]
FSB1AP_N[1:0]	AGTL I/O	**Processor 1 Address Parity:** FSB0AP_N[1:0] provide parity protection on the address bus
FSB1BINIT_N	AGTL I/O	**Processor 1 Bus Initialization:** This signal causes a reset of the bus state machines.
FSB1BNR_N	AGTL I/O	**Processor 1 Block Next Request:** This signal is used to block the current request bus owner from issuing a new request. This signal is used to dynamically control the processor bus pipeline depth.
FSB1BPM[5:4]	AGTL I/O	**Breakpoint /Debug Bus:** These signals are breakpoint and performance monitor signals. These are output from the processor to indicate the status of breakpoints and programmable counters used for monitoring processor performance.

Signal Name	Type	Description
FSB1BPRI_N	AGTL O	**Processor 1 Priority Agent Bus Request:** The MCH is the only Priority Agent on the processor bus. It asserts this signal to obtain ownership of the address bus. This signal has priority over symmetric bus requests and cause the current symmetric owner to stop issuing new transactions unless the FSB1LOCK_N signal was asserted.
FSB1BREQ_N[1:0]	AGTL I/O	**Processor 1 Bus Requests:** The MCH pulls the FSB1BREQ_N[0] signal low during RESET_N. The signal is sampled by the processor on the active-to-inactive transition of FSB1RESET_N. The minimum setup time for this signal is 4 FSB1CLKs. The minimum hold time is 2 clocks and the maximum hold time is 20 FSB1CLKs.
FSB1D_N[63:0]	AGTL I/O	**Processor 1 Data Bus:** These signals are connected to the processor data bus. Data on FSB1D_N[63:0] is transferred at a 4X rate. Note that the data signals may be inverted on the processor bus, depending on the FSB1DBI[3:0] signals.
FSB1DBI_N[3:0]	AGTL I/O	**Processor 1 Dynamic Bus Inversion:** These signals are driven along with the FSB1D_N[63:0] signals. They indicate if the associated signals are inverted. FSB1DBI_N[3:0] are asserted such that the number of data bits driven electrically low (low voltage) within the corresponding 16-bit group never exceeds 8. **FSB1DBI_N[x]Data Bits** FSB1DBI_N[3]FSB1D_N[63:48] FSB1DBI_N[2]FSB1D_N[47:32] FSB1DBI_N[1]FSB1D_N[31:16] FSB1DBI_N[0]FSB1D_N[15:0]
FSB1DBSY_N	AGTL I/O	**Processor 1 Data Bus Busy:** This signal is used by the data bus owner to hold the data bus for transfers requiring more than one cycle.
FSB1DEFER_N	AGTL O	**Processor 1 Data Bus Defer:** Defer indicates that the MCH will terminate the transaction currently being snooped with either a deferred response or with a retry response.
FSB1DP_N[3:0]	AGTL I/O	**Processor 1 Data Bus Parity:** FSB1DP_N[3:0] provide parity protection on the data bus.
FSB1DRDY_N	AGTL I/O	**Processor 1 Data Ready:** This signal is asserted for each cycle that data is transferred.
FSB1HIT_N	AGTL I/O	**Processor 1 Cache Hit:** This signal indicates that a caching agent holds an unmodified version of the requested line. FSB1HIT_N is also driven in conjunction with FSB1HITM_N by the target to extend the snoop window.
FSB1HITM_N	AGTL I/O	**Processor 1 Cache Hit Modified:** This signal indicates that a caching agent holds a modified version of the requested line and that this agent assumes responsibility for providing the line. FSB1HITM_N is also driven in conjunction with FSB1HIT_N to extend the snoop window.
FSB1LOCK_N	AGTL I	**Processor 1 Lock:** This signal indicates to the system that a transaction must occur atomically. This signal must connect the appropriate pins of all processor FSB agents. For a locked sequence of transactions, LOCK_N is asserted from the beginning of the first transaction to the end of the last transaction. When the priority agent asserts BPRI_N to arbitrate for ownership of the processor FSB, it will wait until it observes LOCK_N deasserted. This enables symmetric agents to retain ownership of the processor FSB throughout the bus locked operation and ensure the atomicity of lock.
FSB1MCERR_N	AGTL I/O	**Processor 1 Machine Check Error:** Machine check error
FSB1REQ_N[4:0]	AGTL I/O	**Processor Bus 1 Request Command:** These signals define the attributes of the request. FSB1REQ_N[4:0] are transferred at 2X rate. They are asserted by the requesting agent during both halves of request phase. In the first half the signals define the transaction type to a level of detail that is sufficient to begin a snoop request. In the second half the signals carry additional information to define the complete transaction type.
FSB1RESET_N	AGTL O	**Processor 1 Reset:** The FSB1RESET_N pin is an output from the MCH. The MCH asserts FSB1RESET_N while RESET_N(PCIRST_N from Intel® 631xESB/632xESB I/O Controller Hub) is asserted and for approximately 1 ms after RESET_N is deasserted. The FSB1RESET_N allows the processors to begin execution in a known state.

Signal Name	Type	Description
FSB1RS_N[2:0]	AGTL O	**Processor 1 Response Status Signals:** These signals indicates type of response according to the following: **EncodingResponse Type** 000 Idle state 001 Retry response 010 Deferred response 011 Reserved (not driven by MCH) 100 Hard Failure (not driven by MCH) 101 No data response 110 Implicit Writeback 111 Normal data response
FSB1RSP_N	AGTL O	Processor 1 Response Status Parity:
FSB1STBP_N[3:0] FSB1STBN_N[3:0]	AGTL I/O	**Processor 1 Differential Host Data Strobes:** The differential source synchronous strobes used to transfer FSB1D_N[63:0] and FSB1DBI_N[3:0] at the 4X transfer rate. **Strobe Data Bits** FSB1DSTBP_N[3], FSB1DSTBN_N[3] FSB1D_N[63:48], FSB1DBI_N[3] FSB1DSTBP_N[2], FSB1DSTBN_N[2] FSB1D_N[47:32], FSB1DBI_N[2] FSB1DSTBP_N[1], FSB1DSTBN_N[1] FSB1D_N[31:16], FSB1DBI_N[1] FSB1DSTBP_N[0], FSB1DSTBN_N[0] FSB1D_N[15:0], FSB1DBI_N[0]
FSB1TRDY_N	AGTL O	**Processor Bus 1 Target Ready:** This signal indicates that the target of the processor transaction is able to enter the data transfer phase.
FSB1VREF[4,2,0]	Analog	**Processor 1 Voltage Reference:** Processor 1 voltage reference.

3.1.3 Processor Front Side Bus 2

Signal Name	Type	Description
FSB2A_N[39:3]	AGTL I/O	**Processor 2 Address Bus:** FSB2A_N[39:3] connect to the processor address bus. During processor cycles, FSB2A_N[39:3] are inputs. The MCH drives FSB2A_N[39:3] during snoop cycles on behalf of ESI and Secondary PCI initiators. FSB2A_N[39:3] are transferred at 2X rate. Note that the address is inverted on the processor bus. FSB2A_N[25,12:9,3] are used for Power ON Configuration of the processor during RESET_N deassertion. Refer to Section 4.8.6.3, "POC: Power-On Configuration" on page 107
FSB2ADS_N	AGTL I/O	**Processor 2 Address Strobe:** The processor bus owner asserts FSB2ADS_N to indicate the first of two cycles of a request phase. The MCH can assert this signal for snoop cycles and interrupt messages.
FSB2ADSTB_N[1:0]	AGTL I/O	**Processor 2 Address Strobe:** FSB2ADSTB_N[1:0] are source synchronous strobes used to transfer FSB2A_N[39:3] and FSB2REQ_N[4:0] at the 2X transfer rate. **Strobe Address Bits** FSB2ADSTB_N[0] FSB2A_N[37:36, 16:3], FSB2REQ_N[4:0] FSB2ADSTB_N[1] FSB2A_N[39:38, 35:17]
FSB2AP_N[1:0]	AGTL I/O	**Processor 2 Address Parity:** FSB2AP_N[1:0] provide parity protection on the address bus
FSB2BINIT_N	AGTL I/O	**Processor 2 Bus Initialization:** This signal causes a reset of the bus state machines.
FSB2BNR_N	AGTL I/O	**Processor 2 Block Next Request:** This signal is used to block the current request bus owner from issuing a new request. This signal is used to dynamically control the processor bus pipeline depth.
FSB2BPM[5:4]	AGTL I/O	**Breakpoint /Debug Bus:** These signals are breakpoint and performance monitor signals. These are output from the processor to indicate the status of breakpoints and programmable counters used for monitoring processor performance.

Signal Name	Type	Description
FSB2BPRI_N	AGTL O	**Processor 2 Priority Agent Bus Request:** The MCH is the only Priority Agent on the processor bus. It asserts this signal to obtain ownership of the address bus. This signal has priority over symmetric bus requests and cause the current symmetric owner to stop issuing new transactions unless the FSB2LOCK_N signal was asserted.
FSB2BREQ_N[1:0]	AGTL I/O	**Processor 2 Bus Requests**: The MCH pulls the FSB2BREQ_N[0] signal low during RESET_N. The signal is sampled by the processor on the active-to-inactive transition of FSB2RESET_N. The minimum setup time for this signal is 4 FSB2CLKs. The minimum hold time is 2 clocks and the maximum hold time is 20 FSB2CLKs.
FSB2D_N[63:0]	AGTL I/O	**Processor 2 Data Bus:** These signals are connected to the processor data bus. Data on FSB2D_N[63:0] is transferred at a 4X rate. Note that the data signals may be inverted on the processor bus, depending on the **P0DBI**[3:0] signals.
FSB2DBI_N[3:0]	AGTL I/O	**Processor 2 Dynamic Bus Inversion:** These signals are driven along with the FSB2D_N[63:0] signals. They indicate if the associated signals are inverted. FSB2DBI_N[3:0] are asserted such that the number of data bits driven electrically low (low voFSB2ltage) within the corresponding 16-bit group never exceeds 8. **FSB2DBI_N[x] Data Bits** FSB2DBI_N[3] FSB2D_N[63:48] FSB2DBI_N[2] FSB2D_N[47:32] FSB2DBI_N[1] FSB2D_N[31:16] FSB2DBI_N[0] FSB2D_N[15:0]
FSB2DBSY_N	AGTL I/O	**Processor 2 Data Bus Busy:** This signal is used by the data bus owner to hold the data bus for transfers requiring more than one cycle.
FSB2DEFER_N	AGTL O	**Processor 2 Data Bus Defer:** Defer indicates that the MCH will terminate the transaction currently being snooped with either a deferred response or with a retry response.
FSB2DP_N[3:0]	AGTL I/O	**Processor 2 Data Bus Parity:** FSB2DP_N[3:0] provide parity protection on the data bus.
FSB2DRDY_N	AGTL I/O	**Processor 2 Data Ready:** This signal is asserted for each cycle that data is transferred.
FSB2HIT_N	AGTL I/O	**Processor 2 Cache Hit:** This signal indicates that a caching agent holds an unmodified version of the requested line. FSB2HIT_N is also driven in conjunction with FSB2HITM_N by the target to extend the snoop window.
FSB2HITM_N	AGTL I/O	**Processor 2 Cache Hit Modified:** This signal indicates that a caching agent holds a modified version of the requested line and that this agent assumes responsibility for providing the line. FSB2HITM_N is also driven in conjunction with FSB2HIT_N to extend the snoop window.
FSB2LOCK_N	AGTL I	**Processor 2 Lock:** This signal indicates to the system that a transaction must occur atomically. This signal must connect the appropriate pins of all processor FSB agents. For a locked sequence of transactions, LOCK_N is asserted from the beginning of the first transaction to the end of the last transaction. When the priority agent asserts BPRI_N to arbitrate for ownership of the processor FSB, it will wait until it observes LOCK_N deasserted. This enables symmetric agents to retain ownership of the processor FSB throughout the bus locked operation and ensure the atomicity of lock.
FSB2MCERR_N	AGTL I/O	**Processor 2 Machine Check Error:** Machine check error
FSB2REQ_N[4:0]	AGTL I/O	**Processor Bus 0 Request Command:** These signals define the attributes of the request. FSB2REQ_N[4:0] are transferred at 2X rate. They are asserted by the requesting agent during both halves of request phase. In the first half the signals define the transaction type to a level of detail that is sufficient to begin a snoop request. In the second half the signals carry additional information to define the complete transaction type.
FSB2RESET_N	AGTL O	**Processor 2 Reset:** The FSB2RESET_N pin is an output from the MCH. The MCH asserts FSB2RESET_N while RESET_N(PCIRST_N from Intel® 631xESB/632xESB I/O Controller Hub) is asserted and for approximately 1 ms after RESET_N is deasserted. The FSB2RESET_N allows the processors to begin execution in a known state.

Signal Name	Type	Description
FSB2RS_N[2:0]	AGTL O	**Processor 2 Response Status Signals:** These signals indicate the type of response according to the following: **Encoding Response Type** 000 Idle state 001 Retry response 010 Deferred response 011 Reserved (not driven by MCH) 100 Hard Failure (not driven by MCH) 101 No data response 110 Implicit Writeback 111 Normal data response
FSB2RSP_N	AGTL O	**Processor 2 Response Status Parity:**
FSB2STBP_N[3:0] FSB2STBN_N[3:0]	AGTL I/O	**Processor 2 Differential Host Data Strobes:** The differential source synchronous strobes used to transfer FSB2D_N[63:0] and FSB2DBI_N[3:0] at the 4X transfer rate. **StrobeData Bits** FSB2DSTBP_N[3], FSB2DSTBN_N[3] FSB2D_N[63:48], FSB2DBI_N[3] FSB2DSTBP_N[2], FSB2DSTBN_N[2] FSB2D_N[47:32], FSB2DBI_N[2] FSB2DSTBP_N[1], FSB2DSTBN_N[1] FSB2D_N[31:16], FSB2DBI_N[1] FSB2DSTBP_N[0], FSB2DSTBN_N[0] FSB2D_N[15:0], FSB2DBI_N[0]
FSB2TRDY_N	OAGTL	**Processor Bus 0 Target Ready:** This signal indicates that the target of the processor transaction is able to enter the data transfer phase.
FSB2VREF[4,2,0]	Analog	**Processor 2 Voltage Reference:** Processor 2 voltage reference.

3.1.4 Processor Front Side Bus 3

Signal Name	Type	Description
FSB3A_N[39:3]	AGTL I/O	**Processor 3 Address Bus:** FSB3A_N[39:3] connect to the processor address bus. During processor cycles, FSB3A_N[39:3] are inputs. The MCH drives FSB3A_N[39:3] during snoop cycles on behalf of ESI and Secondary PCI initiators. FSB3A_N[39:3] are transferred at 2X rate. Note that the address is inverted on the processor bus. FSB3A_N[25,12:9,3] are used for Power ON Configuration of the processor during RESET_N deassertion. Refer to Section 4.8.6.3, "POC: Power-On Configuration" on page 107
FSB3ADS_N	AGTL I/O	**Processor 3 Address Strobe:** The processor bus owner asserts FSB3ADS_N to indicate the first of two cycles of a request phase. The MCH can assert this signal for snoop cycles and interrupt messages.
FSB3ADSTB_N[1:0]	AGTL I/O	**Processor 3 Address Strobe:** FSB3ADSTB_N[1:0] are source synchronous strobes used to transfer FSB3A_N[39:3] and FSB3REQ_N[4:0] at the 2X transfer rate. **Strobe Address Bits** FSB3ADSTB_N[0] FSB3A_N[37:36,16:3], FSB3REQ_N[4:0] FSB3ADSTB_N[1] FSB3A_N[39:38, 35:17]
FSB3AP_N[1:0]	AGTL I/O	**Processor 3 Address Parity:** FSB3AP_N[1:0] provide parity protection on the address bus
FSB3BINIT_N	AGTL I/O	**Processor 3 Bus Initialization:** This signal causes a reset of the bus state machines.
FSB3BNR_N	AGTL I/O	**Processor 3 Block Next Request:** This signal is used to block the current request bus owner from issuing a new request. This signal is used to dynamically control the processor bus pipeline depth.
FSB3BPM[5:4]	AGTL I/O	**Breakpoint /Debug Bus:** These signals are breakpoint and performance monitor signals. These are output from the processor to indicate the status of breakpoints and programmable counters used for monitoring processor performance.

Signal Name	Type	Description
FSB3BPRI_N	AGTL O	**Processor 3 Priority Agent Bus Request:** The MCH is the only Priority Agent on the processor bus. It asserts this signal to obtain ownership of the address bus. This signal has priority over symmetric bus requests and cause the current symmetric owner to stop issuing new transactions unless the FSB3LOCK_N signal was asserted.
FSB3BREQ_N[1:0]	AGTL I/O	**Processor 3 Bus Requests**: The MCH pulls the FSB3BREQ_N[0] signal low during RESET_N. The signal is sampled by the processor on the active-to-inactive transition of FSB3RESET_N. The minimum setup time for this signal is 4 FSB3CLKs. The minimum hold time is 2 clocks and the maximum hold time is 20 FSB3CLKs.
FSB3D_N[63:0]	AGTL I/O	**Processor 3 Data Bus:** These signals are connected to the processor data bus. Data on FSB3D_N[63:0] is transferred at a 4X rate. Note that the data signals may be inverted on the processor bus, depending on the **P0DBI**[3:0] signals.
FSB3DBI_N[3:0]	AGTL I/O	**Processor 3 Dynamic Bus Inversion:** These signals are driven along with the FSB3D_N[63:0] signals. They indicate if the associated signals are inverted. FSB3DBI_N[3:0] are asserted such that the number of data bits driven electrically low (low voFSB3ltage) within the corresponding 16-bit group never exceeds 8. **FSB3DBI_N[x]Data Bits** FSB3DBI_N[3]FSB3D_N[63:48] FSB3DBI_N[2]FSB3D_N[47:32] FSB3DBI_N[1]FSB3D_N[31:16] FSB3DBI_N[0]FSB3D_N[15:0]
FSB3DBSY_N	AGTL I/O	**Processor 3 Data Bus Busy:** This signal is used by the data bus owner to hold the data bus for transfers requiring more than one cycle.
FSB3DEFER_N	AGTL O	**Processor 3 Data Bus Defer:** Defer indicates that the MCH will terminate the transaction currently being snooped with either a deferred response or with a retry response.
FSB3DP_N[3:0]	AGTL I/O	**Processor 3 Data Bus Parity:** FSB3DP_N[3:0] provide parity protection on the data bus.
FSB3DRDY_N	AGTL I/O	**Processor 3 Data Ready:** This signal is asserted for each cycle that data is transferred.
FSB3HIT_N	AGTL I/O	**Processor 3 Cache Hit:** This signal indicates that a caching agent holds an unmodified version of the requested line. FSB3HIT_N is also driven in conjunction with FSB3HITM_N by the target to extend the snoop window.
FSB3HITM_N	AGTL I/O	**Processor 3 Cache Hit Modified:** This signal indicates that a caching agent holds a modified version of the requested line and that this agent assumes responsibility for providing the line. FSB3HITM_N is also driven in conjunction with FSB3HIT_N to extend the snoop window.
FSB3LOCK_N	AGTL I	**Processor 3 Lock:** This signal indicates to the system that a transaction must occur atomically. This signal must connect the appropriate pins of all processor FSB agents. For a locked sequence of transactions, LOCK_N is asserted from the beginning of the first transaction to the end of the last transaction. When the priority agent asserts BPRI_N to arbitrate for ownership of the processor FSB, it will wait until it observes LOCK_N deasserted. This enables symmetric agents to retain ownership of the processor FSB throughout the bus locked operation and ensure the atomicity of lock.
FSB3MCERR_N	AGTL I/O	**Processor 3 Machine Check Error:** Machine check error
FSB3REQ_N[4:0]	AGTL I/O	**Processor Bus 0 Request Command:** These signals define the attributes of the request. FSB3REQ_N[4:0] are transferred at 2X rate. They are asserted by the requesting agent during both halves of request phase. In the first half the signals define the transaction type to a level of detail that is sufficient to begin a snoop request. In the second half the signals carry additional information to define the complete transaction type.
FSB3RESET_N	AGTL O	**Processor 3 Reset:** The FSB3RESET_N pin is an output from the MCH. The MCH asserts FSB3RESET_N while RESET_N(PCIRST_N from Intel® 631xESB/632xESB I/O Controller Hub) is asserted and for approximately 1 ms after RESET_N is deasserted. The FSB3RESET_N allows the processors to begin execution in a known state.

Signal Name	Type	Description
FSB3RS_N[2:0]	AGTL O	**Processor 3 Response Status Signals:** These signals indicate the type of response according to the following: **Encoding Response Type** 000 Idle state 001 Retry response 010 Deferred response 011 Reserved (not driven by MCH) 100 Hard Failure (not driven by MCH) 101 No data response 110 Implicit Writeback 111 Normal data response
FSB3RSP_N	AGTL O	**Processor 3 Response Status Parity:**
FSB3STBP_N[3:0] FSB3STBN_N[3:0]	AGTL I/O	**Processor 3 Differential Host Data Strobes:** The differential source synchronous strobes used to transfer FSB3D_N[63:0] and FSB3DBI_N[3:0] at the 4X transfer rate. **StrobeData Bits** FSB3DSTBP_N[3], FSB3DSTBN_N[3] FSB3D_N[63:48], FSB3DBI_N[3] FSB3DSTBP_N[2], FSB3DSTBN_N[2] FSB3D_N[47:32], FSB3DBI_N[2] FSB3DSTBP_N[1], FSB3DSTBN_N[1] FSB3D_N[31:16], FSB3DBI_N[1] FSB3DSTBP_N[0], FSB3DSTBN_N[0] FSB3D_N[15:0], FSB3DBI_N[0]
FSB3TRDY_N	AGTL O	**Processor Bus 0 Target Ready:** This signal indicates that the target of the processor transaction is able to enter the data transfer phase.
FSB3VREF[4,2,0]	Analog	**Processor 3 Voltage Reference:** Processor 3 voltage reference.

3.2 Fully Buffered DIMM Memory Channels

3.2.1 FB-DIMM Branch 0

FB-DIMM branch 0 contains FB-DIMM channels 0 and 1. The following signals are common to both FB-DIMM channels.

Signal Name	Type	Description
FBD01BGBIASEXT	Analog	**FB-DIMM Bypass Bias Input for Band Gap Circuit:**
FBD01CLKN	Analog	**FB-DIMM Clock Negative:** Core Clock Negative Phase
FBD01CLKP	Analog	**FB-DIMM Clock Positive:** Core Clock Positive Phase
FBD01ICOMPBIAS	Analog	**FB-DIMM Transmitter Swing Bias:**
FBD01RESIN	Analog	**FB-DIMM On-die Impedance Compensation:**
FBD01VCCA	Analog	**FB-DIMM VCC:** Analog Voltage for the PLL
FBD01VSSA	Analog	**FB-DIMM VSS:** Analog Voltage for PLL

3.2.1.1 FB-DIMM Channel 0

Signal Name	Type	Description
FBD0NBIN[13:0]	I	**FB-DIMM Channel 0 Northbound Input Data Negative Phase: NOTE: FBD0NBIN[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FBD0NBIP[13:0]	I	**FB-DIMM Channel 0 Northbound Input Data Positive Phase: NOTE: FBD0NBIP[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FBD0SBON[9:0]	O	**FB-DIMM Channel 0 Southbound Output Negative Phase:**
FBD0SBOP[9:0]	O	**FB-DIMM Channel 0 Southbound Output Positive Phase:**

3.2.1.2 FB-DIMM Channel 1

Signal Name	Type	Description
FBD1NBIN[13:0]	I	**FB-DIMM Channel 1 Northbound Input Data Negative Phase: NOTE: FBD1NBIN[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FBD1NBIP[13:0]	I	**FB-DIMM Channel 1 Northbound Input Data Positive Phase: NOTE: FBD1NBIP[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FBD1SBON[9:0]	O	**FB-DIMM Channel 1 Southbound Output Negative Phase:**
FBD1SBOP[9:0]	O	**FB-DIMM Channel 1 Southbound Output Positive Phase:**

3.2.2 FB-DIMM Branch 1

FB-DIMM branch 1 contains FB-DIMM channels 2 and 3. The following signals are common to both FB-DIMM channels.

Signal Name	Type	Description
FBD23BGBIASEXT	Analog	**FB-DIMM Bypass Bias Input for Band Gap Circuit:**
FBD23ICOMPBIAS	Analog	**FB-DIMM Transmitter Swing Bias:**
FBD23RESIN	Analog	**FB-DIMM On-die Impedance Compensation:**
FBD23CLKN	Analog	**FB-DIMM Clock Negative:** Core Clock Negative Phase
FBD23CLKP	Analog	**FB-DIMM Clock Positive:** Core Clock Positive Phase
FBD23VCCA	Analog	**FB-DIMM VCC:** Analog Voltage for the PLL
FBD23VSSA	Analog	**FB-DIMM VSS:** Analog Voltage for PLL

3.2.2.1 FB-DIMM Channel 2

Signal Name	Type	Description
FBD2NBIN[13:0]	I	**FB-DIMM Channel 2 Northbound Input Data Negative Phase: NOTE: FBD2NBIN[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FBD2NBIP[13:0]	I	**FB-DIMM Channel 2 Northbound Input Data Positive Phase: NOTE: FBD2NBIP[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FBD2SBON[9:0]	O	**FB-DIMM Channel 2 Southbound Output Negative Phase:**
FBD2SBOP[9:0]	O	**FB-DIMM Channel 2 Southbound Output Positive Phase:**

3.2.2.2 FB-DIMM Channel 3

Signal Name	Type	Description
FBD3NBIN[13:0]	I	**FB-DIMM Channel 3 Northbound Input Data Negative Phase: NOTE: FBD3NBIN[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FBD3NBIP[13:0]	I	**FB-DIMM Channel 3 Northbound Input Data Positive Phase: NOTE: FBD3NBIP[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FBD3SBON[9:0]	O	**FB-DIMM Channel 3 Southbound Output Negative Phase:**
FBD3SBOP[9:0]	O	**FB-DIMM Channel 3 Southbound Output Positive Phase:**

3.3 PCI Express* Signal List

3.3.1 PCI Express Common Signals

Signal Name	Type	Description
PECLKN	Analog	**PCI Express Common Clock Negative Phase:**
PECLKP	Analog	**PCI Express Common Clock Positive Phase:**
PEICOMPI	Analog	**PCI Express Impedance Compensation:**
PEICOMPO	Analog	**PCI Express Impedance Compensation:**
PEVCCA	Analog	**PCI Express VCC:** Analog Voltage for the PCI Express PLL:
PEVCCBG	Analog	**PCI Express Band Gap VCC:** Band Gap Voltage
PEVSSA	Analog	**PCI Express VSS:** Analog Voltage for PCI Express PLL:
PEVSSBG	Analog	**PCI Express Band Gap VSS:** Band Gap Voltage

3.3.2 PCI Express Port 0, Enterprise South Bridge Interface (ESI)

PCI Express port 0 is a x4 port dedicated to providing the ESI link between the MCH and the Intel® 631xESB/632xESB I/O Controller Hub.

Signal Name	Type	Description Reference
PE0RN[3:0]	PCIEX I	**PCI Express Port 0 (ESI) Negative Phase Inbound:** (Receive) Signals
PE0RP[3:0]	PCIEX I	**PCI Express Port 0 (ESI) Positive Phase Inbound:** (Receive) Signals
PE0TN[3:0]	PCIEX O	**PCI Express Port 0 (ESI) Negative Phase Outbound:** (Transmit) Signals
PE0TP[3:0]	PCIEX O	**PCI Express Port 0 (ESI) Positive Phase Outbound:** (Transmit) Signals

3.3.3 PCI Express Port 1

PCI Express port 1 is a x4 port.

Signal Name	Type	Description
PE1RN[3:0]	PCIEX I	**PCI Express Port 1 Negative Phase Inbound:** (Receive) Signals
PE1RP[3:0]	PCIEX I	**PCI Express Port 1 Positive Phase Inbound:** (Receive) Signals
PE1TN[3:0]	PCIEX O	**PCI Express Port 1 Negative Phase Outbound:** (Transmit) Signals
PE1TP[3:0]	PCIEX O	**PCI Express Port 1 Positive Phase Outbound:** (Transmit) Signals

3.3.4 PCI Express Port 2

PCI Express port 2 is a x4 port. PCI Express port 2 can be combined with PCI Express port 3 to form a single TOE enabled PCI Express x8 port. Normally port 2 and port 3 are used to increase the bandwidth between the MCH and the Intel® 631xESB/632xESB I/O Controller Hub.

Signal Name	Type	Description
PE2RN[3:0]	PCIEX I	**PCI Express Port 2 Negative Phase Inbound:** (Receive) Signals
PE2RP[3:0]	PCIEX I	**PCI Express Port 2 Positive Phase Inbound:** (Receive) Signals
PE2TN[3:0]	PCIEX O	**PCI Express Port 2 Negative Phase Outbound:** (Transmit) Signals
PE2TP[3:0]	PCIEX O	**PCI Express Port 2 Positive Phase Outbound:** (Transmit) Signals

3.3.5 PCI Express Port 3

PCI Express port 3 can be combined with PCI Express port 2 to form a single TOE enabled PCI Express x8 port. Normally port 2 and port 3 are used to increase the bandwidth between the MCH and the Intel® 631xESB/632xESB I/O Controller Hub.

Signal Name	Type	Description
PE3RN[3:0]	PCIEX I	**PCI Express Port 3 Negative Phase:** Inbound (Receive) Signals
PE3RP[3:0]	PCIEX I	**PCI Express Port 3 Positive Phase:** Inbound (Receive) Signals
PE3TN[3:0]	PCIEX O	**PCI Express Port 3 Negative Phase:** Outbound (Transmit) Signals
PE3TP[3:0]	PCIEX O	**PCI Express Port 3 Positive Phase:** Outbound (Transmit) Signals

3.3.6 PCI Express Port 4

PCI Express port 4 is a x4 port. PCI Express port 4 can be combined with PCI Express port 5 to form a single PCI Express x8 port.

Signal Name	Type	Description
PE4RN[3:0]	PCIEX I	**PCI Express Port 4, Negative Phase Inbound:** (Receive) Signals
PE4RP[3:0]	PCIEX I	**PCI Express Port 4 Positive Phase Inbound:** (Receive) Signals
PE4TN[3:0]	PCIEX O	**PCI Express Port 4, Negative Phase Outbound:** (Transmit) Signal:
PE4TP[3:0]	PCIEX O	**PCI Express Port 4, Positive Phase Outbound:** (Transmit) Signals

3.3.7 PCI Express Port 5

PCI Express port 5 is a x4 port. PCI Express port 5 can be combined with PCI Express port 4 to form a single PCI Express x8 port.

Signal Name	Type	Description
PE5RN[3:0]	PCIEX I	**PCI Express Port 5 Negative Phase Inbound:** (Receive) Signals
PE5RP[3:0]	PCIEX I	**PCI Express Port 5 Positive Phase Inbound:** (Receive) Signals
PE5TN[3:0]	PCIEX O	**PCI Express Port 5 Negative Phase Outbound:** (Transmit) Signals
PE5TP[3:0]	PCIEX O	**PCI Express Port 5 Positive Phase Outbound:** (Transmit) Signals

3.3.8 PCI Express Port 6

PCI Express port 6 is a x4 port. PCI Express port 6 can be combined with PCI Express port 7 to form a single PCI Express x8 port.

Signal Name	Type	Description
PE6RN[3:0]	PCIEX I	**PCI Express Port 6 Negative Phase Inbound:** (Receive) Signals
PE6RP[3:0]	PCIEX I	**PCI Express Port 6 Positive Phase Inbound:** (Receive) Signals
PE6TN[3:0]	PCIEX O	**PCI Express Port 6 Negative Phase Outbound:** (Transmit) Signals
PE6TP[3:0]	PCIEX O	**PCI Express Port 6 Positive Phase Outbound:** (Transmit) Signals

3.3.9 PCI Express Port 7

PCI Express port 7 is a x4 port. PCI Express port 7 can be combined with PCI Express port 6 to form a single PCI Express x8 port.

Signal Name	Type	Description
PE7RN[3:0]	PCIEX I	**PCI Express Port 7 Negative Phase Inbound:** (Receive) Signals
PE7RP[3:0]	PCIEX I	**PCI Express Port 7 Positive Phase Inbound:** (Receive) Signals
PE7TN[3:0]	PCIEX O	**PCI Express Port 7 Negative Phase Outbound:** (Transmit) Signals
PE7TP[3:0]	PCIEX O	**PCI Express Port 7 Positive Phase Outbound:** (Transmit) Signals

3.4 System Management Bus Interfaces

There are seven SM Bus interfaces dedicated to specific functions. These functions are:

- System Management
- Four buses dedicated to FB-DIMM serial presents detect, one for each channel
- PCI Hot-Plug

Signal Name	Type	Description
CFGSMBCLK	SMBus I/O	**Slave SMB Clock:** System Management Bus Clock
CFGSMBDATA	SMBus I/O	**Slave SMB Data:** SMB Address/Data
SPD0SMBCLK	SMBus I/O	**FB-DIMM Channel 0 SMB Clock:** FB-DIMM Memory Serial Presents Detect 0, System Management Bus Clock
SPD0SMBDATA	SMBus I/O	**FB-DIMM Channel 0 SMB Data:** FB-DIMM Memory Serial Presents Detect 0, SMB Address/Data
SPD1SMBCLK	SMBus I/O	**FB-DIMM Channel 1 SMB Clock:** FB-DIMM Memory Serial Presents Detect 1, System Management Bus Clock
SPD1SMBDATA	SMBus I/O	**FB-DIMM Channel 1 SMB Data:** FB-DIMM Memory Serial Presents Detect 1, SMB Address/Data
SPD2SMBCLK	SMBus I/O	**FB-DIMM Channel 2 SMB Clock:** FB-DIMM Memory Serial Presents Detect 2, System Management Bus Clock
SPD2SMBDATA	SMBus I/O	**FB-DIMM Channel 2 SMB Data:** FB-DIMM Memory Serial Presents Detect 2, SMB Address/Data
SPD3SMBCLK	SMBus I/O	**FB-DIMM Channel 3 SMB Clock:** FB-DIMM Memory Serial Presents Detect 3, System Management Bus Clock
SPD3SMBDATA	SMBus I/O	**FB-DIMM Channel 3 SMB Data:** FB-DIMM Memory Serial Presents Detect 3, SMB Address/Data
VPPSMBCLK	SMBus I/O	**PCI SMB Clock:** PCI Hot-Plug Master VPI, System Management Bus Clock
VPPSMBDATA	SMBus I/O	**PCI SMB Data:** PCI Hot-Plug Master VPI, SMB Address/Data

3.5 XDP Port Signal List

Signal Name	Type	Description
XDPCOMCRES	Analog	**XDP Bus Compensation:**
XDPD_N[15:0]	GTL+ I/O	**Data Bus:**.
XDPSTBN_N XDPSTBP_N	GTL+ I/O	**Data Bus Strobe Negative and Positive Phases:**
XDPODTCRES	Analog	**XDP Bus Compensation:**
XDPRDY_N	GTL+ I/O	**Data Bus Ready:**
XDPSLWCRES	Analog	**XDP Bus Slew Rate Compensation:**.

3.6 JTAG Bus Signal List

Signal Name	Type	Description
TCK	JTAG I	**Clock:** Clock pin of the JTAG.
TDI	JTAG I	**Data Input:** Serial chain input of the JTAG.
TDO	JTAG O	**Data Output:** Serial chain output of the JTAG.
TMS	JTAG I	**State Machine:** JTAG State machine control
TRST_N	JTAG I	**Reset:** Asynchronous reset of the JTAG.

3.7 Clocks, Reset and Miscellaneous

Signal Name	Type	Description
CORECLKN	Analog	**Differential Processor Core Clock Negative Phase:** These pins receive a low-voltage differential host clock from the external clock synthesizer. This clock is used by all of the MCH logic that is in the Host clock domain.
CORECLKP	Analog	**Differential Processor Core Clock Positive Phase:** These pins receive a low-voltage differential host clock from the external clock synthesizer. This clock is used by all of the MCH logic that is in the Host clock domain.
COREVCCA	Analog	**Core VCC:** Analog Voltage for the PLL
COREVSSA	Analog	**Core VSS:** Analog Voltage for PLL
ERR_N[2:0]	SMBus O	**Error Output:** Error output signal: ERR[0] = Correctable and recoverable error from the memory subsystem ERR[1] = Uncorrectable error from the MCH ERR[2] = Fatal error from the MCH
FSB1DRVCRES	Analog	**Processor Bus Compensation**: Busses 0 and 1
FSB1ODTCRES	Analog	**Processor Bus Compensation**: Busses 0 and 1
FSB1SLWCRES	Analog	**Processor Bus Slew Rate Compensation: Busses 0 and 1**
FSB3DRVCRES	Analog	**Processor Bus Compensation**: Busses 2 and 3
FSB3ODTCRES	Analog	**Processor Bus Compensation**: Busses 2 and 3
FSB3SLWCRES	Analog	**Processor Bus Slew Rate Compensation: Busses 2 and 3**
FSBSLWCTRL	I	FSB Slew Rate Control.
FSBVCCA	Analog	**FSB VCC:** Analog Voltage for the FSBPLL
INT_N[6:0]	CMOS O	**Interrupt signals. Each bit is a separate interrupt.**
PWRGOOD	SMBus I	**Power OK:** When asserted, this signal indicates that all power supplies are in specification.

Signal Name	Type	Description
PSEL[2:0]	I	FSB Speed select. Only “000” configuration is supported. These 3 pins should be pulled-down to ground.
RESET_N	SMBus I	**MCH Reset:** This is the hard reset
RSVD	No Connect	**Reserved Pins:** Leave these pins unconnected.
TESTHI_V3REF	Strap	**Pull-up to V3REF.**

3.8 Power and Ground Signals

Signal Name	Description
V3REF	**3.3V VCC:** Common 3.3 for SMB buses and 3.3V Pins
VCC	**VCC 1.5V** core voltage
VCCFBD	**VCC1.5V** for powering FBD channels
VCCSF	**VCC 1.5V** for Snoop Filter
VCCSPL	Special VCC pin. Power with a 1.5V VCC supply.
VCCPE	**VCC 1.5V** for PCI Express ports
VSS	**Ground Return:** Common return for power supplies
VTT	**VTT Supply:** Termination voltage for FSB. 1.2V or 1.1V termination.

§

4 Register Description

The Intel® 7300 Chipset MCH contains three sets of software accessible registers, accessed via the host processor I/O address space:

- Control registers I/O mapped into the processor I/O space that controls access to PCI configuration spaces.
- Internal configuration registers residing within the MCH are partitioned into logical device register sets ("logical" since they reside within a single physical device). The first register set is dedicated to MCH functionality (controls PCI bus 0, i.e., DRAM configuration, other chipset operating parameters, and optional features). The second register set is dedicated to ESI control. The third register set is dedicated to ESI control.

The MCH supports PCI configuration space accesses using the mechanism denoted as Configuration Mechanism 1 in the PCI specification as defined in the *PCI Local Bus Specification*, Revision 2.3. All the registers are organized by bus, device, function, etc. as defined in the *PCI Express Base Specification*, Revision 1.0a. The MCH supports registers in PCI Express extended space. All MCH registers in Intel® 7300 Chipset appear on PCI Bus #0.

In addition, the MCH registers can be accessed by a memory mapped register access mechanism (as MMIO), a PCI configuration access mechanism (only PCI space registers), and register access mechanisms through JTAG and SMBus. The memory mapped access mechanism is further broken down into different ranges. The internal registers of this chip set can be accessed in 8-bit, 16-bit, or 32-bit quantities, with the exception of CFGADR which can only be accessed as a 32-bit. All multi-byte numeric fields use "little-endian" ordering (i.e., lower addresses contain the least significant parts of the field).

In addition, the MCH can forward accesses to all PCI/PCI Express configuration registers south of the MCH through the same mechanisms.

4.1 Register Terminology

Registers and register bits are assigned one or more of the following attributes. These attributes define the behavior of register and the bit(s) that are contained with in. All bits are set to default values by hard reset. Sticky bits retain their states between hard resets.

Term	Description
RO	**Read Only**. If a register bit is read only, the hardware sets its state. The bit may be read by software. Writes to this bit have no effect.
WO	**Write Only.** The register bit is not implemented as a bit. The write causes some hardware event to take place.
RW	**Read/Write**. A register bit with this attribute can be read and written by software.
RC	**Read Clear:** The bit or bits can be read by software, but the act of reading causes the value to be cleared.
RCW	**Read Clear/Write:** A register bit with this attribute, will get cleared after the read. The register bit can be written.
RWC	**Read/Write Clear**. A register bit with this attribute, can be read or cleared by software. In order to clear this bit, a one must be written to it. Writing a zero will have no effect.

Term	Description
RWS	**Read/Write/Set:** A register bit can be either read or set by software. In order to set this bit, a one must be written to it. Writing a zero to this bit has no effect. Hardware will clear this bit.
RWL	**Read/Write/Lock**. A register bit with this attribute can be read or written by software. Hardware or a configuration bit can lock bit and prevent it from updated.
RWO	**Read/Write Once.** A register bit with this attribute can be written to only once after power up. After the first write, the bit becomes read only. This attribute is applied on a bit by bit basis. For example, if the RWO attribute is applied to a 2 bit field, and only one bit is written, then the written bit cannot be rewritten (unless reset). The unwritten bit, of the field, may still be written once. This is special case of RWL.
RRW	**Read/Restricted Write.** This bit can be read and written by software. However, only supported values will be written. Writes of non supported values will have no effect.
L	**Lock.** A register bit with this attribute becomes Read Only after a lock bit is set.
RV	**Reserved Bit.** This bit is reserved for future expansion and must not be written. The *PCI Local Bus Specification,* Revision 2.2 requires that reserved bits must be preserved. Any software that modifies a register that contains a reserved bit is responsible for reading the register, modifying the desired bits, and writing back the result.
Reserved Bits	Some of the MCH registers described in this section contain reserved bits. These bits are labeled "Reserved". Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back. Note that software does not need to perform a read-merge-write operation for the Configuration Address (CONFIG_ADDRESS) register.
Reserved Registers	In addition to reserved bits within a register, the MCH contains address locations in the configuration space of the Host-ESI bridge entity that are marked either "Reserved" or "Intel Reserved". The MCH responds to accesses to "Reserved" address locations by completing the host cycle. When a "Reserved" register location is read, a zero value is returned. ("Reserved" registers can be 8, 16, or 32 bits in size). Writes to "Reserved" registers have no effect on the MCH. Registers that are marked as "Intel Reserved" must not be modified by system software. Writes to "Intel Reserved" registers may cause system failure. Reads to "Intel Reserved" registers may return a non-zero value.
Default Value upon a Reset	Upon a reset, the MCH sets all of its internal configuration registers to predetermined default states. Some register values at reset are determined by external strapping options. The default state represents the minimum functionality feature set required to successfully bring up the system. Hence, it does not represent the optimal system configuration. It is the responsibility of the system initialization software (usually BIOS) to properly determine the DRAM configurations, operating parameters and optional system features that are applicable, and to program the MCH registers accordingly.
"ST" appended to the end of a bit name	The bit is "sticky" or unchanged by a hard reset. These bits can only be cleared by a PWRGOOD reset.

4.2 Platform Configuration Structure

In some previous chipsets, the MCH and the South Bridge were physically connected by PCI bus 0. From a configuration standpoint, both components appeared to be on PCI bus 0 which was also the system's primary PCI expansion bus. The MCH contained two PCI devices while the south bridge was considered one PCI device with multiple functions.

In the Intel® 7300 Chipset platform the configuration structure is significantly different. The MCH and the Intel® 631xESB/632xESB I/O Controller Hub are physically connected by the ESI interface; thus, from a configuration standpoint, the ESI interface is logically PCI bus 0 (Note that the Intel® 631xESB/632xESB I/O Controller Hub must also be connected via at least a x4 PCI Express link as well). As a result, all devices internal to the MCH and Intel® 631xESB/632xESB I/O Controller Hub appear to be on PCI bus 0. The system's primary PCI expansion bus is physically attached to the Intel® 631xESB/632xESB I/O Controller Hub and, from a configuration perspective, appears to be a hierarchical PCI bus behind a PCI-to-PCI bridge; therefore, it has a programmable PCI Bus number.

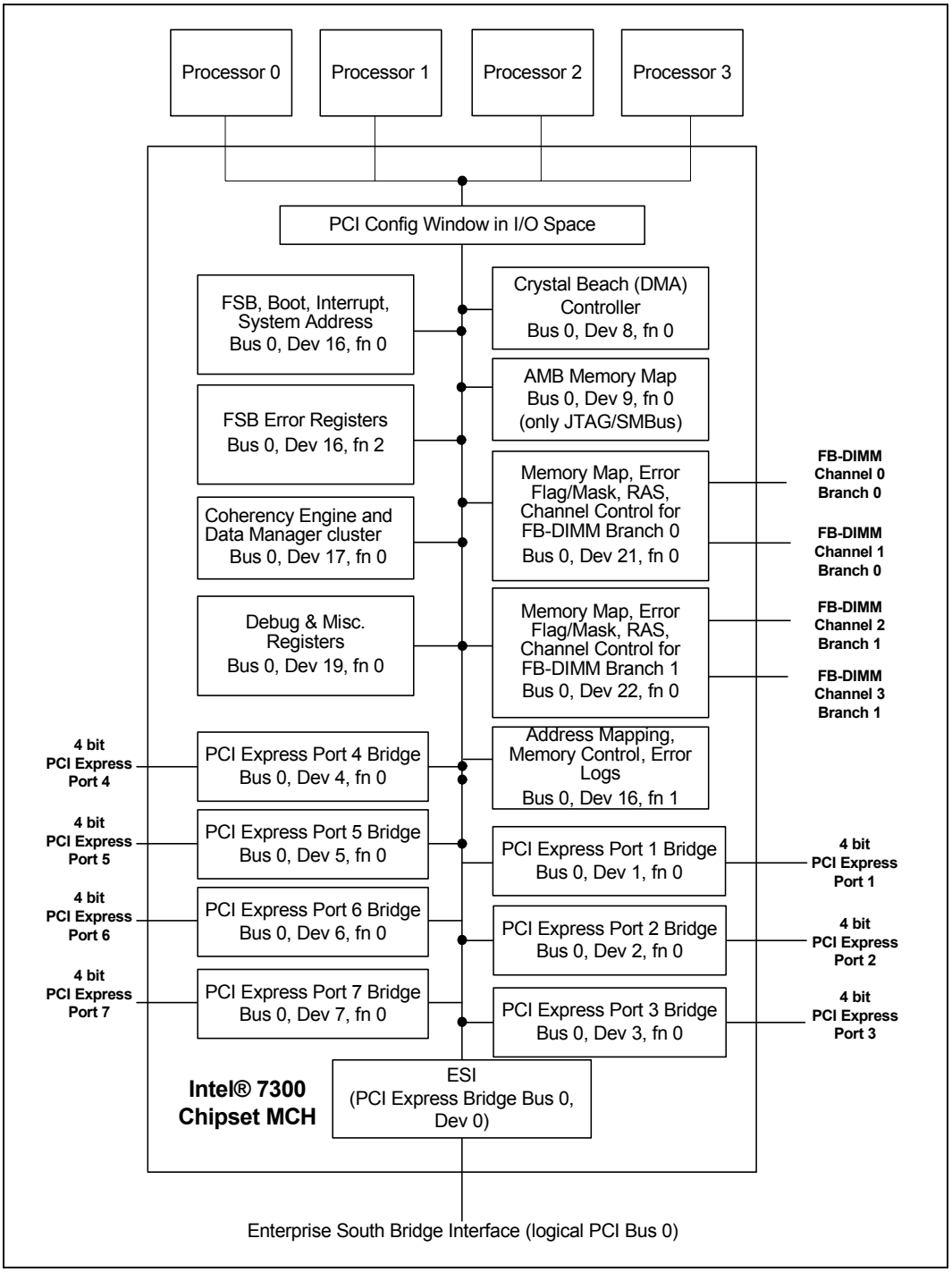
The MCH contains 14 PCI devices within a single physical component. The configuration registers for these devices are mapped as devices residing on PCI bus 0.

- **Device 0:** ESI bridge/PCI Express Port 0. Logically, this appears as a PCI device that resides on PCI bus 0. Physically Device 0, Function 0 contains the PCI Express configuration registers for the ESI port, and other MCH specific registers.
- **Device 1:** PCI Express 1. Logically this appears as a PCI device residing on bus 0. Device 1, Function 0 is routed to the PCI Express configuration registers for PCI Express port 1. PCI Express port 1 resides at DID of 3604h.
- **Device 2:** PCI Express 2. Logically this appears as a PCI device residing on bus 0. Device 2, Function 0 is routed to the PCI Express configuration registers for PCI Express port 2. When PCI Express ports 2 and 3 are combined into a single x8 port, controlled by port 2 registers, Device 3, Function 0 (port 3) configuration registers are inactive. PCI Express port 2 resides at DID of 3605h.
- **Device 3:** PCI Express 3. Logically this appears as a PCI device that resides on bus 0. Device 3, Function 0 contains the PCI Express configuration registers for PCI Express port 3. When PCI Express ports 2 and 3 are combined into a single x8 port, controlled by port 2 registers, these configuration registers are inactive. PCI Express port 3 resides at DID of 3606h.
- **Device 4:** PCI Express 4. Logically this appears as a PCI device that resides on bus 0. Device 4, Function 0 contains the PCI Express configuration registers for PCI Express port 4. When PCI Express ports 4 and 5 are combined into a single x8 port, Device 4, Function 0 contains the configuration registers and Device 5, Function 0 (port 5) configuration registers are inactive. PCI Express port 4 resides at DID of 3607h.
- **Device 5:** PCI Express 5. Logically this appears as a PCI device that resides on bus 0. Device 5, Function 0 contains the PCI Express configuration registers for PCI Express port 5.When PCI Express ports 4 and 5 are combined into a single x8 port Device 4, Function 0 contains the configuration registers, and these configuration registers are inactive. PCI Express port 5 resides at DID of 3608h.
- **Device 6:** PCI Express 6. Logically this appears as a PCI device residing on bus 0. Device 6, Function 0 contains the PCI Express configuration registers for PCI Express port 6.When PCI Express ports 6 and 7 are combined into a single x8 port Device 6, Function 0 contains the configuration registers, and Device 7, Function 0

(port 7) configuration registers are inactive. PCI Express port 6 resides at DID of 3609h.

- **Device 7:** PCI Express 7. Logically this appears as a PCI device residing on bus 0. Device 7, Function 0 contains the PCI Express configuration registers for PCI Express port 7. When PCI Express ports 6 and 7 are combined into a single x8 port Device 6, Function 0 contains the configuration registers, and these configuration registers are inactive. PCI Express port 2 resides at DID of 360Ah.
- **Device 8:** Intel® QuickData Technology DMA Controller. Logically this appears as DMA device residing on bus 0. Device 8, Function 0 contains the DMA controller configuration registers for the DMA channels. Device 8, Function 1 contains the MMIO space configuration registers. The Intel® QuickData Technology DMA controller resides at DID 360Bh.
- **Device 9:** Device 9, Function 0 is routed to the Advanced Memory Buffer memory map. This interface is supported through the JTAG and SMBus interfaces and AMBSELECT register only.
- **Device 16**: Device 16, Function 0 is routed to the Frontside Bus (FSB) Controller, Interrupt and System Address registers. Function 1 is routed to the Frontside Bus Address Mapping, Memory Control, and Error registers. Function 2 is routed to FSB Error Registers. These devices reside at DID 360Ch.
- **Device 17**: Device 17, Function 0 is routed to the Snoop Filter 0 registers. Device 17, Function 3 is routed to the Snoop Filter 1 registers. These devices reside at DID 360Dh.
- **Device 19:** Device 19, Function 0 is routed to the Debug and Miscellaneous registers. These devices reside at DID 360Eh.
- **Device 21:** Device 21, Function 0, FBD Branch 0 Memory Map, Error Flag/Mask, and Channel Control registers. These devices reside at DID 360Fh.
- **Device 22:** Device 22, Function 0, FBD Branch 1 Memory Map, Error Flag/Mask, and Channel Control registers. These devices reside at DID 3610h.

Figure 4-1. Conceptual Intel® 7300 Chipset MCH PCI Configuration Diagram



4.3 Routing Configuration Accesses

Intel® 7300 Chipset MCH supports both PCI Type 0 and Type 1 configuration access mechanisms as defined in the PCI Local Bus Specification, Revision 2.3. PCI Revision 2.3 defines hierarchical PCI busses. Type 0 configuration access are used for registers located within a PCI device that resides on the local PCI bus. i.e. The PCI bus the transaction is initiated on. Type 0 configuration transactions are not propagated beyond the local PCI bus. Type 0 configuration transactions must be claimed by a local device or master aborted.

Type 1 configuration accesses are used for devices residing on subordinate PCI buses. i.e Devices that are connected via PCI-to-PCI bridges. All targets except PCI-to-PCI bridges ignore Type 1 configuration transactions. PCI-to-PCI bridges decode the bus number information in Type 1 transactions. It the transaction is targeted to a device local to the PCI-to-PCI bridge it is translated into a Type 0 transaction and issued to the device. If the transaction is targeted to a bus subordinate (behind) to PCI-to-PCI bridge, it passed through unchanged. Otherwise the Type 1 transaction is dropped.

Accesses to non operational or non existent devices are master aborted. This means that writes are dropped and reads return all 1’s.

4.3.1 Standard PCI Bus Configuration Mechanism

The PCI Bus defines a slot based “configuration space” that supports up to 32 devices. Each device is allowed to contain up to eight functions with each function containing up to 256, 8-bit configuration registers. The PCI specification defines two bus cycles to access the PCI configuration space: Configuration Read and Configuration Write. Memory and I/O spaces are supported directly by the processor. Configuration space is supported by a mapping mechanism implemented within the MCH. The PCI 2.3 specification defines the configuration mechanism to access configuration space. The configuration access mechanism makes use of the CONFIG_ADDRESS Register (at I/O address 0CF8h through 0CFBh) and CONFIG_DATA Register (at I/O address 0CFCh through 0CFFh). To reference a configuration register a DWord I/O write cycle is used to place a value into CONFIG_ADDRESS that specifies the PCI bus, the device on that bus, the function within the device, and a specific configuration register of the device function being accessed. CONFIG_ADDRESS[31] must be set to 1b, to enable a configuration cycle. CONFIG_DATA then becomes a window into the four bytes of configuration space specified by the contents of CONFIG_ADDRESS. Any read or write to CONFIG_DATA will result in the MCH translating the CONFIG_ADDRESS into the appropriate configuration cycle.

The MCH is responsible for translating and routing the processor’s I/O accesses to the CONFIG_ADDRESS and CONFIG_DATA registers to internal MCH configuration registers.

4.3.2 PCI Bus 0 Configuration Mechanism

The MCH decodes the Bus Number (bits 23:16) and the Device Number fields of the CONFIG_ADDRESS register. If the Bus Number field of CONFIG_ADDRESS is 0, the configuration cycle is targeting a device on PCI Bus 0.

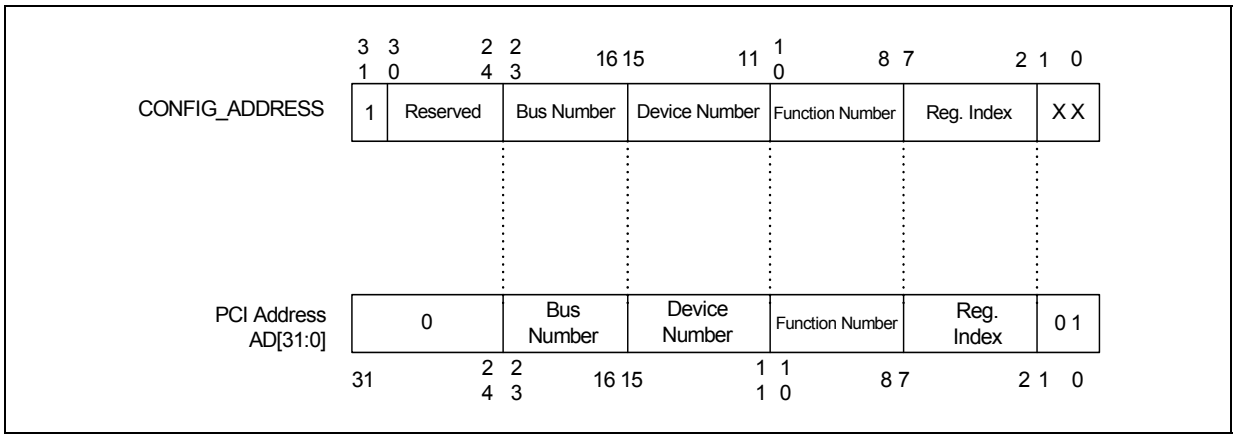
The ESI bridge entity within the MCH is hardwired as Device 0 on PCI Bus 0. The ESI bridge passes PCI south bridge configuration requests to the south bridge.

4.3.3 Primary PCI and Downstream Configuration Mechanism

If the Bus Number in the CONFIG_ADDRESS is non-zero, the MCH will generate a Type 1 PCI configuration cycle. A[1:0] of the ESI request packet for the Type 1 configuration cycle will be 01. Bits 31:2 of the CONFIG_ADDRESS register will be translated to the A[31:2] field of the ESI request packet of the configuration cycle as shown in Figure 4-2. This configuration cycle will be sent over the ESI to Intel® 631xESB/632xESB I/O Controller Hub.

If the cycle is forwarded to the Intel® 631xESB/632xESB I/O Controller Hub via ESI, the Intel® 631xESB/632xESB I/O Controller Hub compares the non-zero Bus Number with the Secondary Bus Number and Subordinate Bus Number Registers of its PCI-to-PCI bridges to determine if the configuration cycle is meant for primary PCI bus, one of the Intel® 631xESB/632xESB I/O Controller Hub's PCI Express ports, or a downstream PCI bus.

Figure 4-2. Type 1 Configuration Address to PCI Address Mapping



4.4 Device Mapping

Each component in a Intel® 7300 Chipset system is uniquely identified by a PCI bus address consisting of; Bus Number, Device Number and Function Number. Device configuration is based on the PCI Type 0 configuration conventions. All PCI devices with in a Intel® 7300 Chipset platform must support Type 0 configuration accesses. All MCH registers in the MCH appear on Bus #0.

All MCH configuration registers reside in the configuration space defined by Bus, Device, Function, Register address. Some registers do not appear in all portions of this space and some mechanisms do not access all portions of this space. In general the configuration space is sparsely populated. The following table defines where the various fields of configuration register addresses appear. Each row defines a different access mechanism, register, interface, or decoder. Each column defines a different field of the configuration address.

Table 4-1. Configuration Address Bit Mapping

	Source/ Destination	Bus	Device	Function	Dword Offset		Byte in Dword	Type
					[11:8]	[5:0]		
PCI Express Config Txns (including ESI)	Both	Bus[7:0]	Device[4:0]	Function[2:0]	Extended Register Addr[3:0]	Register [5:0]	1st DW BE[3:0]	Fmt, Type
PCI Express MMCFG on FSB	Source	A[27:20]	A[19:15]	A[14:12]	A[11:8]	A[7:3] BE[7:4]	BE[7:0]	n/a
PCI Express MMCFG from ESI or PCI Express	Not permitted to access MCH or FB-DIMM regs and will be master aborted. Peer to Peer accesses targeting valid MMIO space will be forwarded through appropriate decoding.							
CPU/Inbound IQD_BAR MMIO Access	Source	0	8	1	A[11:8]	A[7:3] BE[7:4]	BE[7:0]	n/a
CFGADR Register	Source	Bus Number [7:0]	DeviceID [4:0]	Function Number[2:0]	not present	Register Address [5:0]	Not present	n/a
CFC on FSB	Source	CFGADR Register, see row above					BE[7:4]	n/a
JTAG Config Access	Source	Bus Number [7:0]	DeviceID [4:0]	Function Number[2:0]	Extended Register Addr[3:0]	Register Address[7:2]	Register Address [1:0]	n/a
SMBus Config Access	Source	Bus Number [7:0]	Dev[4:0]	Func[2:0]	Reg Number [11:8]	Reg[7:2]	command, Register Number	n/a
Fixed MCH Memory Mapped on FSB	Source	0	16	0	cannot access	A[15:10]	All accesses are 4 byte	n/a
MCH Register Decoding	Destination	00000000	See Table 14-4	Function[2:0]	Dword Offset[9:6]	Dword Offset[5:0]	Byte[3:0]	n/a
FB-DIMM Config Cmds	Destination	A[23:15] always 0	See Note[a]		Cannot access	A[7:3] BE[7:4]	BE[7:0]	n/a

Notes:

a. These accesses are used to select channel/DIMM based on the AMBASE register.

4.4.1 Special Device and Function Routing

All devices in the MCH reside on Bus 0. The following table describes the devices and functions that the MCH implements or routes specially. The DIMM component designator consists of a three-digit code: the first digit is the branch, the second digit is the channel on the branch, and the third digit is the DIMM (FB-DIMM command "DS" field) on the channel.

Table 4-2. Functions Specially Handled by the MCH (Sheet 1 of 2)

Component	Register Group	DID	Device	Function	Comment
MCH	ESI Port 0	3600h	0	0	Depending on what is connected to these ports, some may not be accessible.
MCH	PCI Express Port 1	3604h	1	0	
MCH	PCI Express Port 2	3605h	2	0	
MCH	PCI Express Port 3	3606h	3	0	
MCH	PCI Express Port 4	3607h	4	0	
MCH	PCI Express Port 5	3608h	5	0	
MCH	PCI Express Port 6	3609h	6	0	
MCH	PCI Express Port 7	360Ah	7	0	

Table 4-2. Functions Specially Handled by the MCH (Sheet 2 of 2)

Component	Register Group	DID	Device	Function	Comment
MCH	Intel® QuickData Technology (DMA)	360Bh	8	0	
MCH	Intel® QuickData Technology MMIO Space	N/A	8	1	
DIMM	AMB Memory Mapped registers	N/A	9	0	Route out to AMB per AMBSELECT register only for JTAG/SMBus.
MCH	Processor Bus, Boot, Interrupt, System Address	360Ch	16	0	
MCH	Address Mapping, Memory Control, Error Logs	360Ch	16	1	
MCH	FSB Error Registers	360Ch	16	2	
MCH	FSB Error Registers	360Ch	16	3	
MCH		360D	17	0	
MCH		360D	17	3	
MCH		360E	19	0	
MCH	Memory Map, Error Flag/Mask, RAS, Channel Control for FB-DIMM Branch 0	360Fh	21	0	
MCH	Memory Map, Error Flag/Mask, RAS, Channel Control for FB-DIMM Branch 1	3610h	22	0	
MCH	PCI Express Port 2-3	3605h	2	0	x8 mode. Only port 2 is active
MCH	PCI Express Port 4-5	3607h	4	0	x8 mode. Only port 4 is active
MCH	PCI Express Port 6-7	3609h	6	0	x8 mode. Only port 6 is active

To comply with the PCI specification, accesses to non-existent functions, registers, and bits will be master aborted. This behavior is defined in the following table:

Table 4-3. Access to "Non-Existent" Register Bits

Access to	Writes	Reads
Devices listed in Table 3-2, "Functions Specially Handled by the MCH" on page 53, but to functions not listed	Have no effect	MCH returns all ones
Devices listed in Table 3-2, "Functions Specially Handled by the MCH" on page 53, but to registers not listed in Section 3.8, "Register Definitions"	Have no effect	MCH returns all zeroes
Reserved bits in registers	Software must read-modify-write to preserve the value	MCH returns all zeroes

4.5 I/O Mapped Registers

There are only two I/O addresses that affect MCH state. The first address is the DWORD location (CF8h) references a read/write register that is named CONFIG_ADDRESS. The second DWORD address (CFCh) references a read/write register named CONFIG_DATA. These two addresses are used for the PCI CFCh / CF8h configuration access mechanism.

4.5.1 CFGADR: Configuration Address Register

CFGADR is written only when a processor I/O transaction to I/O location CF8h is referenced as a DWord; a Byte or Word reference will not access this register, but will generate an I/O space access. Therefor the only I/O space taken up by this register is the DWORD at location CF8h. I/O devices that share the same address but use BYTE or WORD registers are not affected because their transactions will pass through the host bridge unchanged.

The CFGADR register contains the Bus Number, Device Number, Function Number, and Register Offset for which a subsequent CFGDAT access is intended. The mapping between fields in this register and PCI Express configuration transactions is defined by Table 4-1.

I/O Address: CF8h			
Bit	**Attr**	**Default**	**Description**
31	RW	0h	**CFGE: Configuration Enable** Unless this bit is set, accesses to the **CFGDAT** register will not produce a configuration access, but will be treated as other I/O accesses. This bit is strictly an enable for the CFC/CF8 access mechanism and is not forwarded to ESI or PCI Express.
30:24	RV	00h	Reserved.
23:16	RW	00h	**Bus Number** If 0, the MCH examines device to determine where to route. If non-zero, route as per PBUSN and SBUSN registers.
15:11	RW	0h	**Device Number** This field is used to select one of the 32 possible devices per bus.
10:8	RW	0h	**Function Number** This field is used to select the function of a locally addressed register.
7:2	RW	00h	**Register Offset** If this register specifies an access to MCH registers, this field specifies a group of four bytes to be addressed. The bytes accessed are defined by the Byte enables of the CFGDAT register access
1:0	RW	0h	Writes to these bits have no effect, reads return 0

4.5.2 CFGDAT: Configuration Data Register

CFGDAT provides data for the 4 bytes of configuration space defined by CFGADR. This register is only accessed if there is an access to I/O address, CFCh on the processor bus and CFGADR.CFGE (configuration enable) bit is set. The byte enables with the I/O access define how many configuration bytes are accessed.

I/O Address: CFCh			
Bit	**Attr**	**Default**	**Description**
31:0	RW	0	**Configuration Data Window** The data written or read to the configuration register (if any) specified by **CFGADR**

4.6 MCH Fixed Memory Mapped Registers

These registers are mapped into the fixed chipset specific range located from FE60 0000h - FE6F FFFFh.These appear at fixed addresses to support the boot process. These registers also appear in the regular PCI Express configuration space.

The following table defines the memory address of the registers in this region.

Table 4-4. Mapping for Fixed Memory Mapped Registers

Register	Memory Address
BOFL0	FE60_C000
BOFL1	FE60_C400
BOFL2	FE60_C800
BOFL3	FE60_CC00
SPAD0	FE60_D000
SPAD1	FE60_D400
SPAD2	FE60_D800
SPAD3	FE60_DC00
SPADS0	FE60_E000
SPADS1	FE60_E400
SPADS2	FE60_E800
SPADS3	FE60_EC00
AMBASE[31:0]	FE61_4800
AMBASE[63:32]	FE61_4C00
HECBASE	FE61_6400

4.7 Detailed Configuration Space Maps

Table 4-5. Device 0, Function 0: PCI Express PCI Space

DID			VID	00h	PEXSLOTCAP		80h
PEXSTS			PEXCMD	04h	PEXSLOTSTS	PEXSLOTCTRL	84h
CCR			RID	08h		PEXRTCTRL	88h
BIST	HDR	PRI_LT	CLS	0Ch	PEXRTSTS		8Ch
				10h			90h
				14h			94h
				18h			98h
				1Ch			9Ch
				20h			A0h
				24h			A4h
				28h			A8h
SID			SVID	2Ch			ACh
				30h			B0h
			CAPPTR	34h			B4h
				38h			B8h
		INTP	INTL	3Ch			BCh
				40h			C0h
		IQDPRES		44h			C4h
PEXCTRL				48h			C8h
		PEXCTRL3	PEXCTRL2	4Ch			CCh
PMCAP				50h			D0h
PMCSR				54h	ESICTRL		D4h
MSICTRL		MSINXPTR	MSICAPID	58h			D8h
MSIAR				5Ch			DCh
MSIDR				60h	PXPLWTCTRL		E0h
PEXCTRL4				64h	DEVPRES		E4h
PEXCTRL5				68h			E8h
PEXCAP		PEXCAPL		6Ch			ECh
PEXDEVCAP				70h			F0h
PEXDEVSTS		PEXDEVCTRL		74h			F4h
PEXLNKCAP				78h		ERRINJCON	F8h
PEXLNKSTS		PEXLNKCTRL		7Ch			FCh

Table 4-6. Device 0, Function 0: PCI Express Extended Registers

PEXENHCAP	100h		180h
UNCERRSTS	104h		184h
UNCERRMSK	108h		188h
UNCERRSEV	10Ch		18Ch
CORERRSTS	110h		190h
CORERRMSK	114h		194h
AERRCAPCTRL	118h		198h
HDRLOG0	11Ch		19Ch
HDRLOG1	120h		1A0h
HDRLOG2	124h		1A4h
HDRLOG3	128h		1A8h
RPERRCMD	12Ch		1ACh
RPERRSTS	130h		1B0h
RPERRSID	134h		1B4h
	138h		1B8h
	13Ch		1BCh
SPCAPID	140h		1C0h
PEX_ERR_DOCMD	144h		1C4h
EMASK_UNCOR_PEX	148h		1C8h
EMASK_COR_PEX	14Ch		1CCh
EMASK_RP_PEX	150h		1D0h
PEX_FAT_FERR	154h		1D4h
PEX_NF_COR_FERR	158h		1D8h
PEX_FAT_NERR	15Ch		1DCh
PEX_NF_COR_NERR	160h		1E0h
	164h		1E4h
	168h		1E8h
	16Ch		1ECh
	170h		1F0h
	174h		1F4h
	178h		1F8h
	17Ch		1FCh

Table 4-7. Device 0, Function 0: PCI Express Error Injection and Mask Registers

	200h		280h
	204h		284h
	208h		288h
	20Ch		28Ch
	210h		290h
	214h		294h
	218h		298h
	21Ch		29Ch
	220h		2A0h
	224h		2A4h
	228h		2A8h
	22Ch		2ACh
	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
PEXEINJCTL	278h		2F8h
PEXEINJMSK	27Ch		2FCh

Table 4-8. Device 0, Function 0: PCI Express IBIST Registers

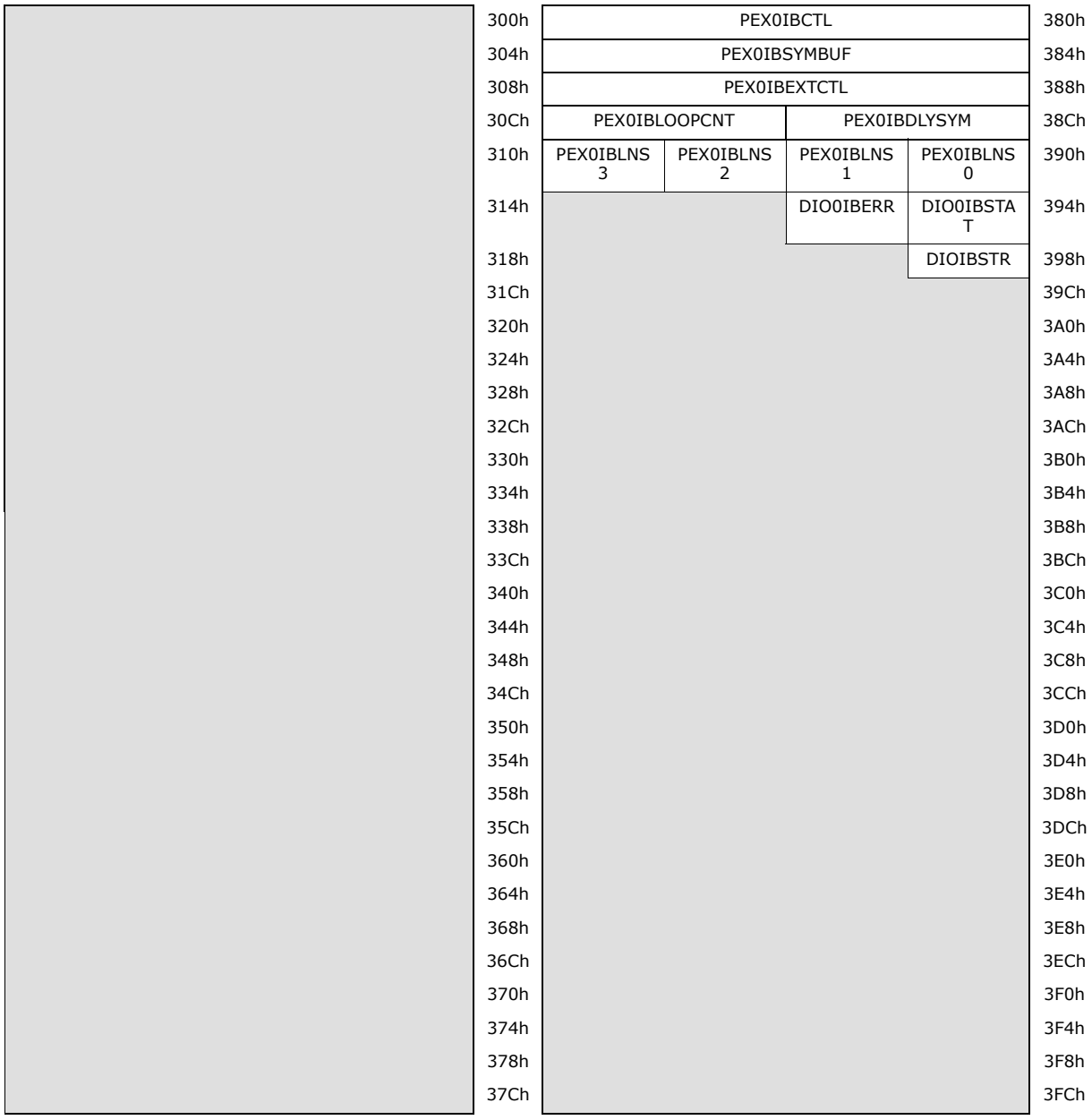


Table 4-9. Device 1-7, Function 0: PCI Express PCI Space

DID			VID	00h	PEXSLOTCAP			80h
PEXSTS			PEXCMD	04h	PEXSLOTSTS		PEXSLOTCTRL	84h
CCR			RID	08h			PEXRTCTRL	88h
BIST	HDR	PRI_LT	CLS	0Ch	PEXRTSTS			8Ch
				10h				90h
				14h				94h
SEC_LT	SUBUSN	SBUSN	PBUSN	18h				98h
SECSTS		IOLIM	IOBASE	1Ch				9Ch
MLIM		MBASE		20h				A0h
PMLIM		PMBASE		24h				A4h
PMBU				28h				A8h
PMLU				2Ch				ACh
				30h				B0h
			CAPPTR	34h				B4h
				38h				B8h
BCTRL		INTP	INTL	3Ch				BCh
				40h				C0h
				44h				C4h
PEXCTRL				48h				C8h
		PEXCTRL3	PEXCTRL2	4Ch				CCh
PMCAP				50h				D0h
PMCSR				54h				D4h
MSICTRL		MSINXPTR	MSICAPID	58h				D8h
MSIAR				5Ch				DCh
MSIDR				60h				E0h
PEXCTRL4				64h				E4h
PEXCTRL5				68h				E8h
PEXCAP		PEXCAPL		6Ch				ECh
PEXDEVCAP				70h		SVID_NXTP	SVID_CAPID	F0h
PEXDEVSTS		PEXDEVCTRL		74h	SID	SVID		F4h
PEXLNKCAP				78h		ERRINJCON		F8h
PEXLNKSTS		PEXLNKCTRL		7Ch				FCh

Table 4-10. Device 1-7, Function 0: PCI Express Extended Registers

PEXENHCAP	100h		180h
UNCERRSTS	104h		184h
UNCERRMSK	108h		188h
UNCERRSEV	10Ch		18Ch
CORERRSTS	110h		190h
CORERRMSK	114h		194h
AERRCAPCTRL	118h		198h
HDRLOG0	11Ch		19Ch
HDRLOG1	120h		1A0h
HDRLOG2	124h		1A4h
HDRLOG3	128h		1A8h
RPERRCMD	12Ch		1ACh
RPERRSTS	130h		1B0h
RPERRSID	134h		1B4h
	138h		1B8h
	13Ch		1BCh
SPCAPID	140h		1C0h
PEX_ERR_DOCMD	144h		1C4h
EMASK_UNCOR_PEX	148h		1C8h
EMASK_COR_PEX	14Ch		1CCh
EMASK_RP_PEX	150h		1D0h
PEX_FAT_FERR	154h		1D4h
PEX_NF_COR_FERR	158h		1D8h
PEX_FAT_NERR	15Ch		1DCh
PEX_NF_COR_NERR	160h		1E0h
	164h		1E4h
PEX_UNIT_FERR	168h		1E8h
PEX_UNIT_NERR	16Ch		1ECh
	170h		1F0h
	174h		1F4h
	178h		1F8h
	17Ch		1FCh

Table 4-11. Device 1-7, Function 0: PCI Express Error Injection and Mask Registers

	200h		280h
	204h		284h
	208h		288h
	20Ch		28Ch
	210h		290h
	214h		294h
	218h		298h
	21Ch		29Ch
	220h		2A0h
	224h		2A4h
	228h		2A8h
	22Ch		2ACh
	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
PEXEINJCTL	278h		2F8h
PEXEINJMSK	27Ch		2FCh

Table 4-12. Device 1-7, Function 0: PCI Express IBIST Registers

	300h	PEX[7:1]IBCTL			380h
	304h	PEX[7:1]IBSYMBUF			384h
	308h	PEX[7:1]IBEXTCTL			388h
	30Ch	PEX[7:1]IBLOOPCNT		PE[7:1]IBDLYSYM	38Ch
	310h	PEX[7:1]IB LNS3	PEX[7:1]IB LNS2	PEX[7:1]IB LNS1	PEX[7:1]IB LNS0	390h
	314h					394h
	318h					398h
	31Ch					39Ch
	320h					3A0h
	324h					3A4h
	328h					3A8h
	32Ch					3ACh
	330h					3B0h
	334h					3B4h
	338h					3B8h
	33Ch					3BCh
	340h					3C0h
	344h					3C4h
	348h					3C8h
	34Ch					3CCh
	350h					3D0h
	354h					3D4h
	358h					3D8h
	35Ch					3DCh
	360h					3E0h
	364h					3E4h
	368h					3E8h
	36Ch					3ECh
	370h					3F0h
	374h					3F4h
	378h					3F8h
	37Ch					3FCh

Table 4-13. Device 8, Function 0, Intel® QuickData Technology DMA Configuration Map

DID			VID			00h			80h
PCISTS			PCICMD			04h			84h
CCR					RID	08h			88h
	HDR					0Ch			8Ch
IQD_BAR						10h			90h
						14h			94h
						18h			98h
						1Ch			9Ch
						20h			A0h
						24h			A4h
						28h			A8h
SID			SVID			2Ch			ACh
						30h			B0h
					CAPPTR	34h			B4h
						38h			B8h
			INTP		INTL	3Ch			BCh
						40h			C0h
						44h			C4h
						48h			C8h
						4Ch			CCh
PMCAP						50h			D0h
PMCSR						54h			D4h
MSICTRL			MSINXPTR		MSICAPID	58h			D8h
MSIAR						5Ch			DCh
MSIDR						60h			E0h
						64h			E4h
						68h			E8h
PEXCAP			PEXNXPTR		PEXCAPID	6Ch			ECh
PEXDEVCAP						70h			F0h
PEXDEVSTS			PEXDEVCTRL			74h			F4h
						78h			F8h
						7Ch			FCh

These MMIO registers are placed in the configuration ring for CPU MMIO accesses but the TOE device will be able to read/write to this space using the fast, bypass inbound access method. Note that this MMIO space is not accessible by MMCFG or CFC/CF8 from the FSB and the mapping to the configuration space is an internal Intel® 7300 Chipset feature. It can be accessed by JTAG/SMBUS however.

Table 4-14. Device 9, Function 0: AMB Switching Window Registers

Route out to AMB per AMBSELECT register	0h - FFh

This function is only accessible by SMBus or JTAG. Accesses from other sources will be routed to ESI and get master aborted

Access to this function is routed out to FB-DIMM channel as per AMBSELECT register subject to AMBPRESENT register settings.

Table 4-15. Device 16, Function 0: Processor Bus, Boot, and Interrupt

DID		VID		00h	LAPIC_STATE0	80h
				04h	LAPIC_STATE1	84h
CCR			RID	08h	LAPIC_STATE2	88h
	HDR			0Ch	LAPIC_STATE3	8Ch
				10h	LAPIC_STATE4	90h
				14h	LAPIC_STATE5	94h
				18h	LAPIC_STATE6	98h
				1Ch	LAPIC_STATE7	9Ch
				20h	LAPIC_STATE8	A0h
				24h	LAPIC_STATE9	A4h
				28h	LAPIC_STATE10	A8h
SID		SVID		2Ch	LAPIC_STATE11	ACh
				30h	LAPIC_STATE12	B0h
				34h	LAPIC_STATE13	B4h
				38h	LAPIC_STATE14	B8h
				3Ch	LAPIC_STATE15	BCh
CPURSTCAPTMR		SYRE		40h	BOFL0	C0h
POC				44h	BOFL1	C4h
AMBASE				48h	BOFL2	C8h
				4Ch	BOFL3	CCh
AMR				50h	SPAD0	D0h
MAXDIMM PERCH	MAXCH	AMBSELECT		54h	SPAD1	D4h
PAM2	PAM1	PAM0		58h	SPAD2	D8h
PAM6	PAM5	PAM4	PAM3	5Ch	SPAD3	DCh
EXSMRTOP	EXSMRC	SMRAMC	EXSMRAMC	60h	SPADS0	E0h
HECBASE				64h	SPADS1	E4h
				68h	SPADS2	E8h
REDIRCTL				6Ch	SPADS3	ECh
FSBC0				70h	COHGLOBC	F0h
				74h	COHC	F4h
FSBC1				78h	COHC2	F8h
				7Ch	COHS	FCh

Table 4-16. Device 16, Function 0:

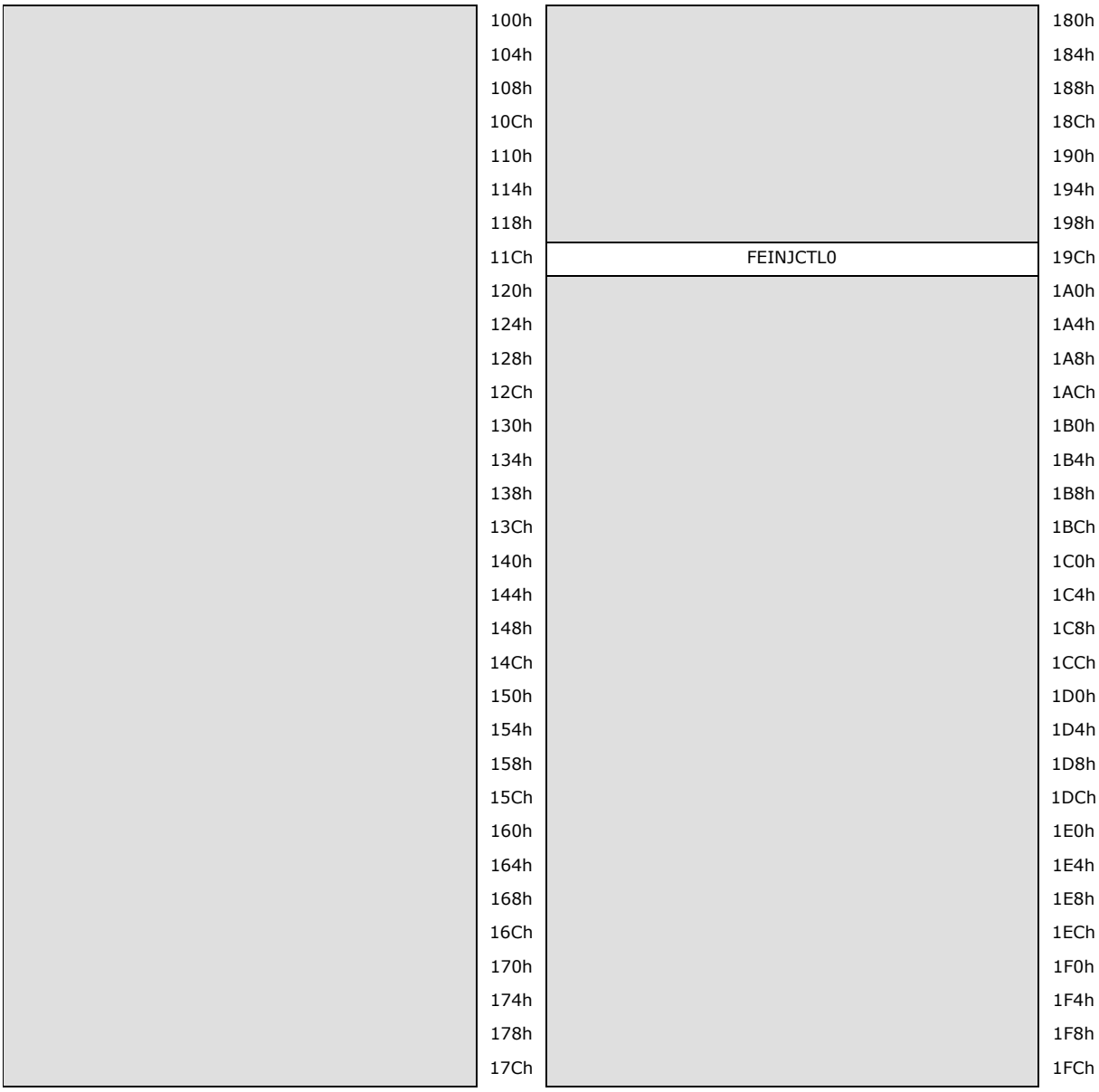


Table 4-17. Device 16, Function 1: Memory Branch Map, Control, Errors

DID		VID		00h		MIR0	80h
				04h		MIR1	84h
CCR			RID	08h		MIR2	88h
	HDR			0Ch		AMIR0	8Ch
				10h		AMIR1	90h
				14h		AMIR2	94h
				18h	FERR_FAT_FBD		98h
				1Ch	NERR_FAT_FBD		9Ch
				20h	FERR_NF_FBD		A0h
				24h	NERR_NF_FBD		A4h
				28h	EMASK_FBD		A8h
SID		SVID		2Ch	ERR0_FBD		ACh
				30h	ERR1_FBD		B0h
				34h	ERR2_FBD		B4h
				38h	MCERR_FBD		B8h
				3Ch	NRECMEMA		BCh
MC				40h	NRECMEMB		C0h
MS				44h	NRECFBDA		C4h
DRTA				48h	NRECFBDB		C8h
DRTB				4Ch	NRECFBDC		CCh
ERRPER				50h	NRECFBDD		D0h
	DDRFRQ			54h	NRECFBDE		D4h
MCA				58h		NRECFBDF	D8h
MCCTRL				5Ch	REDMEMA		DCh
			GBLACT	60h		RECMEMA	E0h
THRTCTRL	THRTHI	THRTMID	THRTLOW	64h	RECMEMB		E4h
THRTSTS1		THRTSTS0		68h	RECFBDA		E8h
		TOLM		6Ch	RECFBDB		ECh
				70h	RECFBDC		F0h
NRECCFGLOG				74h	RECFBDD		F4h
RECCFGLOG				78h	RECFBDE		F8h
REDMEMB				7Ch		RECFBDF	FCh

Table 4-18. Device 16, Function 1: Memory Error Injection Mask Registers

MEM0EINJMSK0		100h		180h
MEM1EINJMSK1	MEM0EINJMSK1	104h		184h
MEM1EINJMSK0		108h		188h
		10Ch		18Ch
		110h		190h
		114h		194h
		118h		198h
		11Ch		19Ch
		120h		1A0h
		124h		1A4h
		128h		1A8h
		12Ch		1ACh
		130h		1B0h
		134h		1B4h
		138h		1B8h
		13Ch		1BCh
		140h		1C0h
		144h		1C4h
		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

Table 4-19. Device 16, Function 2: RAS

DID		VID		00h			80h
				04h			84h
CCR			RID	08h			88h
	HDR			0Ch			8Ch
				10h			90h
				14h			94h
				18h			98h
				1Ch			9Ch
				20h			A0h
				24h			A4h
				28h			A8h
SID		SVID		2Ch			ACh
				30h	NRECSF		B0h
				34h			B4h
				38h	RECSF		B8h
				3Ch			BCh
FERR_Global_LO				40h	NERR_NF_INT	NERR_FAT_INT	C0h	FERR_NF_INT	FERR_FAT_INT
NERR_Global				44h	NRECINT		C4h
FERR_Global_HI				48h	RECINT		C8h
				4Ch		EMASK_INT	CCh
				50h	MCERR_INT	ERR2_INT	ERR1_INT	ERR0_INT	D0h
				54h			D4h
				58h			D8h
				5Ch			DCh
				60h			E0h
				64h			E4h
				68h			E8h
				6Ch			ECh
				70h			F0h
				74h			F4h
				78h			F8h
				7Ch			FCh

Table 4-20. Device 16, Function 3 : Processor Buses 2 and 3, More Interrupts

DID	VID	00h	LAPIC_STATE16	80h
		04h	LAPIC_STATE17	84h
CCR	RID	08h	LAPIC_STATE18	88h
HDR		0Ch	LAPIC_STATE19	8Ch
		10h	LAPIC_STATE20	90h
		14h	LAPIC_STATE21	94h
		18h	LAPIC_STATE22	98h
		1Ch	LAPIC_STATE23	9Ch
		20h	LAPIC_STATE24	A0h
		24h	LAPIC_STATE25	A4h
		28h	LAPIC_STATE26	A8h
SID	SVID	2Ch	LAPIC_STATE27	ACh
		30h	LAPIC_STATE28	B0h
		34h	LAPIC_STATE29	B4h
		38h	LAPIC_STATE30	B8h
		3Ch	LAPIC_STATE31	BCh
		40h		C0h
		44h		C4h
		48h		C8h
		4Ch		CCh
INTXSTAT0		50h		D0h
INTXSTAT1		54h		D4h
INTxROUTECTL		58h		D8h
		5Ch		DCh
		60h		E0h
		64h		E4h
		68h		E8h
INTRSTS		6Ch		ECh
FSBC2		70h		F0h
		74h		F4h
FSBC3		78h		F8h
		7Ch		FCh

Table 4-21. Device 17, Function 0: COH and Data Manager Cluster Registers

DID			VID	00h											80h
				04h											84h
CCR			RID	08h											88h
	HDR			0Ch											8Ch
				10h	SFDEF0										90h
				14h											94h
				18h	SFINIT0										98h
				1Ch											9Ch
				20h											A0h
				24h											A4h
				28h	SFERRINJ0										A8h
SID			SVID	2Ch											ACh
				30h											B0h
				34h											B4h
				38h											B8h
				3Ch											BCh
NERR_NF_FSB0	NERR_FAT_FSB0	FERR_NF_FSB0	FERR_FAT_FSB0	40h	NERR_NF_FSB1	NERR_FAT_FSB1	FERR_NF_FSB1	FERR_FAT_FSB1							C0h
NRECFSB0				44h	NRECFSB1										C4h
RECFSB0				48h	RECFSB1										C8h
NRECADDRL0				4Ch	NRECADDRL1										CCh
EMASK_FSB0			NRECADDRH0	50h	EMASK_FSB1			NRECADDRH1							D0h
ERR1_FSB0		ERR0_FSB0		54h	ERR1_FSB1		ERR0_FSB1								D4h
MCERR_FSB0		ERR2_FSB0		58h	MCERR_FSB1		ERR2_FSB1								D8h
				5Ch											DCh
				60h											E0h
				64h											E4h
				68h											E8h
				6Ch											ECh
				70h											F0h
				74h											F4h
				78h											F8h
				7Ch											FCh

Table 4-22. Device 17, Function 0: COH and Data Manager Cluster Registers

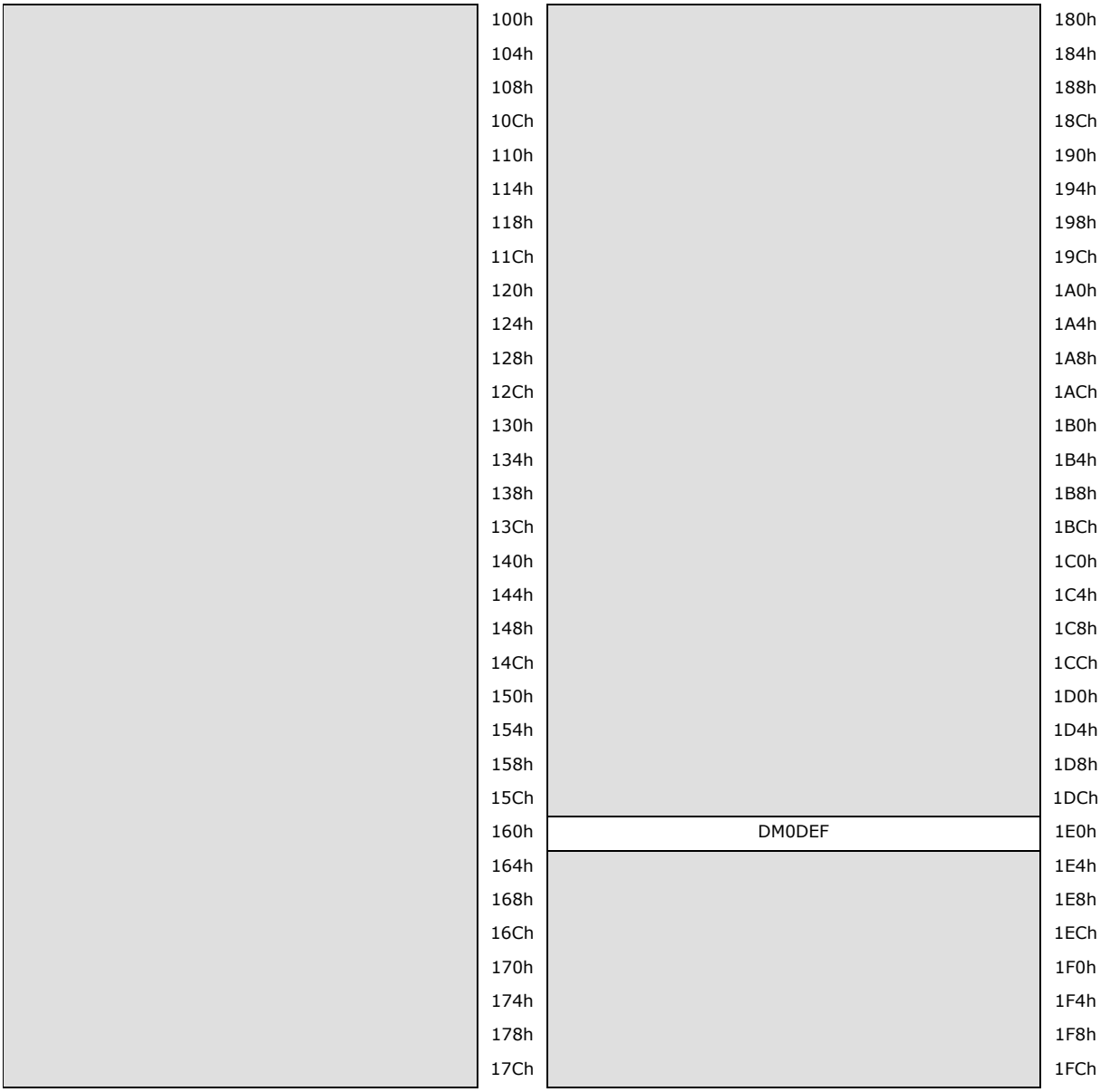


Table 4-23. Device 17, Function 3: Data Manager Cluster Registers

DID		VID		00h					80h
				04h					84h
CCR			RID	08h					88h
	HDR			0Ch					8Ch
				10h	SFDEF1				90h
				14h					94h
				18h	SFINIT1				98h
				1Ch					9Ch
				20h					A0h
				24h					A4h
				28h	SFERRINJ1				A8h
SID		SVID		2Ch					ACh
				30h					B0h
				34h					B4h
				38h					B8h
				3Ch					BCh
NERR_NF_FSB2	NERR_FAT_FSB2	FERR_NF_FSB2	FERR_FAT_FSB2	40h	NERR_NF_FSB3	NERR_FAT_FSB3	FERR_NF_FSB3	FERR_FAT_FSB3	C0h
NRECFSB2				44h	NRECFSB3				C4h
RECFSB2				48h	RECFSB3				C8h
NRECADDRL2				4Ch	NRECADDRL3				CCh
EMASK_FSB2			NRECADDRH2	50h	EMASK_FSB3			NRECADDRH3	D0h
ERR1_FSB2		ERR0_FSB2		54h	ERR1_FSB3		ERR0_FSB3		D4h
MCERR_FSB2		ERR2_FSB2		58h	MCERR_FSB3		ERR2_FSB3		D8h
				5Ch					DCh
				60h					E0h
				64h					E4h
				68h					E8h
				6Ch					ECh
				70h					F0h
				74h					F4h
				78h					F8h
				7Ch					FCh

Table 4-24. Device 17, Function 3: Data Manager Cluster Registers

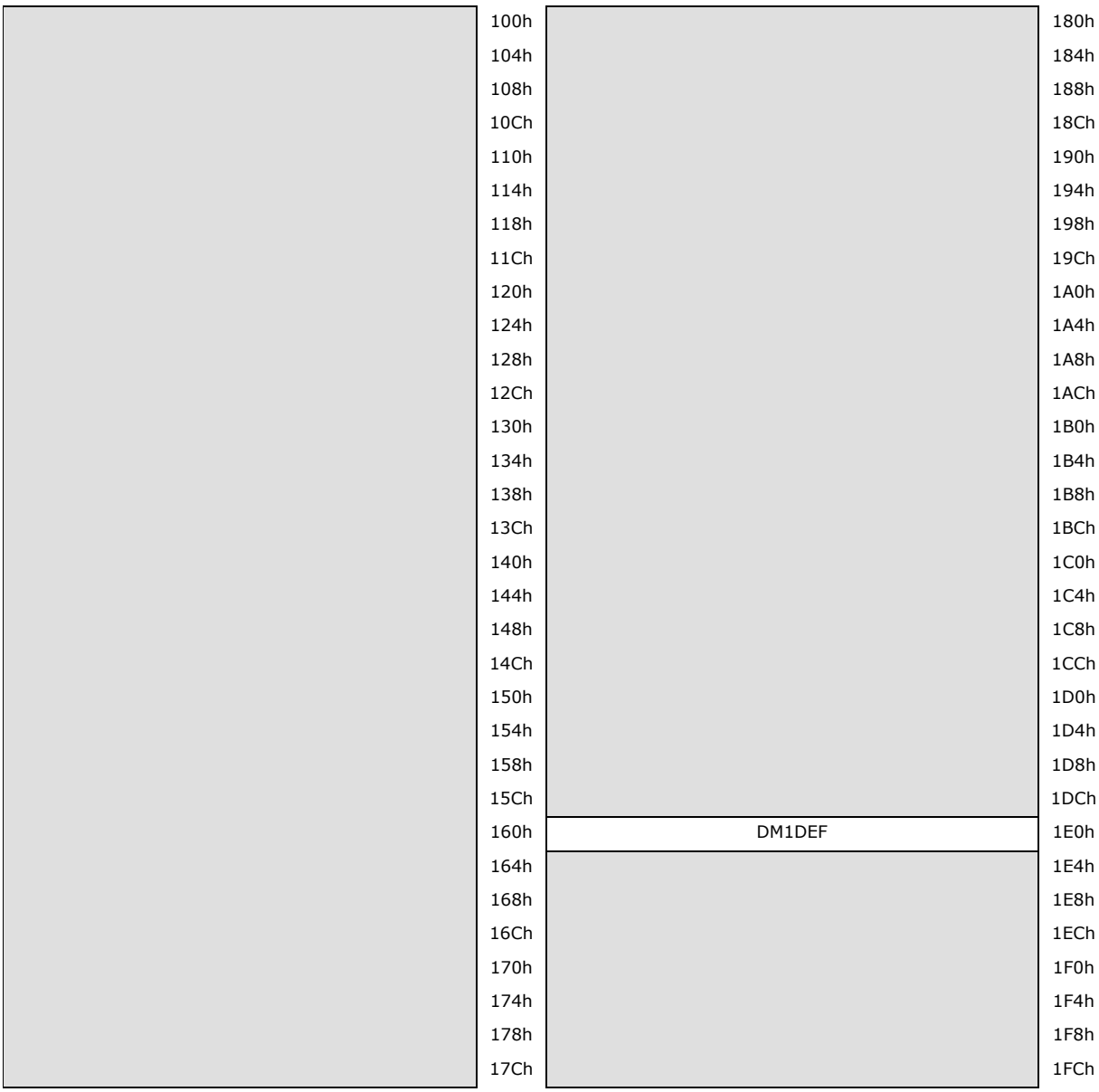


Table 4-25. Device 19, Function 0: Debug Block Registers

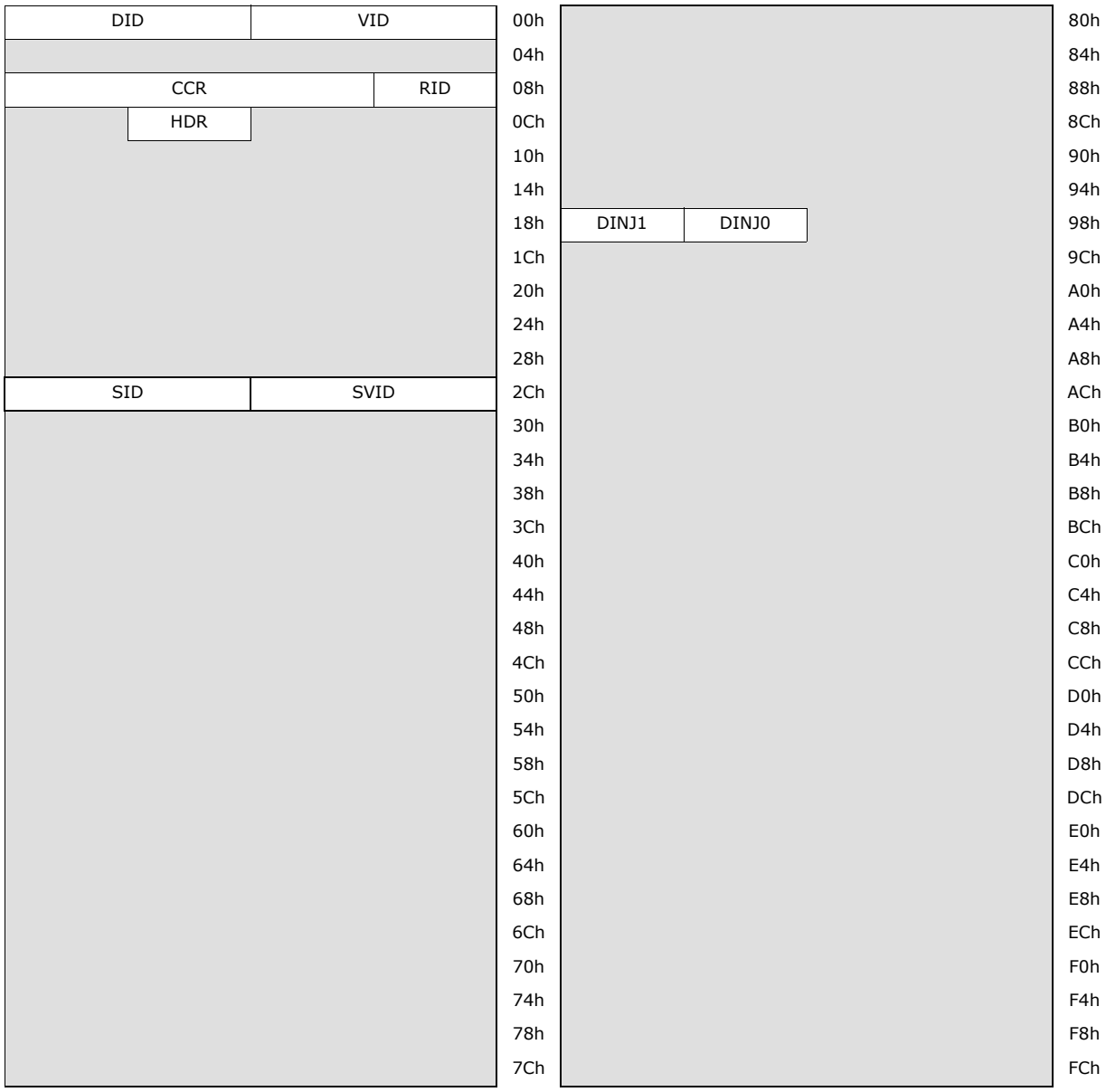


Table 4-26. Device 19, Function 0: PCI Express Global Control Registers

	100h		180h
	104h		184h
	108h		188h
	10Ch		18Ch
	110h		190h
	114h		194h
	118h		198h
	11Ch		19Ch
	120h		1A0h
	124h		1A4h
	128h		1A8h
	12Ch		1ACh
	130h		1B0h
	134h		1B4h
	138h		1B8h
	13Ch		1BCh
	140h		1C0h
	144h		1C4h
	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
PEXGCTRL	17Ch		1FCh

Table 4-27. Device 21, 22, Function 0: FBD DIMM Map, Control, RAS

DID			VID	00h	MTR4		MTR0		80h
				04h	MTR5		MTR1		84h
CCR			RID	08h	MTR6		MTR2		88h
	HDR			0Ch	MTR7		MTR3		8Ch
				10h	DMIR0				90h
				14h	DMIR1				94h
				18h	DMIR2				98h
				1Ch	DMIR3				9Ch
				20h	DMIR4				A0h
				24h	DMIR5				A4h
				28h	DMIR6				A8h
SID			SVID	2Ch					ACh
				30h					B0h
				34h					B4h
				38h					B8h
				3Ch					BCh
SPCPS	SPCPC			40h	BADRAMA				C0h
FBDICMD1	FBDICMD0	FBDLVL1	FBDLVL0	44h	BADRAMB				C4h
FBDST				48h	BADRAMC				C8h
FBDHPC				4Ch	BADCNTA				CCh
FBDRST				50h	BADCNTB				D0h
				54h		FBDPSCNTRL	FBDSBTXCFG1	FBDSBTXCFG0	D4h
FBDISTS1			FBDISTS0	58h					D8h
				5Ch					DCh
				60h	CERRCNTA0				E0h
AMBPRESENT1			AMBPRESENT0	64h	CERRCNTB0				E4h
				68h	CERRCNTC0				E8h
				6Ch	CERRCNTD0				ECh
				70h	CERRCNTA1				F0h
SPD1			SPD0	74h	CERRCNTB1				F4h
SPDCMD0				78h	CERRCNTC1				F8h
SPDCMD1				7Ch	CERRCNTD1				FCh

Table 4-28. Device 21, Function 0: FB-DIMM Channel 0 IBIST Registers

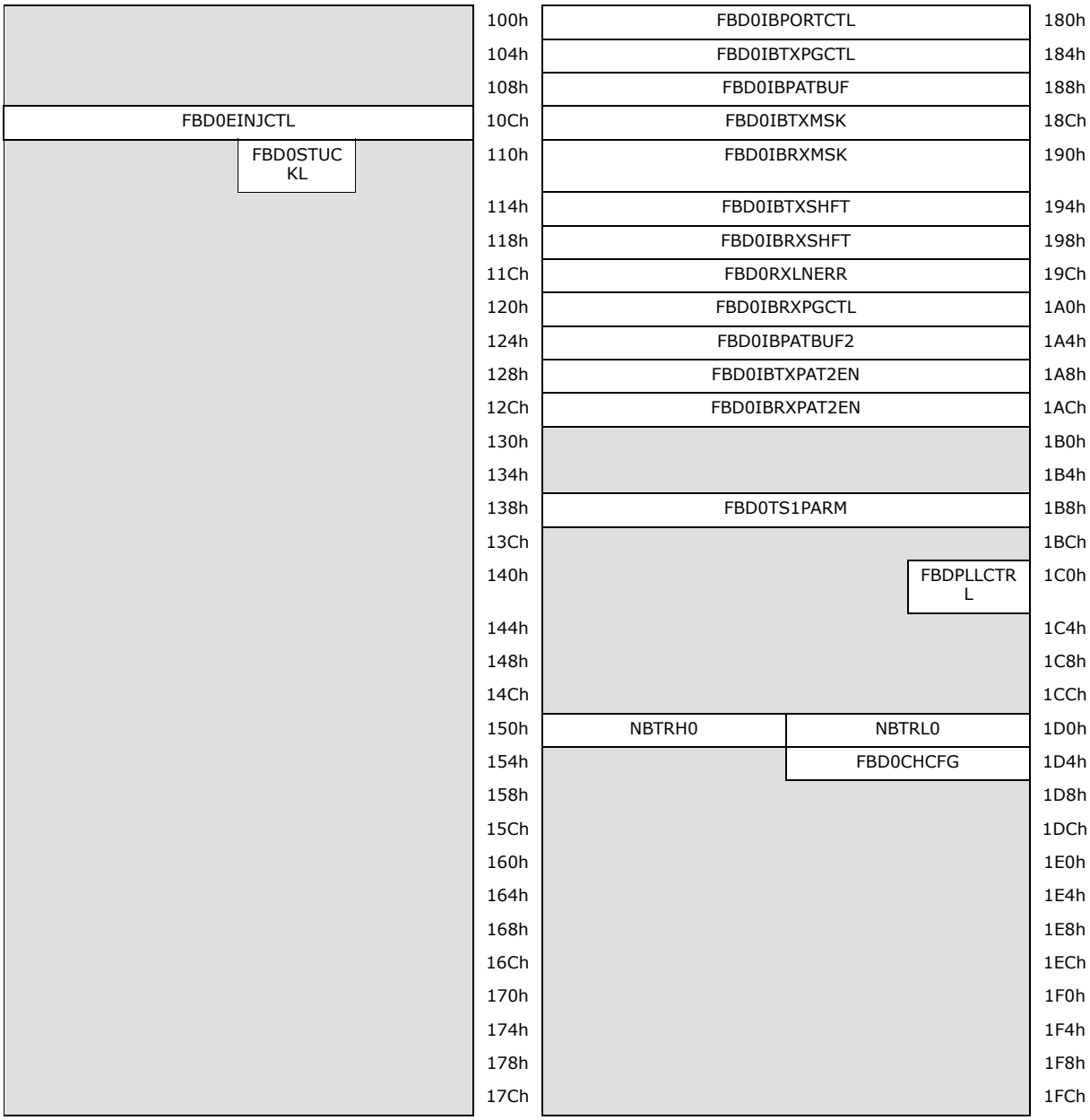


Table 4-29. Device 21, Function 0: FB-DIMM Channel 1 IBIST Registers

	200h	FBD1IBPORTCTL		280h
	204h	FBD1IBTXPGCTL		284h
	208h	FBD1IBPATBUF		288h
FBD1EINJCTL	20Ch	FBD1IBTXMSK		28Ch
FBD1STUCKL	210h	FBD1IBRXMSK		290h
	214h	FBD1IBTXSHFT		294h
	218h	FBD1IBRXSHFT		298h
	21Ch	FBD1RXLNERR		29Ch
	220h	FBD1IBRXPGCTL		2A0h
	224h	FBD1IBPATBUF2		2A4h
	228h	FBD1IBTXPAT2EN		2A8h
	22Ch	FBD1IBRXPAT2EN		2ACh
	230h			2B0h
	234h			2B4h
	238h	FBD1TS1PARM		2B8h
	23Ch			2BCh
	240h			2C0h
	244h			2C4h
	248h			2C8h
	24Ch			2CCh
	250h	NBTRH1	NBTRL1	2D0h
	254h		FBD1CHCFG	2D4h
	258h			2D8h
	25Ch			2DCh
	260h			2E0h
	264h			2E4h
	268h			2E8h
	26Ch			2ECh
	270h			2F0h
	274h			2F4h
	278h			2F8h
	27Ch			2FCh

Table 4-30. Device 22, Function 0: FB-DIMM Channel 2 IBIST Registers

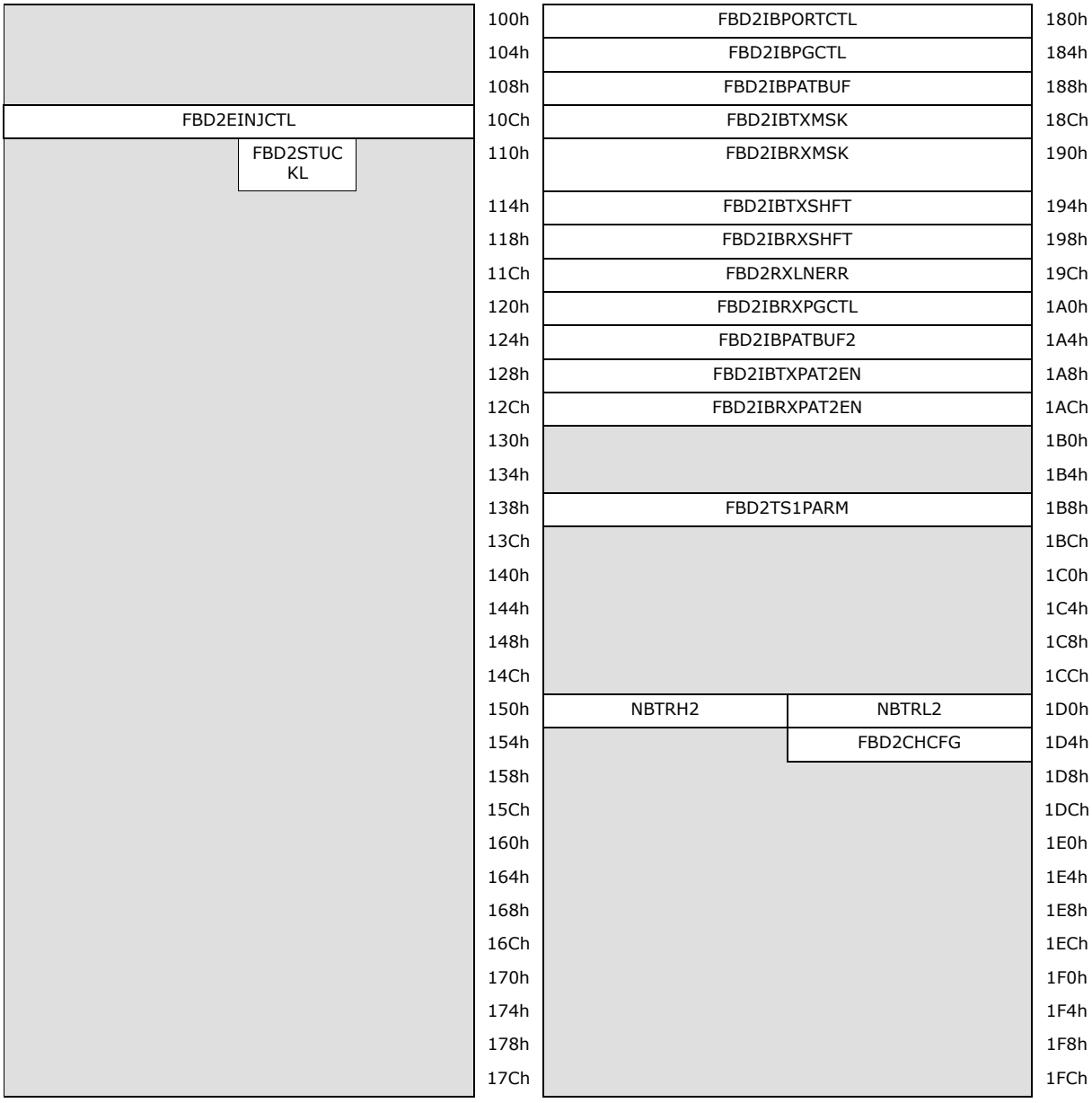


Table 4-31. Device 22, Function 0: FB-DIMM Channel 3 IBIST Registers

	200h	FBD3IBPORTCTL		280h
	204h	FBD3IBPGCTL		284h
	208h	FBD3IBPATBUF		288h
FBD3EINJCTL	20Ch	FBD3IBTXMSK		28Ch
FBD3STUCKL	210h	FBD3IBRXMSK		290h
	214h	FBD3IBTXSHFT		294h
	218h	FBD3IBRXSHFT		298h
	21Ch	FBD3RXLNERR		29Ch
	220h	FBD3IBRXPGCTL		2A0h
	224h	FBD3IBPATBUF2		2A4h
	228h	FBD3IBTXPAT2EN		2A8h
	22Ch	FBD3IBRXPAT2EN		2ACh
	230h			2B0h
	234h			2B4h
	238h	FBD3TS1PARM		2B8h
	23Ch			2BCh
	240h			2C0h
	244h			2C4h
	248h			2C8h
	24Ch			2CCh
	250h	NBTRH3	NBTRL3	2D0h
	254h		FBD3CHCFG	2D4h
	258h			2D8h
	25Ch			2DCh
	260h			2E0h
	264h			2E4h
	268h			2E8h
	26Ch			2ECh
	270h			2F0h
	274h			2F4h
	278h			2F8h
	27Ch			2FCh

4.8 Register Definitions

4.8.1 PCI Standard Registers

These registers appear in every function of every device.

4.8.1.1 VID: Vendor Identification Register

This register identifies Intel as the manufacturer of the Intel® 7300 Chipset. Writes to this register have no effect.

Device: 0-8
Function: 0
Offset: 00h

Device: 16
Function: 0-3
Offset: 00h

Device: 17
Function: 0,3
Offset: 00h

Device: 19,21,22
Function: 0
Offset: 00h

Bit	Attr	Default	Description
15:0	RO	8086h	**Vendor Identification Number** The value assigned to Intel.

4.8.1.2 DID: Device Identification Register

This register combined with the Vendor Identification register uniquely identifies the Intel® 7300 Chipset Function in the event that a driver is required. Writes to this register have no effect. See Table Table 4-2 for the DID of each function.

Device: 0-8
Function: 0
Offset: 02h

Device: 16
Function: 0-3
Offset: 02h

Device: 17
Function: 0,3
Offset: 02h

Device: 19,21,22
Function: 0
Offset: 02h

Bit	Attr	Default	Description
15:0	RO	See Table 4-2	**Device Identification Number** Identifies each function of the Intel® 7300 Chipset

4.8.2 Revision ID (RID): Background

The Revision ID (RID) is a traditional 8-bit Read Only (RO) register located at offset 08h in the standard PCI header of every PCI/PCI Express compatible device and function. Historically, a new value for RID, which describes the current revision number of the device/component, is assigned for Intel chipsets for every stepping. However,

there is a need to provide an alternative value for software compatibility when a particular driver or patch unique to that stepping or an earlier stepping is required to prevent Windows software for instance from flagging differences in RID during device enumeration. The solution is to implement a mechanism to read one of two possible values from the RID register:

1. **Stepping Revision ID** (SRID): This is the default power on value for mask/metal steppings
2. **Compatible Revision ID** (CRID): The CRID functionality gives BIOS the flexibility to load OS drivers optimized for a previous revision of the silicon instead of the current revision of the silicon in order to reduce driver updates and minimize changes to the OS image for minor optimizations to the silicon for yield improvement, or feature enhancement reasons that do not negatively impact the OS driver functionality.

Reading the RID in the MCH returns either the SRID or CRID depending on the state of a register select flip-flop. Following reset, the register select flip flop is reset and the SRID is returned when the RID is read at offset 08h. The SRID value reflects the actual product stepping. To select the CRID value, BIOS/configuration software writes a key value of 69h to Bus 0, Device 0, Function 0 (ESI port) of the MCH's RID register at offset 08h. This sets the SRID/CRID register select flip-flop and causes the CRID to be returned when the RID is read at offset 08h.

The RID register in the ESI port (Bus 0 device 0 Function 0) is a "write-once" sticky register and gets locked after the first write. This causes the CRID to be returned on all subsequent RID register reads. Software should read and save all device SRID values by reading MCH device RID registers before setting the SRID/CRID register select flip flop.

The RID values for all devices and functions in MCH are controlled by the SRID/CRID register select flip flop, thus writing the key value (69h) to the RID register in Bus 0, Device 0, Function 0 sets all MCH device RID registers to return the CRID. Writing to the RID register of other devices has no effect on the SRID/CRID register select flip-flop. Only a power good reset can change the RID selection back to SRID.

4.8.2.1 Stepping Revision ID (SRID)

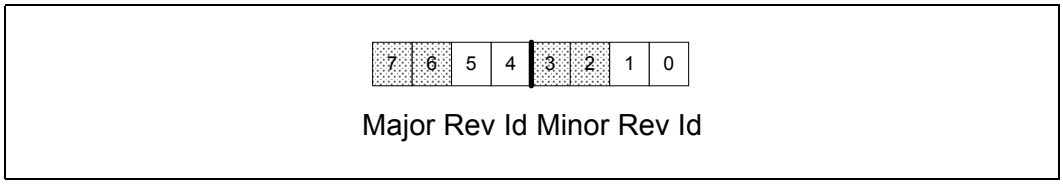
The SRID is a 4-bit hardwired value assigned by Intel, based on product's stepping. The SRID is not a directly addressable PCI register. The SRID value is reflected through the RID register when appropriately addressed. The 4 bits of the SRID are reflected as the two least significant bits of the major and minor revision field respectively. See Figure 4-3.

4.8.2.2 Compatible Revision ID (CRID)

The CRID is an 4-bit hardwired value assigned by Intel during manufacturing process. Normally, the value assigned as the CRID will be identical to the SRID value of a previous stepping of the product with which the new product is deemed "compatible".

The CRID is not a directly addressable PCI register. The CRID value is reflected through the RID register when appropriately addressed.The 4 bits of the CRID are reflected as the two least significant bits of the major and minor revision field respectively. See Figure 4-3.

Figure 4-3. Intel® 7300 Chipset Implementation of RID Based on SRID and CRID Registers



4.8.2.3 RID: Revision Identification Register

This register contains the revision number of the Intel® 7300 Chipset.

Device: 0-8
Function: 0
Offset: 08h

Device: 16
Function: 0-3
Offset: 08h

Device: 17
Function: 0,3
Offset: 08h

Device: 19,21,22
Function: 0
Offset: 08h

Bit	Attr	Default	Description
7:4	(DEV 0) =RWOST others = RO	0h	**Major Revision** Steppings which require all masks to be regenerated. 0000: A stepping for Intel® 7300 Chipset 0001: B stepping for Intel® 7300 Chipset 0010: C stepping for Intel® 7300 Chipset ..... This field should be set appropriately by the hardware based on the current stepping even though the default value may indicate otherwise
3:0	(DEV 0) =RWOST others = RO	0h	**Minor Revision** Incremented for each stepping which does not modify all masks. Reset for each major revision. 0x0: x0 stepping 0x1: x1 stepping 0x2: x2 stepping .... This field should be set appropriately by the hardware based on the current stepping even though the default value may indicate otherwise

4.8.2.4 CCR: Class Code Register

This register contains the Class Code for the device.

Device[a]: 0-7
Function: 0
Offset: 09h

Device: 16
Function: 0-3
Offset: 09h

Device: 17
Function: 0,3
Offset: 09h

Device: 19,21,22
Function: 0
Offset: 09h

Bit	Attr	Default	Description
23:16	RO	06h	**Base Class**. This field indicates the general device category. For the Intel® 7300 Chipset, this field is hardwired to 06h, indicating it is a "Bridge Device".
15:8	RO	(DEV1-7) = 04h others = 00h	Sub-Class. This field qualifies the Base Class, providing a more detailed specification of the device function. For PCI Express Devices 1,2,3,4,5,6,7 default is 04h, indicating "PCI to PCI Bridge" For all other Devices: 0,9,10,12,14,16,17,18,19,21,22 default is 00h, indicating "Host Bridge". See footnote[a] for Intel® QuickData Technology device CCR.
7:0	RO	00h	Register-Level Programming Interface. This field identifies a specific programming interface (if any), that device independent software can use to interact with the device. There are no such interfaces defined for "Host Bridge" types, and this field is hardwired to 00h.

Notes:

a. The Intel® QuickData Technology CCR for device 8 is defined separately in Section 4.8.27.3, "CCR: Class Code Register" on page 269.

4.8.2.5 HDR: Header Type Register

This register identifies the header layout of the configuration space.

Device: 0-8
Function: 0
Offset: 0Eh

Device: 16
Function: 0-3
Offset: 0Eh

Device: 17
Function: 0,3
Offset: 0Eh

Device: 19,21,22
Function: 0
Offset: 0Eh

Bit	Attr	Default	Description
7	RO	(DEV16, 17) = 1h} others =0h	**Multi-function Device**. Selects whether this is a multi-function device, that may have alternative configuration layouts. This bit is hardwired to 0 for all devices in the Intel® 7300 Chipset except for: Device 16, function 0-3, which is set to '1'. Device 17, function 0,3, which is set to '1'

Device: 0-8
Function: 0
Offset: 0Eh

Device: 16
Function: 0-3
Offset: 0Eh

Device: 17
Function: 0,3
Offset: 0Eh

Device: 19,21,22
Function: 0
Offset: 0Eh

Bit	Attr	Default	Description
6:0	RO	(DEV1-7) = 01h others = 00h	**Configuration Layout**. This field identifies the format of the configuration header layout for a PCI-to-PCI bridge from bytes 10h through 3Fh. For PCI Express Devices 1,2,3,4,5,6,7 default is 01h, indicating "PCI to PCI Bridge" For all other Devices: 0,8,9,16,17,19,21,22 default is 00h, indicating a conventional type 00h PCI header

4.8.2.6 SVID: Subsystem Vendor Identification Register

This register identifies the manufacturer of the system. This 16-bit register combined with the Device Identification Register uniquely identify any PCI device. They appear in every function except the PCI Express functions.

Device: 0, 8
Function: 0
Offset: 2Ch

Device: 16
Function: 0-3
Offset: 2Ch

Device: 17
Function: 0,3
Offset: 2Ch

Device: 19,21,22
Function: 0
Offset: 2Ch

Bit	Attr	Default	Description
15:0	RWO	8086h	**Vendor Identification Number**. The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

A write to any of the above registers on Intel® 7300 Chipset will write to all of them.

4.8.2.7 SID: Subsystem Identity

This register identifies the system. They appear in every function except the PCI Express functions.

Device: 0,8
Function: 0
Offset: 2Eh

Device: 16
Function: 0-3
Offset: 2Eh

Device: 17
Function: 0, 3
Offset: 2Eh

Device: 19,21,22
Function: 0
Offset: 2Eh

Bit	Attr	Default	Description
15:0	RWO	8086h	**Subsystem Identification Number:** The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.8.3 Address Mapping Registers

These registers control transaction routing to one of the three interface types (Memory, PCI Express, or ESI) based on address. Routing to particular ports of a given interface type are defined by the following registers:

Table 4-32. Address Mapping Registers

Interface type	Address Routing registers
Memory	**MIR, AMIR, PAM, SMRAM, EXSMRC, EXSMRAMC, TOLM, EXSMRTOP, AMBASE, AMR**
PCI Express	MBASE/MLIM (devices 1-7) PMBASE/PMLIM (devices 1-7) PMBU/PMBL (devices 1-7) IOBASE/IOLIM (devices 1-7) SBUSN,SUBUSN (devices 1-7) BCTRL (devices 1-7) HECBASE, PEXCMD (devices 1-7)
ESI	Subtractive decode[a] (device 0)

Notes:

a. Any request not falling in the above ranges will be subtractively decoded and sent to Intel® 631xESB/632xESB I/O Controller Hub via the ESI

The Intel® 7300 Chipset allows programmable memory attributes on 13 Legacy memory segments of various sizes in the 640 Kbytes to 1 Mbytes address range. Seven Programmable Attribute Map (PAM) Registers are used to support these features.

Each PAM Register controls one or two regions, typically 16 Kbytes in size

4.8.3.1 PAM0: Programmable Attribute Map 0

This register controls the read, write, and shadowing attributes of the BIOS area from 0F0000h-0FFFFFh.

Two bits are used to specify memory attributes for each memory segment. These bits apply to both host accesses and PCI initiator accesses to the PAM areas. These attributes are:

RE - Read Enable. When RE = 1, the CPU read accesses to the corresponding memory segment are claimed by the Intel® 7300 Chipset and directed to main memory. Conversely, when RE = 0, the host read accesses are directed to ESI (Intel® 631xESB/632xESB I/O Controller Hub) to be directed to the PCI bus.

WE - Write Enable. When WE = 1, the host write accesses to the corresponding memory segment are claimed by the Intel® 7300 Chipset and directed to main memory. Conversely, when WE = 0, the host write accesses are directed to ESI (Intel® 631xESB/632xESB I/O Controller Hub) to be directed to the PCI bus.

The RE and WE attributes permit a memory segment to be Read Only, Write Only, Read/Write, or disabled. For example, if a memory segment has RE = 1 and WE = 0, the segment is Read Only.

Device: 16
Function: 0
Offset: 59h

Bit	Attr	Default	Description
7:6	RV	00	*Reserved*
5:4	RW	00	**ESIENABLE0: 0F0000-0FFFFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0F0000 to 0FFFFF. Bit5 = Write enable, Bit4 = Read enable. Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM
3:0	RV	0h	*Reserved*

4.8.3.2 PAM1: Programmable Attribute Map 1

This register controls the read, write, and shadowing attributes of the BIOS areas from 0C0000h-0C7FFFh.

Device: 16
Function: 0
Offset: 5Ah

Bit	Attr	Default	Description
7:6	RV	00	*Reserved*
5:4	RW	00	**ESIENABLE1: 0C4000-0C7FFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0C4000 to 0C7FFF Bit5 = Write enable, Bit4 = Read enable. Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM

Device: 16 Function: 0 Offset: 5Ah			
Bit	**Attr**	**Default**	**Description**
3:2	RV	00	*Reserved*
1:0	RW	00	**LOENABLE1: 0C0000-0C3FFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0C0000 to 0C3FFF. Bit1 = Write enable, Bit0 = Read enable Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM

4.8.3.3 PAM2: Programmable Attribute Map 2

This register controls the read, write, and shadowing attributes of the BIOS areas from 0C8000h-0CFFFFh.

Device: 16 Function: 0 Offset: 5Bh			
Bit	**Attr**	**Default**	**Description**
7:6	RV	00	*Reserved*
5:4	RW	00	**ESIENABLE2: 0CC000-0CFFFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0CC000-0CFFFF. Bit5 = Write enable, Bit4 = Read enable. Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM
3:2	RV	00	*Reserved*
1:0	RW	00	**LOENABLE2: 0C8000-0CBFFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0C8000-0CBFFF. Bit1 = Write enable, Bit0 = Read enable Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM

4.8.3.4 PAM3: Programmable Attribute Map 3

This register controls the read, write, and shadowing attributes of the BIOS areas from 0D0000h-0D7FFFh.

Device: 16
Function: 0
Offset: 5Ch

Bit	Attr	Default	Description
7:6	RV	00	*Reserved*
5:4	RW	00	**ESIENABLE3: 0D4000-0D7FFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0D4000-0D7FFF. Bit5 = Write enable, Bit4 = Read enable. Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM
3:2	RV	00	*Reserved*
1:0	RW	00	**LOENABLE3: 0D0000-0D3FFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0D0000-0D3FFF. Bit1 = Write enable, Bit0 = Read enable Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM

4.8.3.5 PAM4: Programmable Attribute Map 4

This register controls the read, write, and shadowing attributes of the BIOS areas from 0D8000h-0DFFFFh.

Device: 16
Function: 0
Offset: 5Dh

Bit	Attr	Default	Description
7:6	RV	00	*Reserved*
5:4	RW	00	**ESIENABLE4: 0DC000-0DFFFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0DC000-0DFFFF. Bit5 = Write enable, Bit4 = Read enable. Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM
3:2	RV	00	*Reserved*
1:0	RW	00	**LOENABLE4: 0D8000-0DBFFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0D8000-0DBFFF. Bit1 = Write enable, Bit0 = Read enable Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM

4.8.3.6 PAM5: Programmable Attribute Map 5

This register controls the read, write, and shadowing attributes of the BIOS areas from 0E0000h-0E7FFFh.

Device: 16 Function: 0 Offset: 5Eh			
Bit	**Attr**	**Default**	**Description**
7:6	RV	00	*Reserved*
5:4	RW	00	**ESIENABLE5: 0E4000-0E7FFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0E4000-0E7FFF. Bit5 = Write enable, Bit4 = Read enable. Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM
3:2	RV	00	*Reserved*
1:0	RW	00	**LOENABLE5: 0E0000-0E3FFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0E0000-0E3FFF. Bit1 = Write enable, Bit0 = Read enable Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM

4.8.3.7 PAM6: Programmable Attribute Map 6

This register controls the read, write, and shadowing attributes of the BIOS areas from 0E8000h-0EFFFFh.

Device: 16 Function: 0 Offset: 5Fh			
Bit	**Attr**	**Default**	**Description**
7:6	RV	00	*Reserved*
5:4	RW	00	**ESIENABLE6: 0EC000-0EFFFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0EC000-0EFFFF. Bit5 = Write enable, Bit4 = Read enable. Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM
3:2	RV	00	*Reserved*
1:0	RW	00	**LOENABLE6: 0E8000-0EBFFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0E8000-0EBFFF. Bit1 = Write enable, Bit0 = Read enable Encoding Description 00: DRAM Disabled - All accesses are directed to ESI 01: Read Only - All reads are serviced by DRAM. Writes are forwarded to ESI 10: Write Only - All writes are sent to DRAM. Reads are serviced by ESI 11: Normal DRAM Operation - All reads and writes are serviced by DRAM

4.8.3.8 SMRAMC: System Management RAM Control

The SMRAMC register controls how accesses to Compatible and Extended SMRAM spaces are treated. The Open, Close, and Lock bits function only when EXSMRC.G_SMRAME bit is set to a 1. Also, the OPEN bit must be reset before the LOCK bit is set.

Device: 16
Function: 0
Offset: 61h

Bit	Attr	Default	Description
7	RV	0	Reserved
6	RWL	0	**D_OPEN: SMM Space Open** When D_OPEN=1 and D_LCK=0, the SMM space DRAM is made visible even when SMM decode is not active. This is intended to help BIOS initialize SMM space. Software should ensure that D_OPEN=1 and D_CLS=1 are not set at the same time. This register can be locked by D_LCK.
5	RW	0	**D_CLS: SMM Space Closed** When D_CLS = 1 SMM space DRAM is not accessible to data references, even if SMM decode is active. Code references may still access SMM space DRAM. This will allow SMM software to reference through SMM space to update the display even when SMM is mapped over the VGA range. Software should ensure that D_OPEN=1 and D_CLS=1 are not set at the same time. Note that the D_CLS bit only applies to Compatible SMM space.
4	RWL	0	**D_LCK: SMM Space Locked** When D_LCK is set to 1 then D_OPEN is reset to 0 and D_LCK, D_OPEN, C_BASE_SEG, H_SMRAM_EN, all LT.MSEG.BASE, all LT.MSEG.SIZE, ESMMTOP, TSEG_SZ and TSEG_EN become read only. D_LCK can be set to 1 via a normal configuration space write but can only be cleared by a Full Reset. The combination of D_LCK and D_OPEN provide convenience with security. The BIOS can use the D_OPEN function to initialize SMM space and then use D_LCK to "lock down" SMM space in the future so that no application software (or BIOS itself) can violate the integrity of SMM space, even if the program has knowledge of the D_OPEN function.
3	RV	0	*Reserved*
2:0	RO	010	**C_BASE_SEG: Compatible SMM Space Base Segment** This field indicates the location of legacy SMM space. SMM DRAM is not remapped. It is simply made visible if the conditions are right to access SMM space, otherwise the access is forwarded to ESI/VGA. Since the Intel® 7300 Chipset supports only the SMM space between A0000 and BFFFF, this field is hardwired to 010.

4.8.3.9 EXSMRC: Extended System Management RAM Control

The Extended SMRAM register controls the configuration of Extended SMRAM space. The Extended SMRAM (E_SMRAM) memory provides a write-back cacheable SMRAM memory space that is above 1 MByte.

Device: 16
Function: 0
Offset: 62h

Bit	Attr	Default	Description
7	RWL	0	**H_SMRAME: Enable High SMRAM** Controls the SMM memory space location (i.e. above 1 MByte or below 1 MByte) When G_SMRAME is 1 and H_SMRAME is set to 1, the high SMRAM memory space is enabled. SMRAM accesses within the range FEDA_0000h to FEDB_FFFFh are remapped to DRAM addresses within the range 000A_0000h to 000B_FFFFh. Once D_LCK has been set, this bit becomes read only.
6	RV	0	*Reserved*.
5	RV	0	*Reserved*

Device: 16 Function: 0 Offset: 62h			
Bit	**Attr**	**Default**	**Description**
4	RV	0	*Reserved*
3	RWL	0	**G_SMRAME: Global SMRAM Enable** If set to a 1, then Compatible SMRAM functions are enabled, providing 128 KB of DRAM accessible at the A0000h address while in SMM (ADS# with SMM decode). To enable Extended SMRAM function this bit has be set to 1. Refer to the section on SMM for more details. Once D_LCK is set, this bit becomes read only. (Moved from SMRAM bit3)
2:1	RWL	00	**TSEG_SZ: TSEG Size** Selects the size of the TSEG memory block if enabled. Memory from (ESMMTOP - TSEG_SZ) to ESMMTOP - 1 is partitioned away so that it may only be accessed by the processor interface and only then when the SMM bit (SMMEM#) is set in the request packet. Non-SMM accesses to this memory region are sent to ESI when the TSEG memory block is enabled. Note that once D_LCK is set, these bits become read only. 00: 512kB 01: 1MB 10: 2MB 11: 4MB
0	RWL	0	**T_EN: TSEG Enable** Enabling of SMRAM memory for Extended SMRAM space only. When G_SMRAME =1 and TSEG_EN = 1, the TSEG is enabled to appear in the appropriate physical address space. Note that once D_LCK is set, this bit becomes read only.

4.8.3.10 EXSMRTOP: Extended System Management RAM Top

This register defines the location of the Extended (TSEG) SMM range by defining the top of the TSEG SMM range (ESMMTOP).

Device: 16 Function: 0 Offset: 63h			
Bit	**Attr**	**Default**	**Description**
7:4	RV	0h	Reserved
3:0	RWL	1h	**ESMMTOP: Top of Extended SMM Space (TSEG)** This field contains the address that corresponds to address bits 31 to 28. This field points to the top (+1) of extended SMM space below 4GB. Addresses below 4GB (A[39:32] must be 0) that fall in this range are decoded to be in the extended SMM space and should be routed according to Section 6.1.2.3, "Extended SMRAM Space (TSEG)" on page 335: ESMMTOP-TSEG_SZ <= Address < ESMMTOP TSEG_SZ can be 512KB, 1MB, 2MB, or 4MB, depending on the value of EXSMRC.TSEG_SZ. ESMMTOP is relocatable to accommodate software that wishes to configure the TSEG SMM space before MMIO space is known.See Section 6.2.2.1, "Access to SMM Space (Processor only)" on page 344. This field defaults to point to the same address as default value of TOLM. Note that ESMMTOP cannot be greater than TOLM otherwise the chipset will not function deterministically. Note that once D_LCK is set, this field becomes read only.

4.8.3.11 EXSMRAMC: Expansion System Management RAM Control

Device: 16 Function: 0 Offset: 60h			
Bit	**Attr**	**Default**	**Description**
7	RWC	0	**E_SMERR: Invalid SMRAM Access** This bit is set when CPU has accessed the defined memory ranges in High SMM Memory and Extended SMRAM (T-segment) while not in SMM space and with the D-OPEN bit = 0. Intel® 7300 Chipset will set this bit if any In-Bound access from I/O device targeting SMM range that gets routed to ESI port (master abort). Refer to Table 6-10, "Address Disposition for Inbound Transactions" on page 347 for details. Intel® 7300 Chipset will not set this bit when processor does a cache line eviction (EWB or IWB) to SMM ranges regardless of SMMEM# on FSB. It is software's responsibility to clear this bit. The software must write a 1 to this bit to clear it.
6:0	RV	0h	*Reserved*

Other address mapping registers such as BCTRL (VGAEN), MBASE/LIMIT, PMBASE/LIMIT, etc. are included with the PCI Express registers described in this chapter.

4.8.3.12 HECBASE: PCI Express Extended Configuration Base Address Register

This register defines the base address of the enhanced PCI Express configuration memory.

Device: 16 Function: 0 Offset: 64h			
Bit	**Attr**	**Default**	**Description**
31:24	RV	0h	*Reserved*
23:12	RW	001h	**HECBASE: PCI Express Extended Configuration Base** This register contains the address that corresponds to bits 39 to 28 of the base address for PCI Express extended configuration space. Configuration software will read this register to determine where the 256MB range of addresses resides for this particular host bridge. This register defaults to the same address as the default value for TOLM.
11:0	RV	0h	*Reserved*

4.8.4 AMB Memory Mapped Registers

Intel® 7300 Chipset supports four FBD channels. Intel® 7300 Chipset supports up to 32 FB-DIMM (each with its Advanced Memory Buffer (AMB)) on four channels. Software needs to program AMBPRESENT for each AMB on the platform. There are up to eight function per AMB component with 256B configuration register space per function.

Intel® 7300 Chipset supports a memory mapped register region for software to access individual AMB configuration registers. Memory mapped access to AMB register region is converted by Intel® 7300 Chipset to FBD channel command encoding subject to AMBPRESENT register settings (see Section 4.8.23.11). This region is re-locatable by programming AMBASE register. The size of this region is 128 KB. It is mapped to each AMB addressing slot in 2 KB block. If the corresponding AMBPRESENT bit is not set, then Intel® 7300 Chipset will not send configuration transaction to that AMB addressing slot.

To support SMBus and JTAG access using traditional PCI configuration mechanism, Intel® 7300 Chipset provides a "switching window" using a dedicated PCI device/function and AMBSELECT register. AMBSELECT register can be programmed to select an AMB. Bus 0, device 9, function 0, is mapped to the AMB's configuration registers selected by AMBSELECT register (see "AMBSELECT: AMB Switching Window Select Register" on page 101).

Access to bus 0, device 9, function 0, is limited to SMBus and JTAG only, accesses from other sources such as FSB to this function will be mastered aborted by Intel® 7300 Chipset as non-existent PCI function.

4.8.4.1 AMBASE: AMB Memory Mapped Register Region Base Register

Device: 16
Function: 0
Offset: 48h

Bit	Attr	Default	Description
63:40	RO	0h	**AMBASE_Upper: Upper AMBASE address field**: The upper bits of the 64-bit addressable space are initialized to 0 as default and is unusable in Intel® 7300 Chipset.
39:17	RW	007F00h	**AMBASE:** This marks the 128KB memory-mapped registers region used for accessing AMB registers. It can be placed as MMIO region within the physical limits of the system. Since the Intel® 7300 Chipset uses only 40-bit addressable space, hence only bits 39:17 are valid. The default base address is at FE00_0000h. This field could be re-located by software.
16:0	RV	0h	*Reserved*

4.8.4.2 AMR: AMB Memory Mapped Registers Region Range Register

Device: 16
Function: 0
Offset: 50h

Bit	Attr	Default	Description
31:0	RO	0002_0000h	**AMBASE_Region_Size:** The size of AMB memory mapped register region in bytes. For Intel® 7300 Chipset, the value is 128KB: 2 KB per AMB addressing slot for a total of 16 AMB per channel, 32 KB per FBD channel for a total of four channels.

4.8.4.3 AMBSELECT: AMB Switching Window Select Register

When SMBus/JTAG is accessing bus 0, device 9, function 0, the actual configuration registers being accessed is determined by this register.

Device: 16
Function: 0
Offset: 54h

Bit	Attr	Default	Description
15:9	RV	0h	*Reserved*
8:7	RW	0h	**Channel_Select:** Specify the FBD channel being accessed via bus 0, device 9, function 0 for SM Bus and JTAG only.

Device: 16 Function: 0 Offset: 54h			
Bit	**Attr**	**Default**	**Description**
6:3	RW	0h	**AMB_Select:** Specify the AMB slot being accessed via bus 0, device 9, function 0 for SM Bus and JTAG only.
2:0	RW	0h	**Function_Select:** Specify the function being accessed via bus 0, device 9, function 0 for SM Bus and JTAG only.

4.8.4.4 MAXCH: Maximum Channel Number Register

Device: 16 Function: 0 Offset: 56h			
Bit	**Attr**	**Default**	**Description**
7:0	RO	04h	Maximum_number_channels: Set by hardware to indicate the maximum number of FBD channels that Intel® 7300 Chipset supports. MIRs needs to be programmed appropriately.

4.8.4.5 MAXDIMMPERCH: Maximum DIMM PER Channel Number Register

This register controls the maximum number of AMB DIMM per FBD channel that Intel® 7300 Chipset supports for AMB configuration register access. This register applies only to DIMM modules in the FBD channel, i.e. those AMB with DS[3:0] encoding from 0h to 7h. This register is mainly provided for software.

Device: 16 Function: 0 Offset: 57h			
Bit	**Attr**	**Default**	**Description**
7:0	RO	08h	Maximum_number_DIMM_per_channel: Set by hardware to indicate the maximum number of FBD DIMM AMB per channel that Intel® 7300 Chipset supports.

4.8.4.6 Map to AMB Registers

In Table 4-33, each 2 KB range is mapped to individual AMB configuration registers by address translation of Intel® 7300 Chipset. The re-locatable base address of this memory mapped register region is specified in AMBASE register. Configuration transaction targeting the region is converted to FBD command by Intel® 7300 Chipset and sends to FBD channel subject to AMBPRESENT register settings (see "AMBPRESENT[1:0][1:0]: FBD AMB Slot Present Register" on page 253).

The AMB register's PCI function (3 bits) and offset (8 bits) are used as the offset (11 bits) from the base of each 2KB range for the specific AMB configuration register space.

Table 4-33. Register Offsets in Memory Mapped AMB Registers Region (Sheet 1 of 2)

map to channel_0, AMB_0 registers	7FFh-0h
map to channel_0, AMB_1 registers	FFFh-800h
map to channel_0, AMB_2 registers	17FFh-1000h
map to channel_0, AMB_3 registers	1FFFh-1800h
map to channel_0, AMB_4 registers	27FFh-2000h
map to channel_0, AMB_5 registers	2FFFh-2800h
map to channel_0, AMB_6 registers	37FFh-3000h
map to channel_0, AMB_7 registers	3FFFh-3800h
map to channel_0, AMB_8 registers	47FFh-4000h
map to channel_0, AMB_9 registers	4FFFh-4800h
map to channel_0, AMB_A registers	57FFh-5000h
map to channel_0, AMB_B registers	5FFFh-5800h
map to channel_0, AMB_C registers	67FFh-6000h
map to channel_0, AMB_D registers	6FFFh-6800h
map to channel_0, AMB_E registers	77FFh-7000h
map to channel_0, AMB_F registers	7FFFh-7800h
map to channel_1, AMB_0 registers	87FFh-8000h
map to channel_1, AMB_1 registers	8FFFh-8800h
map to channel_1, AMB_2 registers	97FFh-9000h
map to channel_1, AMB_3 registers	9FFFh-9800h
map to channel_1, AMB_4 registers	A7FFh-A000h
map to channel_1, AMB_5 registers	AFFFh-A800h
map to channel_1, AMB_6 registers	B7FFh-B000h
map to channel_1, AMB_7 registers	BFFFh-B800h
map to channel_1, AMB_8 registers	C7FFh-C000h
map to channel_1, AMB_9 registers	CFFFh-C800h
map to channel_1, AMB_A registers	D7FFh-D000h
map to channel_1, AMB_B registers	DFFFh-D800h
map to channel_1, AMB_C registers	E7FFh-E000h
map to channel_1, AMB_D registers	EFFFh-E800h
map to channel_1, AMB_E registers	F7FFh-F000h
map to channel_1, AMB_F registers	FFFFh-F800h
map to channel_2, AMB_0 registers	107FFh-10000h
map to channel_2, AMB_1 registers	10FFFh-10800h
map to channel_2, AMB_2 registers	117FFh-11000h
map to channel_2, AMB_3 registers	11FFFh-11800h
map to channel_2, AMB_4 registers	127FFh-12000h
map to channel_2, AMB_5 registers	12FFFh-12800h
map to channel_2, AMB_6 registers	137FFh-13000h
map to channel_2, AMB_7 registers	13FFFh-13800h
map to channel_2, AMB_8 registers	147FFh-14000h
map to channel_2, AMB_9 registers	14FFFh-14800h

Table 4-33. Register Offsets in Memory Mapped AMB Registers Region (Sheet 2 of 2)

map to channel_2, AMB_A registers	157FFh-15000h
map to channel_2, AMB_B registers	15FFFh-15800h
map to channel_2, AMB_C registers	167FFh-16000h
map to channel_2, AMB_D registers	16FFFh-16800h
map to channel_2, AMB_E registers	177FFh-17000h
map to channel_2, AMB_F registers	17FFFh-17800h
map to channel_3, AMB_0 registers	187FFh-18000h
map to channel_3, AMB_1 registers	18FFFh-18800h
map to channel_3, AMB_2 registers	197FFh-19000h
map to channel_3, AMB_3 registers	19FFFh-19800h
map to channel_3, AMB_4 registers	1A7FFh-1A000h
map to channel_3, AMB_5 registers	1AFFFh-1A800h
map to channel_3, AMB_6 registers	1B7FFh-1B000h
map to channel_3, AMB_7 registers	1BFFFh-1B800h
map to channel_3, AMB_8 registers	1C7FFh-1C000h
map to channel_3, AMB_9 registers	1CFFFh-1C800h
map to channel_3, AMB_A registers	1D7FFh-1D000h
map to channel_3, AMB_B registers	1DFFFh-1D800h
map to channel_3, AMB_C registers	1E7FFh-1E000h
map to channel_3, AMB_D registers	1EFFFh-1E800h
map to channel_3, AMB_E registers	1F7FFh-1F000h
map to channel_3, AMB_F registers	1FFFFh-1F800h

4.8.5 Interrupt Redirection Registers

4.8.5.1 REDIRCTL: Redirection Control Register

This register controls the priority algorithm of the interrupt redirection mechanism.

Device: 16
Function: 0
Offset: 6Eh

Bit	Attr	Default	Description
15	RW	0	**CHECK_APICID: Check APIC ID** When set, Intel® 7300 Chipset will check if an XTPR update APICID field is non-zero. If the APICID field is all zero's, and this bit is set, the flat vs. cluster mode information is ignored.
14	RW	0	ALLDEST: All destinations are valid Causes the Intel® 7300 Chipset to ignore the XTPR registers for doing interrupt redirection. Any destination indicated in the interrupt request is valid.
13	RW	0	**FFS: Find first algorithm** 0: Use an LRU algorithm to send interrupt to highest XTPR index value when tie. 1: Do not use the LRU algorithm in case of ties, instead, use a find first algorithm. The first index in the valid pool will always be chosen. This emulates the legacy policy.

Device: 16 Function: 0 Offset: 6Eh			
Bit	**Attr**	**Default**	**Description**
12	RW	0	**ARBRST: Arbitration Reset** Neither BIOS nor O.S. should use this bit. 1: This bit resets the arbitration LRU state to a known state for repeatability of tests.
11:8	RW	Ch	**BUCKET2: Scratch bits for compatibility purposes** **These RW bits have no effect on the HW.**
7:4	RW	8h	**BUCKET1: Scratch bits for compatibility purposes** **These RW bits have no effect on the HW.**
3:0	RW	4h	**BUCKET0: Scratch bits for compatibility purposes** **These RW bits have no effect on the HW.**

4.8.5.2 INTRSTS: Interrupt Status Register

This register allows software to read the current state of the global cluster vs. flat bit

Device: 16 Function: 3 Offset: 6Eh			
Bit	**Attr**	**Default**	**Description**
7:2	RO	0	Reserved
1	RWCS	0	MCH received an XTPR message with APIC enable bit set and the Logical APICID field is all 0's. This bit can be cleared by SW.
0	RWCS	0	Interrupt Received to a LAPIC that is disabled: MCH sets this bit when such an interrupt is received (IPI or I/O).

4.8.6 Boot and Reset Registers

4.8.6.1 SYRE: System Reset

This register controls Intel® 7300 Chipset reset behavior. Any resets produced by a write to this register must be delayed until the configuration write is completed on the initiating interface (PCI Express, HI, processor bus, SMBus, JTAG).

There is no "SOFT RESET" bit in this register. That function is invoked through the ESI. There are no CORE:FBD gear ratio definitions in this register. Those are located in the DDRFRQ register.

Device: 16 Function: 0 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
15	RW	0	**SAVCFG: Preserve Configuration** When this bit is set, Intel® 7300 Chipset configuration register contents (except for this bit) are not cleared by hard reset. As this bit is cleared by reset, software must set it after each reset if this behavior is desired for the next reset. If this bit is set, BOFL will not be cleared by reset. Software should use the Boot Flag Reset bit to re-enable the BOFL mechanism.

Device: 16 Function: 0 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
14	RW	0	**CPURST: Processor Reset** If set, the Intel® 7300 Chipset will assert processor RESET# on all four buses as soon as the MCH has been quiesced and there are no transactions, if any, that are pending. Intel® 7300 Chipset will then reset this bit and deassert RESET# following the timing rules described in Section 7.11, "Power Up and System Reset" Intel® 7300 Chipset does not have any mechanism to drain transactions before effecting the CPU RESET#. Note that it is the responsibility of software to ensure that the system is quiesced before sending the configuration write (last command) to set this field in the Intel® 7300 Chipset in order to drive the CPU RESET# signal. Any violation of this usage pattern would render the system unstable and potentially catastrophic.
13	RWST	0	**CPUBIST: Processor Built-In-Self-Test** If set, A[3]# is asserted during Power-On-Configuration (POC), and the processor will run BIST before engaging processor bus protocol.
12	RW	0	**Unused:** This field will not have any associated functionality in the Intel® *7300 Chipset* even if software can read/write to it.
11	RV	0	*Reserved1*
10	ROST	0	**S3: S3 Sleep State** The Intel® 7300 Chipset sets this bit when it sends an Ack-S3 message to the ESI port. The Intel® 7300 Chipset clears this bit after it has placed appropriate FBD channels into the FBD.Calibrate state in response to de-assertion of the RESETI# signal. **Note:** The MCH does not stay in the S3 state. It changes to S3 before transitioning into deeper Sleep states.
9	RW	0	**ROR: Processor Reset on Refresh** If set, the Intel® 7300 Chipset will assert processor RESET# on both busses when a refresh cycle completes.
8	RWST	0	**BNR_INDP_BINIT_MODE: BNR independent of BINIT Mode** 0: The Chipset associates BNR with BINIT and for CPUs that do NOT follow the "BNR independent of BINIT" feature set. 1: Enables the Chipset to use the "BNR independent of BINIT" feature set. i.e no dependency is required between BNR and BINIT. Refer to the BNR#, BINIT# sampling rules in the Intel(R) Pentium(R) 4 and Intel Xeon (TM) Processor External Hardware Specification, Rev 2.5, Ref#14035.
7:0	RV	0h	*Reserved*

4.8.6.2 CPURSTCAPTMR: CPU Reset Done Cap Latency Timer

This register implements the cap latency method for the CPU_RST_DONE/CPU_RST_DONE_ACK using a 12-bit variable timer.

Device: 16 Function: 0 Offset: 42h			
Bit	**Attr**	**Default**	**Description**
15:12	RV	0h	***Reserved***

Device: 16 Function: 0 Offset: 42h			
Bit	**Attr**	**Default**	**Description**
11:0	RWST	7FFh	**DCRT: ESI CPU Reset Done Ack Determinism Timer** This field provides the determinism timer threshold for the Intel® 7300 Chipset for handling the CPU_RESET_DONE/CPU_RESET_DONE_ACK message before deasserting the CPU_RESET#. It uses this 12-bit counter to schedule the CPU_RESET_DONE message on the ESI and then waits for the CPU_RESET_DONE_ACK message to come back and waits for the timer expiry before deasserting CPU_RESET#. Cap_latency = Max(CPU_RST_DONE_ACK_round trip_latency, DCRT). It is expected that the DCRT field is set larger than the expected round trip latency. This provides the necessary leeway for absorbing clock synchronization, jitter, deskew and other variations that will affect the determinism on the ESI port. Hence the data is always sent back only after the expiry of the DCRT field at the heartbeat boundary. It is sticky through reset to permit to allow different types of BIOS flows that may require a hard reset of the Intel® 7300 Chipset. Maximum value is 4095 core clocks A default of 2047 clocks (7FFh) is used.

4.8.6.3 POC: Power-On Configuration

Contrary to its name, this register defines configuration values driven at reset. At power-on, no bits in this register are active as PWRGOOD clears them all. This register only activates configuration on subsequent resets.

Intel® 7300 Chipset drives the contents of this register on FSB[3:0]A_N[25, 15, 12:9] whenever it asserts processor RESET_N. These values are driven during processor RESET_N assertion, and for two host clocks past the trailing edge of processor RESET_N.

This register is sticky through reset; that is, the contents of the register remain unchanged during and following a Hard Reset. This allows system configuration software to modify the default values and reset the system to pass those values to all host bus devices. The default values shown above represent the state of the register following a *power-good* or *power-up* reset.

The POC bits in general do not affect Intel® 7300 Chipset operation except for driving FSB[3:0]A_N[25,15,12:9, 3]. Intel® 7300 Chipset does not support internal POC bit sampling for FSB[3:0]A_N[7].

Read after write to POC register will read updated value but the architectural behavior will not be affected until hard-reset de-assertion. A warm reset (CPU reset) will not cause the contents of the POC register to be altered.

Power-on configuration signals not controlled by the POC register are listed in the notes below.

Note: If the SYRE.CPUBIST register bit is set, then FSB[3:0]A_N[3] will be driven during reset for CPU IBIST.

Note: Intel® 7300 Chipset will drive FSB[3:0]A_N[12:11] differently on each FSB during reset and these are used as the default cluster ID's for the Processors. These default values can be modified by setting POC[8]. Each FSB has a different cluster ID value.

FSB0	FSB1	FSB2	FSB3
00	01	10	11 (Default)

Note: Intel® 7300 Chipset will drive FSB[3:0]BREQ_N[0] during reset.

Device: 16 Function: 0 Offset: 44h			
Bit	**Attr**	**Default**	**Description**
31:22	RV	0	**Reserved**
21	RWST	0	**A[25]** If set, FSB[3:0]A_N[25] is asserted
20:12	RV	0	**Reserved**
11	RWST	0	**A[15]** If set, FSB[3:0]A_N[15] is asserted
8:7	RWST	00	**APIC Cluster ID [1:0]** This field influences the values asserted on FSB[3:0]A_N[12:11] during reset which are used as the default cluster ID's for the CPU's. Each FSB has a different cluster ID value. Only APIC Cluster ID[1] (bit 8) affects the values driven on FSB[3:0]A_N[12:11]. APIC Cluster ID FSB0 FSB1 FSB2 FSB3 `0X 00 01 10 11 `1X 10 11 00 01
6	RWST	0	**DISBINIT: Disable BINIT# Observation** If set, FSB[3:0]A_N[10] will be asserted and All host bus agents will disable BINIT# observation logic. By default, Binit observation is enabled. RAS error logging for BINIT assertion is controlled through POC.DISBINIT as specified in Table 7-37, "Intel® 7300 Chipset Chipset Error List" on page 452.
5	RWST	0	**MCERR: Disable MCERR# Observation** If set, FSB[3:0]A_N[9] will be asserted and all host bus agents will disable MCERR# observation by the processors. By default MCERR observation is enabled. This field is independent of the MCERR# assertion logic described in the RAS chapter.
4:0	RV	0	**Reserved**

4.8.6.4 SPAD[3:0]: Scratch Pad

These scratch pad registers each provide 32 read/writable bits that can be used by software. They are also aliased to fixed memory addresses.

Device: 16 Function: 0 Offset: DCh, D8h, D4h, D0h			
Bit	**Attr**	**Default**	**Description**
31:0	RW	00000000h	Scratch Pad value. These bits have no effect on the hardware.

4.8.6.5 SPADS[3:0]: Sticky Scratch Pad

These sticky scratch pad registers each provide 32 read/writable bits that can be used by software. They are also aliased to fixed memory addresses.

Device: 16 Function: 0 Offset: ECh, E8h, E4h, E0h			
Bit	**Attr**	**Default**	**Description**
31:0	RWST	00000000h	Scratch Pad value. These sticky bits have no effect on the hardware.

4.8.6.6 BOFL[3:0]: Boot Flag

These registers can be used to select the system boot strap processor or for other cross processor communication purposes. When this register is read, the contents of the register is cleared. Therefore, a processor that reads a non-zero value owns the semaphore. Any value can be written to this register at any time.

An example of usage would be for all processors to read the register. The first one gets a non-zero value and owns the semaphore. Since the read clears the value of the register, all other processors will see a zero value and will spin until they receive further notification. After the winning processor is done, it writes a non-zero value of its choice into the register, arming it for subsequent uses. These registers are also aliased to fixed memory addresses.

Device: 16
Function: 0
Offset: CCh, C8h, C4h, C0h

Bit	Attr	Default	Description
31:0	RCW[a]	A5A5A5A5h	**SemaVal: Semaphore Value** Can be written to any value. Value is cleared when there is a read.

Notes:

a. Read/Write but contents of this register will get cleared on read.

4.8.7 Control and Interrupt Registers

4.8.7.1 COHGLOBC: COH Global Control

Coherency Global Control Register. Other clusters may maintain a shadow copy of relevant bits. The shadow bits can be accessible via scan methodologies only.

Device: 16
Function: 0
Offset: F0h

Bit	Attr	Default	Description
31:28	RW	Fh	**FSBEN: FSB3,2,1 and FSB0 Enable** The field is defined as the following: 0000: reserved 0001: FSB3,2,1 is disabled. FSB0 is enabled. 0010: FSB1 is enabled. FSB3,2,0 is disabled. ..... 1111: All FSB's enabled (default) Hard-reset is needed after changing value in this register. ***Note:*** It is recommended that BIOS disable an unpopulated FSB (i.e. when no CPUs are installed on that segment) through the appropriate setting in this register. This can help prevent any side-effects during boot, circuit and other functional/performance issues.
27:0	RW	C087F40h	Reserved

4.8.7.2 COHC: COH Control

Device: 16 Function: 0 Offset: F4h			
Bit	**Attr**	**Default**	**Description**
31:25	RW	0h	**Reserved**
24	RW	0	**ENABLENOSNOOPATTR: No Snoop Attribute for Inb Requests** 0: Disable No Snoop Attribute 1: Enables No Snoop Attribute This field is set to '1' by software when the SF is disabled (See SFCHOP fuse or COHGLOBC.SFbypass register field). When the Snoop filter is enabled, this field is set to 0 (default) and the SF will decide whether to snoop the FSB or not based on its presence vector.
23:16	RW	58h	**Reserved**
15	RW	0	**EARLYSNPDISABLE: Disable Early Snoop Mode** 0: Intel® 7300 Chipset may snoop the remote bus after snoop-filter look up (default) 1: The early snoop mode is disabled to eliminate certain corner cases. The remote snoop will start at two clocks after local snoop phase. When SFBYPASS=1 in COHGLOBC register, EARLYSNPDISABLE should be set to one.
14:0	RW	0106h	**Reserved**

4.8.7.3 COHC2: COH Control 2

Device: 16 Function: 0 Offset: F8h			
Bit	**Attr**	**Default**	**Description**
31:0	RW	23D80958h	**Reserved**

4.8.7.4 COHS: COH Status

Device: 16 Function: 0 Offset: FCh			
Bit	**Attr**	**Default**	**Description**
31:0	RW	0h	**Reserved**

4.8.7.5 FSBC[3:0]: Processor Bus Controller

This register controls the Processor Buses. There is one register for each bus and they must be programmed identically.

Device: 16
Function: 0,3
Offset: 78h, 70h

Bit	Attr	Default	Description
31:15	RW	1FC5h	***Reserved***
14	RW	0	**WDQCEN: Enable BPRI-ADS Quiet cycle Optimization** 0: No BPRI optimization. The FSB will assume that no BREQs are observable and will use the ADS sampled from the CPU for spacing out the Chipset ADS after BPRI is asserted. This may result in a worst case ADS assertion latency of up to 3 clocks from BPRI and will depend on the alignment of the CPU ADS (default) 1: Use 2 clocks for BPRI to chipset ADS (to account for the quiet cycle optimization).
13:4	RW	0h	**Reserved**
3:2	RW	11	**MCHREQ** 00: Intel® 7300 Chipset issues up to 1 requests before de-asserting BPRI to allow the processor to issue requests. 01: Intel® 7300 Chipset issues up to 2 requests (ADS cycles) before de-asserting BPRI to allow the processor to issue requests. 10: Intel® 7300 Chipset issues up to 3 requests (ADS cycles) before de-asserting BPRI to allow the processor to issue requests. 11: Intel® 7300 Chipset issues up to 4 requests (ADS cycles) before de-asserting BPRI to allow the processor to issue requests. **Note**: The chipset will try to optimize successive ADS requests (for snoop, deferred reply completions) as best as possible.
1:0	RW	11	**CPUREQ** 00: Intel® 7300 Chipset allows CPUs no more than 1 requests before asserting BPRI when the MCH has transactions to issue on the FSB. 01: Intel® 7300 Chipset allows CPUs no more than 2 back-to-back requests (ADS cycles) before asserting BPRI when the MCH has transactions to issue on the FSB. 10: Intel® 7300 Chipset allows CPUs no more than 3 back-to-back requests (ADS cycles) before asserting BPRI when the MCH has transactions to issue on the FSB. 11: Intel® 7300 Chipset allows CPUs no more than 4 back-to-back requests (ADS cycles) before asserting BPRI when the MCH has transactions to issue on the FSB. Note: Intel® 7300 Chipset samples BREQ's to deassert BPRI, but once BPRI is deasserted, BREQ's are no longer sampled until BPRI is asserted again

4.8.7.6 DM[1:0]DEF: Data Manager De-feature Register

This register contains the de-feature bits used to de-feature FSB Data Poisoning.

DM0DEF controls the Data Manager corresponding to FSB0 and FSB1. DM1DEF controls the Data Manager corresponding to FSB2 and FSB3.

Device: 17
Function: 3, 0
Offset: 1E0h

Bit	Attr	Default	Description
31:12	RV	0h	**Reserved**
11	RW	0h	Dis_DM_FSBPARPOISON1: 0: Enables Dm fsb poison 1: Disables Dm fsb poison
10	RW	0h	Dis_DM_FSBPARPOISON0: 0: Enables Dm fsb poison 1: Disables Dm fsb poison
9:0	RV	0h	**Reserved**

4.8.7.7 LAPIC_STATE[31:0]: Local APIC State Registers

These registers define the LAPIC state of the processors connected to Intel® 7300 Chipset. Section 4.8.5, "Interrupt Redirection Registers" on page 104 describes the usage of this register. Up to eight logical processors on each bus are supported. The contents of these registers are modified by the xTPR_Update transaction on the processor bus. Index into the LAPIC_STATE registers is defined by Table 4-34.

Table 4-34. LAPIC_STATE Index

Index	Value
4:3	00 for FSB0, 01 for FSB1, 10 for FSB2 and 11 for FSB3
2	Ab[22] XOR Ab[30]
1	Ab[21]
0	Ab[29]

These registers are used for interrupt redirection by the chipset.

Device: 16
Function: 3
Offset: BCh, B8h, B4h, B0h, ACh, A8h, A4h, A0h, 9Ch, 98h, 94h, 90h, 8Ch, 88h, 84h, 80h
Function: 0
Offset: BCh, B8h, B4h, B0h, ACh, A8h, A4h, A0h, 9Ch, 98h, 94h, 90h, 8Ch, 88h, 84h, 80h

Bit	Attr	Default	Description
31	Dev#16/ Fn#0, Offset 80h = RW others = RO	0	**CLUSTER:** Used in interrupt redirection for lowest priority delivery. 0: flat 1: cluster This bit is updated in LAPIC_STATE0 register with the Cluster Mode bit in "any" xTPR_Update transaction provided the APIC enable bit in the xTPR_Update transaction is set. if the APIC enable bit is clear in a xTPR_Update transaction, this bit remains unchanged.
30:24	RV	00h	*Reserved.*

Device: 16 Function: 3 Offset: BCh, B8h, B4h, B0h, ACh, A8h, A4h, A0h, 9Ch, 98h, 94h, 90h, 8Ch, 88h, 84h, 80h Function: 0 Offset: BCh, B8h, B4h, B0h, ACh, A8h, A4h, A0h, 9Ch, 98h, 94h, 90h, 8Ch, 88h, 84h, 80h			
Bit	**Attr**	**Default**	**Description**
23	RW	0	**TPREN: TPR Enable** This bit reflects the value of Ab[31]#. When Ab[31]# is asserted, the value of this bit will be 0.
22:16	RV	0h	*Reserved.*
15:8	RW	0h	**PHYSID: Physical APIC ID** The physical ID of the APIC agent associated with the LAPIC_STATE entry. This field is updated with Aa[19:12] of the xTPR_Update transaction.
7:0	RW	0h	**LOGID: Logical APIC ID** The logical ID of the APIC agent associated with the LAPIC_STATE entry. This field is updated with Aa[11:4] of the xTPR_Update transaction.

4.8.7.8 INTXSTAT0: Interrupt Status Debug Register

This register returns the state of the inputs to the INTx routing logic.

Device: 16 Function: 3 Offset: 50h			
Bit	**Attr**	**Default**	**Description**
31:30	RV	0h	***Reserved***
29	RO	0h	**Internal INTA status**
28	RO	0h	**Intel® QuickData Technology DMA INTA status**
27	RO	0h	**Port 7 INTD status**
26	RO	0h	**Port 7 INTC status**
25	RO	0h	**Port 7 INTB status**
24	RO	0h	**Port 7 INTA status**
23	RO	0h	**Port 6 INTD status**
22	RO	0h	**Port 6 INTC status**
21	RO	0h	**Port 6 INTB status**
20	RO	0h	**Port 6 INTA status**
19	RO	0h	**Port 5 INTD status**
18	RO	0h	**Port 5 INTC status**
17	RO	0h	**Port 5 INTB status**
16	RO	0h	**Port 5 INTA status**
15	RO	0h	**Port 4 INTD status**
14	RO	0h	**Port 4 INTC status**
13	RO	0h	**Port 4 INTB status**
12	RO	0h	**Port 4 INTA status**
11	RO	0h	**Port 3 INTD status**
10	RO	0h	**Port 3 INTC status**
09	RO	0h	**Port 3 INTB status**
08	RO	0h	**Port 3 INTA status**
07	RO	0h	**Port 2 INTD status**

Device: 16 Function: 3 Offset: 50h			
Bit	**Attr**	**Default**	**Description**
06	RO	0h	**Port 2 INTC status**
05	RO	0h	**Port 2 INTB status**
04	RO	0h	**Port 2 INTA status**
03	RO	0h	**Port 1 INTD status**
02	RO	0h	**Port 1 INTC status**
01	RO	0h	**Port 1 INTB status**
00	RO	0h	**Port 1 INTA status**

4.8.7.9 INTXSTAT1: Interrupt Status Debug Register

This register returns the state of the interrupt sources with fixed routing and the interrupt routing destinations.

Device: 16 Function: 3 Offset: 54h			
Bit	**Attr**	**Default**	**Description**
31:18	RV	0h	***Reserved***
17	RO	0h	**Internal INTD**
16	RO	0h	**Internal INTC**
15	RO	0h	**Internal INTB**
14	RO	0h	**IQD DMA INTD**
13	RO	0h	**IQD DMA INTC**
12	RO	0h	**IQD DMA INTB**
11	RV	0h	***Reserved***
10	RO	0h	**INTx[6] pin**
09	RO	0h	**INTx[5] pin**
08	RO	0h	**INTx[4] pin**
07	RO	0h	**INTx[3] pin**
06	RO	0h	**INTx[2] pin**
05	RO	0h	**INTx[1] pin**
04	RO	0h	**INTx[0] pin**
03	RO	0h	**ESI INTD**
02	RO	0h	**ESI INTC**
01	RO	0h	**ESI INTB**
00	RO	0h	**ESI INTA**

4.8.7.10 INTxROUTECTL: Legacy PCI INTx Interrupt Route Control Register

Device: 16
Function: 3
Offset: 58h

Bit	Attr	Default	Description
127:120	RV	0h	**Reserved**
119:116	RW	0001	**Internal Root Port Generated INTA Route:** Controls routing of the coalesced INTA interrupt from all internally generated root port INTA interrupts (for HP/AER/PM). 0000 -> Route to ESI INTA message 0001 -> Route to ESI INTB message 0010 -> Route to ESI INTC message 0011 -> Route to ESI INTD message 0100 -> Route to INT_N[0] pin 0101 -> Route to INT_N[1] pin .. 1010 -> Route to INT_N[6] pin 1011 - 1111 -> Reserved
115:112	RW	0h	**Intel® QuickData Technology DMA INTA Route:**
111:108	RW	0011	**Port7 INTD Route:**
107:104	RW	0010	**Port7 INTC Route:**
103:100	RW	0001	**Port7 INTB Route:**
99:96	RW	0h	**Port7 INTA Route:**
95:92	RW	0h	**Port6 INTD Route:**
91:88	RW	0h	**Port6 INTC Route:**
87:84	RW	0h	**Port6 INTB Route:**
83:80	RW	0h	**Port6 INTA Route:**
79:73	RW	0h	**Port5 INTD Route:**
75:72	RW	0h	**Port5 INTC Route:**
71:68	RW	0h	**Port5 INTB Route:**
67:64	RW	0h	**Port5 INTA Route:**
63:60	RW	0h	**Port4 INTD Route:**
59:56	RW	0h	**Port4 INTC Route:**
55:52	RW	0h	**Port4 INTB Route:**
51:48	RW	0h	**Port4 INTA Route:**
47:44	RW	0h	**Port3 INTD Route:**
43:40	RW	0h	**Port3 INTC Route:**
39:36	RW	0h	**Port3 INTB Route:**
35:32	RW	0h	**Port3 INTA Route:**
31:28	RW	0h	**Port2 INTD Route:**
27:24	RW	0h	**Port2 INTC Route:**
23:20	RW	0h	**Port2 INTB Route:**
19:16	RW	0h	**Port2 INTA Route:**
15:12	RW	0h	**Port1 INTD Route:**
11:8	RW	0h	**Port1 INTC Route:**
7:4	RW	0h	**Port1 INTB Route:**

Device: 16 Function: 3 Offset: 58h			
Bit	**Attr**	**Default**	**Description**
3:0	RW	0h	**Port1 INTA Route:** Controls routing of INTA interrupt received from the PCIE link of Port 1. 0000 -> Route to ESI INTA message 0001 -> Route to ESI INTB message 0010 -> Route to ESI INTC message 0011 -> Route to ESI INTD message 0100 -> Route to INT_N[0] pin 0101 -> Route to INT_N[1] pin .. 1010 -> Route to INT_N[6] pin 1011 - 1111 -> Reserved

4.8.8 Snoop Filter Control and Configuration

4.8.8.1 SFDEF1: SF Control Register

This register is the Snoop Filter control register

Device: 17 Function: 3 Offset: 90h			
Bit	**Attr**	**Default**	**Description**
31:26	RV	0h	Reserved
25:23	RV	111b	Reserved
22:17	RV	010011b	Reserved
16:15	RWST	0h	INTERLEAVEMODE[1:0]: Determines the interleave mode. NOTE: Proper bits need to be set in the CE as well. 00: Reserved 01: Odd Interleave 10: Reserved 11: Reserved
14:11	RW	Ah	Reserved
10	RWST	0	DISWAYHINT: Controls Way-hint mode Active way management uses hints from the processor to select a victim. The SF will decode Ab[37:33] address on the 'b' phase of the address cycle and select the victim based on the Way Hint received from the processor. 0: way-hint is enabled (default). 1: way-hint is disabled.
9:0	RW	1E0h	Reserved

4.8.8.2 SFDEF0: SF Control Register

This register is the Snoop Filter control register

Device: 17 Function: 0 Offset: 90h			
Bit	**Attr**	**Default**	**Description**
31:26	RV	0h	Reserved
25:23	RV	111b	Reserved

Device: 17 Function: 0 Offset: 90h			
Bit	**Attr**	**Default**	**Description**
22:17	RV	010011b	Reserved
16:15	RWST	0h	INTERLEAVEMODE[1:0]: Determines the interleave mode. NOTE: Proper bits need to be set in the CE as well. 00: Reserved 01: Reserved 10: Even Interleave 11: Reserved
14:11	RW	Ah	Reserved
10	RWST	0	DISWAYHINT: Controls Way-hint mode Active way management uses hints from the processor to select a victim. The SF will decode Ab[37:33] address on the 'b' phase of the address cycle and select the victim based on the Way Hint received from the processor. 0: way-hint is enabled (default). 1: way-hint is disabled.
9:0	RW	1E0h	Reserved

4.8.8.3 SFINIT[1:0]: SF Initialization Register

Device: 17 Function: 0, 3 Offset: 9Ah			
Bit	**Attr**	**Default**	**Description**
15:2	RV	0h	Reserved
1	RWST	1	FASTCLRONRST: Fast clear on next reset 1'b0: disable clearing arrays on the next reset. 1'b1: clears array on the next reset.
0	RW	0	Reserved

4.8.9 PCI Express Device Configuration Registers

This section describes the registers associated with the PCI Express Interface, with the exception of the registers described in Section 4.8.13.18.

The PCI Express register structure is exposed to the operating system and requires a separate device per port. Ports 1-7 are assigned devices 1 through 7 while Port 0 is the ESI interconnect to South Bridge. The PCI Express ports 1-7 determine the maximum link width of the port through the DEVPRES register and the PXPLWTCTRL register. All "present" device ports as determined by the DEVPRES register are visible to OS even if unconnected. Configuration accesses to unconnected ports is allowed to permit device remapping, Hot-plug etc.

If the IOU port groups do not bifuricate(e.g. x4 from a x8 link), then the unused ports will not be present or visible to the OS. The DEVPRES register indicates which devices are visible to software.

Table 4-35. When will a PCI Express Device be Accessible?

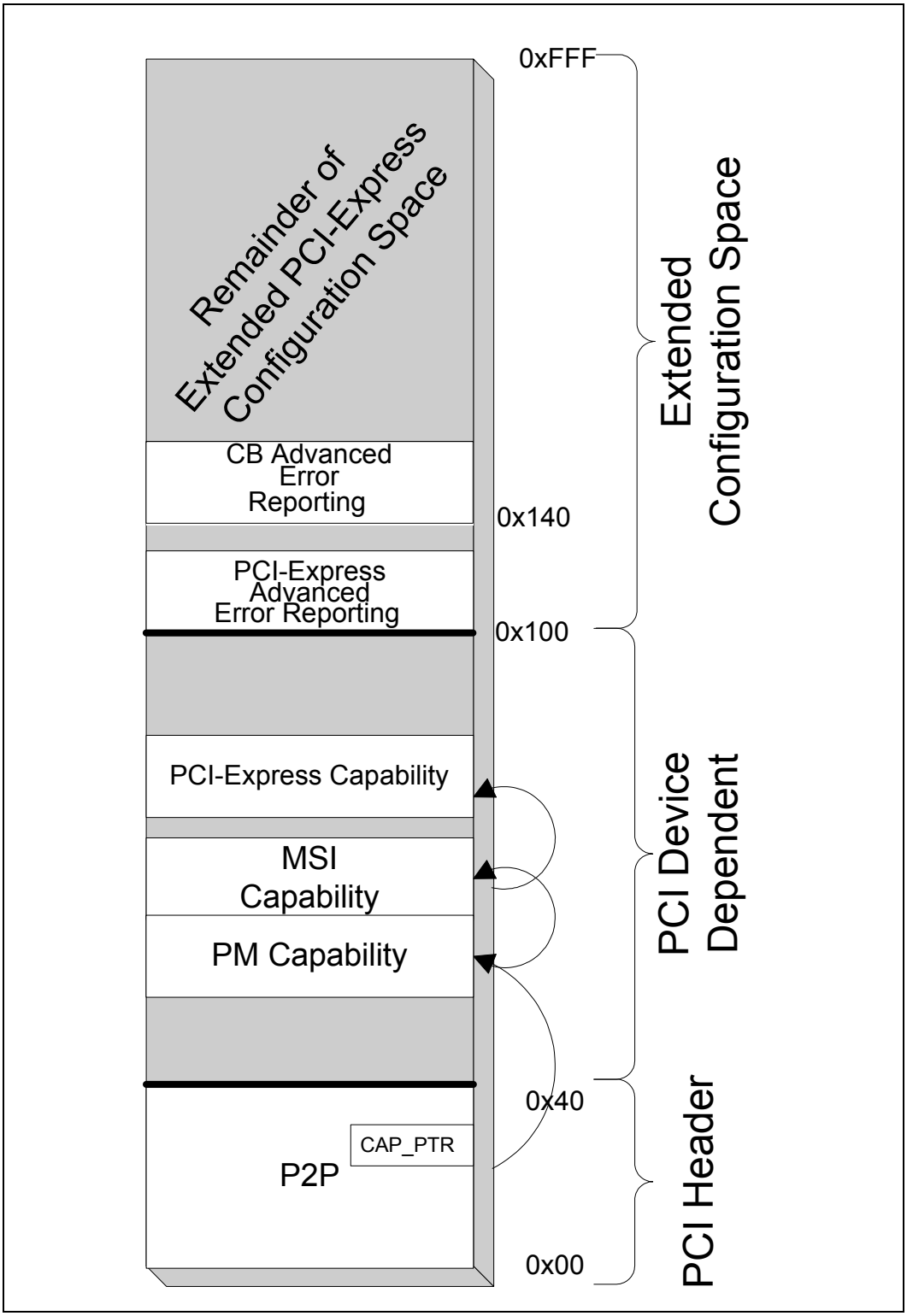
PCI Express Port	Device	x8	Registers may be accessed if:
7	7	possible combination	Port 7 is connected to a x4 device
6	6		Port6 is connected to a x4 or x8 device
5	5	possible combination	Port5 is connected to a 4x device
4	4		Port 4 is connected to connected to a x4 or x8
3	3	Possible combination	Port3 is connected to a x4 device
2	2		Port2 is connected to a x4 or x8 device
1	1	x4 port only	Port1 is connected to a x4 device
0	0	x4 port only	Port0 is connected to south bridge through ESI and cannot be combined with any other port

Figure 4-4 illustrates how each PCI Express port's configuration space appears to software. Each PCI Express port's configuration space has four regions:

- **Standard PCI Header** - This region closely resembles a standard PCI-to-PCI bridge header.
- **PCI Device Dependent Region** - The region is also part of standard PCI configuration space and contains the PCI capability structures. For Intel® 7300 Chipset, the supported capabilities are:
  - Message Signalled Interrupts
  - Hot-Plug
  - 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 Intel® 7300 Chipset supports the Enhanced Error Signalling capability.
- **Capability Working Register Sets** - These ranges are indirectly accessed through Data and Select registers in the capability structures. For the Intel® 7300 Chipset, working register sets exist for the Standard hot-plug Controller and Power Management capabilities.

Figure 4-4 shows the configuration register offset addresses for each of the PCI Express ports as defined in the *PCI Express Base Specification*, Revision 1.0a. It is also compatible with the standard PCI 2.3 capability structure and comprises of a linked list where each capability has a pointer to the next capability in the list. For PCI Express extended capabilities, the first structure is required to start at 0x100 offset.

Figure 4-4. PCI Express Configuration Space



4.8.10 PCI Express Header

The following registers define the standard PCI 2.3 compatible and extended PCI Express configuration space for each of the PCI Express x4 links in the Intel® 7300 Chipset. Unless otherwise specified, the registers are enumerated as a vector [7:1] mapping to each of the seven PCI Express ports uniquely while the ESI port is referred by index 0.

4.8.10.1 PEXCMD[7:0]: Command Register

This register defines the PCI 2.3 compatible command register values applicable to PCI Express space.

Device: 7-0
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 the Intel® 7300 Chipset to route interrupt messages received at the PCI Express port. However, this bit controls the generation of legacy interrupts to the ESI for PCI Express errors detected internally in this port (e.g. Malformed TLP, CRC error, completion time out etc.) or when receiving root port error messages or interrupts due to HP/PM events generated in legacy mode within the Intel® 7300 Chipset. Refer to the INTP register in Section 4.8.10.27, "INTP[7:0]: Interrupt Pin Register" on page 131 for interrupt routing to ESI. 1: Legacy Interrupt mode is disabled 0: Legacy Interrupt mode is enabled
9	RO	0	**FB2B: Fast Back-to-Back Enable** Not applicable to PCI Express and is hardwired to 0
8	RW	0	SERRE: SERR Message Enable This field handles the reporting of fatal and non-fatal errors by enabling the error pins ERR[2:0]. 1: Intel® 7300 Chipset is enabled to send fatal/non-fatal errors. 0: Intel® 7300 Chipset is disabled from generating fatal/non-fatal errors. The errors are also enabled by the PEXDEVCTRL register in Section 4.8.13.4, "PEXDEVCTRL[7:0]: PCI Express Device Control Register" on page 150 In addition, for Type 1 configuration space header devices, e.g. Virtual P2P bridge), this bit, when set, enables transmission of ERR_NONFATAL and ERR_FATAL error messages[a] forwarded from the PCI Express interface. This bit does not affect the transmission of forwarded ERR_COR messages. Refer to the Intel® 7300 Chipset RAS Error Model.
7	RO	0	IDSELWCC: IDSEL Stepping/Wait Cycle Control Not applicable to PCI Express. Hardwired to 0.
6	RW	0	PERRE: Parity Error Response Enable When set, this field enables parity checking.
5	RO	0	**VGAPSE: VGA palette snoop Enable** Not applicable to PCI Express. Hardwired to 0.
4	RO	0	**MWIEN: Memory Write and Invalidate Enable** Not applicable to PCI Express. Hardwired to 0.
3	RO	0	SCE: Special Cycle Enable Not applicable to PCI Express. Hardwired to 0.

Device: 7-0 Function: 0 Offset: 04h			
Bit	**Attr**	**Default**	**Description**
2	RW	0	**BME: Bus Master Enable** Controls the ability of the PCI Express port to forward memory or I/O transactions. 1: Enables the PCI Express port to successfully complete the memory or I/O read/write requests. 0: The Bus Master is disabled. Intel® 7300 Chipset will treat upstream memory writes/reads, I/O writes/reads, and MSIs as illegal cycles and return Unsupported Request Status (equivalent to Master abort) in PCI Express Requests other than inbound memory or I/O (e.g configuration, outbound) are not controlled by this bit. The BME is typically used by the system software for operations such as Hot-plug, device configuration ***Note:*** When the CPURESET# signal is asserted during a power good or hard reset and after the ESI completes its training, the LPC device in the Intel® 631xESB/632xESB I/O Controller Hub (or other NIC/SIO4 cards could potentially send inbound requests even before the CPURESET# is deasserted.
1	(port 7-1) = RW (port 0) = RO	0	**MSE: Memory Space Enable** Controls the bridge's response as a target to memory accesses on the primary interface that address a device that resides behind the bridge in both the nonprefetchable and prefetchable memory ranges (high/low) or targets a memory-mapped location within the bridge itself 1: Enables the Memory and Pre-fetchable memory address ranges (MMIO) defined in the **MBASE/MLIM, PMBASE/PMLIM, PMBU/PMLU** registers. 0: Disables the entire memory space seen by the PCI Express port on the primary side (Intel® 7300 Chipset). Requests will then be subtractively claimed by Intel® 631xESB/632xESB I/O Controller Hub. For port 0, this bit is hardwired to 0 since the ESI is not a P2P bridge.
0	(port 7-1) = RW (port 0) = RO	0	**IOAE: Access Enable** 1: Enables the I/O address range defined in the IOBASE and IOLIM registers. 0: Disables the entire I/O space seen by the PCI Express port on the primary. Requests will be then be subtractively claimed by Intel® 631xESB/632xESB I/O Controller Hub For port 0, this bit is hardwired to 0 since the ESI is not a P2P bridge.

Notes:

a. In addition, BCCTRL.BCSERRE also gates the transmission of ERR_FATAL, NON_FATAL and ERR_COR messages received from the PCI Express interface. See Section 4.8.10.28, "BCTRL[7:1]: Bridge Control Register" on page 132

4.8.10.2 PEXSTS[7:0]: Status Register

The PEXSTS is a 16-bit status register that reports the occurrence of error conditions associated with the primary side of the "virtual" PCI-PCI bridge embedded in the selected PCI Express cluster of Intel® 7300 Chipset.

Device: 7-0 Function: 0 Offset: 06h			
Bit	**Attr**	**Default**	**Description**
15	RWC	0	**DPE: Detected Parity Error** This bit is set when the PCI Express port receives an uncorrectable data error or Address/Control parity errors regardless of the Parity Error Response Enable bit (PERRE). This applies only to parity errors that target the PCI Express port interface i.e. outbound writes/inbound read completions corresponding to outbound P/inbound NP requests. The detected parity error maps to B1, F6, M2 and M4 (uncorrectable data error from FSB, Memory or internal sources) of the Intel® 7300 Chipset.

Device: 7-0 Function: 0 Offset: 06h			
Bit	**Attr**	**Default**	**Description**
14	RWC	0	**SSE: Signaled System Error** 1: The PCI Express port generated internal FATAL/NON FATAL errors (IO0-IO17) through the ERR[2:0] pins with SERRE bit enabled. Software clears this bit by writing a '1' to it. 0: No internal PCI Express port errors are signaled.
13	RWC	0	**RMA: Received Master Abort** This bit is set when a requestor (primary side for Type 1 header configuration space header device) receives a completion with Unsupported Request Completion Status. 1: Assert this RMA bit when the primary side performs operations for an unsupported transaction. These apply to inbound configs, I/O accesses, locks, bogus memory reads and any other request that is master aborted internally. These are terminated on the PCI Express link with a UR completion status, but only if a completion is required. Software clears this bit by writing a 1 to it. PEXDEVSTS.URD is set and UNCERRSTS[20].IO2Err is set in addition. 0: No Master Abort is generated See Section 7.5.14.1, "Unsupported Transactions and Unexpected Completions" on page 412 for more details.
12	RWC	0	**RTA: Received Target Abort** This bit is set when a requestor (primary side for Type 1 header configuration space header device) receives a completion with Completer Abort Completion Status. e.g. For supported requests that cannot be completed because of address decoding problems or other errors. These are terminated on the PCI Express link with a CA completion status, but only if a completion is required. Software clears this bit by writing a 1 to it. This field also influences the SECSTS.SSTA field on the secondary side completion abort. **Note:** Currently the Intel® 7300 Chipset only signals RTA for Peer to Peer transactions in the primary side and not for events that happen internally in the Intel® 7300 Chipset (such as in the MC).
11	RO	0	**STA: Signaled Target Abort** Target Abort does not exist on the primary side of the PCI Express port. Hardwired to 0.
10:9	RO	0h	**DEVSELT: DEVSEL# Timing** Not applicable to PCI Express. Hardwired to 0.
8	RWC	0	**MDPERR: Master Data Parity Error** This bit is set by the PCI Express port if the Parity Error Response Enable bit (PERRE) is set and it receives error B1, F2, F6, M2 and M4 (uncorrectable data error or Address/Control parity errors or an internal failure). If the Parity Error Enable bit (PERRE) is cleared, this bit is never set. Refer to Table 4-36, "PEXSTS and SECSTS Master/Data Parity Error RAS Handling" on page 127
7	RO	0	**FB2B: Fast Back-to-Back** Not applicable to PCI Express. Hardwired to 0.
6	RV	0	*Reserved.* (by PCI SIG)
5	RO	0	**66MHZCAP: 66 MHz capable.** Not applicable to PCI Express. Hardwired to 0.
4	RO	1	**CAPL: Capabilities List** This bit indicates the presence of PCI Express capabilities list structure in the PCI Express port. Hardwired to 1. (Mandatory)
3	RO	0	**INTxSTAT: INTx Status** Indicates that an INTx interrupt message is pending internally in the PCI Express port. This bit does not get set for interrupts forwarded to the root port from downstream devices in the hierarchy. The intx status bit should be deasserted when all the relevant events viz RAS errors/HP/PM internal to the port that requires legacy interrupts are cleared by software.
2:0	RV	0h	*Reserved.* (by PCI SIG)

4.8.10.3 CLS[7:0]: Cache Line Size

This register contains the Cache Line Size and is set by BIOS/Operating system. It does not affect the PCI Express port functionality in Intel® 7300 Chipset.

Device: 7-0 Function: 0 Offset: 0Ch			
Bit	**Attr**	**Default**	**304Description**
7:0	RW	00h	CLS: Cache Line Size This is an 8-bit value that indicates the size of the cache line and is specified in DWORDs. It does not affect the Intel® 7300 Chipset.

4.8.10.4 PRI_LT[7:0]: Primary Latency Timer

This register denotes the maximum timeslice for a burst transaction in legacy PCI 2.3 on the primary interface. It does not affect/influence PCI Express functionality.

Device: 7-0 Function: 0 Offset: 0Dh			
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	Prim_Lat_timer: Primary Latency Timer Not applicable to PCI Express. Hardwired to 00h.

4.8.10.5 BIST[7:0]: 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® 7300 Chipset.

Device: 7-0 Function: 0 Offset: 0Fh			
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	BIST_TST: BIST Tests Not supported. Hardwired to 00h

4.8.10.6 BAR0[7:0]: Base Address Register 0

Base address registers are used for mapping internal registers to an MMIO or I/O space. It does not affect the Intel® 7300 Chipset. The base address register 0 is not supported/defined in the PCI Express port of the Intel® 7300 Chipset.

4.8.10.7 BAR1[7:0]: Base Address Register 1

The base address register 1 is not supported/defined in the Intel® 7300 Chipset.

4.8.10.8 EXP_ROM[0]: Expansion ROM Registers

The ESI port (device 0, function 0) does not implement any Base address registers in the Intel® 7300 Chipset from offset 10h to 24h. Similarly no Expansion ROM base address register is defined in offset 30h. Also no Cardbus CIS pointer is defined in offset 28h. The MIN_GNT (offset 3Eh) and MAX_LAT (3Fh) registers are also not implemented as they are not applicable to the ESI interface.

4.8.10.9 PBUSN[7:1]: Primary Bus Number

This register identifies the bus number on the on the primary side (Intel® 7300 Chipset) of the PCI Express port.

Device: 7-1
Function: 0
Offset: 18h

Bit	Attr	Default	Description
7:0	RO	00h	PBUBSNUM: Primary Bus Number Configuration software typically programs this field with the number of the bus on the primary side of the bridge. Since the PCI Express virtual PCI-PCI bridge is an internal device and its primary bus is always 0, these bits are read only and are hardwired to 0.

4.8.10.10 SBUSN[7:1]: Secondary Bus Number

This register identifies the bus number assigned to the secondary side (PCI Express) of the "virtual" PCI-PCI bridge. This number is programmed by the PCI configuration software to allow mapping of configuration cycles to devices connected to PCI Express.

Device: 7-1
Function: 0
Offset: 19h

Bit	Attr	Default	Description
7:0	RW	00h	SECBUSNUM: Secondary Bus Number This field is programmed by configuration software with the lowest bus number of the buses connected to the PCI Express port.

4.8.10.11 SUBUSN[7:1]: Subordinate Bus Number

This register identifies the subordinate bus (if any) that resides at the level below the secondary PCI Express interface. This number is programmed by the PCI configuration software to allow mapping of configuration cycles to devices subordinate to the secondary PCI Express port.

Device: 7-1
Function: 0
Offset: 1Ah

Bit	Attr	Default	Description
7:0	RW	00h	SUBBUSNUM: Subordinate Bus Number This register is programmed by configuration software with the number of the highest subordinate bus that is behind the PCI Express port.

4.8.10.12 SEC_LT[7:1]: Secondary Latency Timer

This register denotes the maximum timeslice for a burst transaction in legacy PCI 2.3 on the secondary interface. It does not affect/influence PCI Express functionality.

Device: 7-1
Function: 0
Offset: 1Bh

Bit	Attr	Default	Description
7:0	RO	00h	Slat_tmr: Secondary Latency Timer Not applicable to PCI Express. Hardwired to 00h.

4.8.10.13 IOBASE[7:1]: I/O Base Register

The I/O Base and I/O Limit registers (see Section 4.8.10.14, "IOLIM[7:1]: I/O Limit Register" on page 125) define an address range that is used by the PCI Express port to determine when to forward I/O transactions from one interface to the other using the following formula:

IO_BASE <= A[15:12]<=IO_LIMIT

Only the upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] are treated as 0. The bottom of the defined I/O address range will be aligned to a 4KB boundary while the top of the region specified by IO_LIMIT will be one less than a 4 KB multiple.

Device: 7-1
Function: 0
Offset: 1Ch

Bit	Attr	Default	Description
7:4	RW	0h	IOBASE: I/O Base Address Corresponds to A[15:12] of the I/O addresses at the PCI Express port.
3:0	RO	0h	IOCAP: I/O Address capability 0h – 16 bit I/O addressing, (supported) 1h – 32 bit I/O addressing, others - Reserved. The Intel® 7300 Chipset does not support 32 bit addressing, so these bits are hardwired to 0.

4.8.10.14 IOLIM[7:1]: I/O Limit Register

The I/O Base and I/O Limit registers define an address range that is used by the PCI Express bridge to determine when to forward I/O transactions from one interface to the other using the following formula:

IO_BASE <= A[15:12] <=IO_LIMIT

Only the upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] of the I/O limit register is treated as FFFh.

Device: 7-1
Function: 0
Offset: 1Dh

Bit	Attr	Default	Description
7:4	RW	0h	IOLIMIT: I/O Address Limit Corresponds to A[15:12] of the I/O addresses at the PCI Express port.
3:0	RO	0h	IOLCAP: I/O Address Limit Capability 0h – 16 bit I/O addressing, (supported) 1h – 32 bit I/O addressing, others - Reserved. The Intel® 7300 Chipset does not support 32 bit I/O addressing, so these bits are hardwired to 0.

4.8.10.15 SECSTS[7:1]: Secondary Status

SECSTS is a 16-bit status register that reports the occurrence of error conditions associated with secondary side (i.e. PCI Express side) of the "virtual" PCI-PCI bridge embedded within Intel® 7300 Chipset.

Device: 7-1 Function: 0 Offset: 1Eh			
Bit	**Attr**	**Default**	**Description**
15	RWC	0	SDPE: Detected Parity Error This bit is set by the Intel® 7300 Chipset whenever it receives a poisoned TLP in the PCI Express port regardless of the state the Parity Error Response bit (in the BCTRL.PRSPEN register). This corresponds to IO4. Refer to Table 4-36, "PEXSTS and SECSTS Master/Data Parity Error RAS Handling" on page 127
14	RWC	0	**SRSE: Received System Error** This bit is set by the Intel® 7300 Chipset when it receives a ERR_FATAL or ERR_NONFATAL message. Note that BCTRL.BCSERRE is not a gating item for the recording of this error on the secondary side). This corresponds but not limited IO11 and IO1. Refer to Table 4-36, "PEXSTS and SECSTS Master/Data Parity Error RAS Handling" on page 127
13	RWC	0	SRMAS: Received Master Abort Status This bit is set when the PCI Express port receives a Completion with "Unsupported Request Completion" Status.
12	RWC	0	SRTAS: Received Target Abort Status This bit is set when the PCI Express port receives a Completion with "Completer Abort" Status. Note: Intel® 7300 Chipset will complete all outbound NP requests on the FSB that received the Completer Abort status on the PCI Express similar to Master abort case. i.e drive all 1s for reads with normal data response and "no data" for writes.
11	RWC	0	SSTAS: Signaled Target Abort This bit is set when the PCI Express port completes a request with "Completer Abort" Status when the PEXSTS.RTA is set since the Intel® 7300 Chipset acts as a virtual PCI bridge and passes the completion abort from the primary to the secondary side.
10:9	RO	00	SDEVT: DEVSEL# Timing Not applicable to PCI Express. Hardwired to 0
8	RWC	0	SMDPERR: Master Data Parity Error This bit is set by the PCI Express port on the secondary side (PCI Express link) if the Parity Error Response Enable bit (PRSPEN) in the Section 4.8.10.28, "BCTRL[7:1]: Bridge Control Register" on page 132 is set and either of the following two conditions occurs: • The PCI Express port receives a Completion marked poisoned • The PCI Express port poisons a write Request. If the Parity Error Response Enable bit is cleared, this bit is never set. Refer to Table 4-36, "PEXSTS and SECSTS Master/Data Parity Error RAS Handling" on page 127 for details on the data parity error handling matrix in the Intel® 7300 Chipset. **Note**: Intel® 7300 Chipset does not poison the write request within the secondary side but only forwards a poisoned TLP received from the primary side to the external PCI Express card.
7	RO	0	SFB2BTC: Fast Back-to-Back Transactions Capable Not applicable to PCI Express. Hardwired to 0.
6	RV	0	*Reserved.* (by PCI SIG)
5	RO	0	S66MHCAP: 66 MHz capability Not applicable to PCI Express. Hardwired to 0.
4:0	RV	0h	*Reserved.* (by PCI SIG)

Table 4-36. PEXSTS and SECSTS Master/Data Parity Error RAS Handling

Register Name	OB Post (e.g. outbound Memory writes)	OB Compl (e.g. Inbound Memory NP requests)	IN Post (e.g. inbound memory writes)	IB Cmpl (e.g. outbound Memory/I/O NP requests)
PEXSTS[15].DPE	yes[a]	yes[b]	no[c]	no[c]
PEXSTS[8].MDPERR (gated by PEXCMD.PERRE)	no[d]	yes[e]	yes[f]	yes[g]
SECSTS[15].SDPE	no[h]	no[h]	yes	yes
SECSTS[8].SMDPERR (gated by BCTRL.PRSPEN)	yes[i]	yes[j]	no[k]	yes[l]

Notes:

a. Due to Poisoned TLP (section 6.7.2.1 (Error forwarding) of *PCI-Express Base Specification*, Revision 1.0 from primary to secondary (Outbound write request)
b. As a result of receiving poisoned TLP on the internal (primary) bus for an inbound memory read request (See DPE field (bit 15) definition in Section 4.8.10.2, "PEXSTS[7:0]: Status Register" on page 121 or as a consequence of section 6.7.2.1 (Error forwarding rule) of *PCI-Express Base Specification*, Revision 1.0 for poisoned completion going from primary to secondary that sets the PEXSTS.DPE.
c. This field is no because the TLPs are forwarded from secondary to primary and not received as per definition of PEXSTS.DPE field.
d. This field is a "no" because the OB posted data is not a completion on the primary side (PEXSTS.MDPERR definition) and the MCH is such there is no data poisoning within the primary I/O cluster. It only receives poisoned TLP from the internal core such as DM/Memory/FSB.
e. Due to definition of MDPERR field (bit 8) of Section 4.8.10.2, "PEXSTS[7:0]: Status Register" on page 121 since it gets inbound read request completions and acts as master.
f. Due to Poisoned TLP (section 6.7.2.1 (Error forwarding) of *PCI-Express Base Specification*, Revision 1.0 from secondary to primary (inbound write request)
g. Due to Poisoned TLP (section 6.7.2.1 (Error forwarding) of *PCI-Express Base Specification*, Revision 1.0 from secondary to primary (e.g outbound read request that had a poisoned TLP returned inbound and was forwarded to the transmitting side.
h. Since it only forwards outbound TLPs/completions on the secondary side from the primary
i. As a consequence of section 6.7.2.1 (Error forwarding rule) of *PCI-Express Base Specification*, Revision 1.0a for poisoned TLP due to an outbound write going from primary to secondary that sets the SECSTS.MDPERR
j. Due to Poisoned TLP (section 6.7.2.1 (Error forwarding) of *PCI-Express Base Specification*, Revision 1.0 from primary to secondary (e.g inbound read request that had poisoned TLP returned and was forwarded to the transmitting side
k. The secondary side does not poison the data (inbound write) when it forwards to the primary and hence it is a "no"
l. Base definition of SECSTS.SMDPERR since a completion TLP that is poisoned is received corresponding to an outbound read request.

4.8.10.16 MBASE[7:1]: Memory Base

The Memory Base and Memory Limit registers define a memory mapped I/O non-prefetchable address range (32-bit addresses) and Intel® 7300 Chipset directs accesses in this range to the PCI Express port based on the following formula:

MEMORY_BASE <= A[31:20] <= MEMORY_LIMIT

The upper 12 bits of both the Memory Base and Memory Limit registers are read/write and corresponds to the upper 12 address bits, AD[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, AD[19:0], of the memory base address are zero. Similarly, the bridge assumes that the lower 20 address bits, AD[19:0], of the memory limit address (not implemented in the Memory Limit register) are FFFFFh. Thus, the bottom of the defined memory address range will be aligned to a 1 MB boundary and the top of the defined memory address range will be one less than a 1 MB boundary. Refer to Section 6, "System Address Map" for further details on address mapping.

Device: 7-1 Function: 0 Offset: 20h			
Bit	**Attr**	**Default**	**Description**
15:4	RW	0h	MBASE: Memory Base Address Corresponds to A[31:20] of the memory address on the PCI Express port.
3:0	RV	0h	*Reserved.* (by PCI SIG)

4.8.10.17 MLIM[7:1]: Memory Limit

This register controls the processor to PCI Express non-prefetchable memory access routing based on the following formula as described above:

MEMORY_BASE <= A[31:20] <= MEMORY_LIMIT

The upper 12 bits of the register are read/write and correspond to the upper 12 address bits A[31:20] of the 32 bit address. The bottom 4 bits of this register are read-only and return zeroes when read. This register must be initialized by the configuration software. For the purpose of address decode address bits A[19:0] are assumed to be FFFFFh.

Memory range covered by MBASE and MLIM registers, are used to map non-prefetchable PCI Express address ranges (typically where control/status memory-mapped I/O data structures reside) and PMBASE and PMLIM are used to map prefetchable address ranges. This segregation allows application of USWC space attribute to be performed in a true plug-and-play manner to the prefetchable address range for improved PCI Express memory access performance.

Note also that configuration software is responsible for programming all address range registers such as MIR, MLIM, MBASE, IOLIM, IOBASE, PMBASE, PMLIM, PMBU, PMLU (coherent, MMIO, prefetchable, non-prefetchable, I/O) with the values that provide exclusive address ranges i.e. prevent overlap with each other and/or with the ranges covered with the main memory. There is no provision in the Intel® 7300 Chipset hardware to enforce prevention of overlap, and proper operation of the system in the case of overlap is not guaranteed.

Device: 7-1 Function: 0 Offset: 22h			
Bit	**Attr**	**Default**	**Description**
15:4	RW	0h	MLIMIT: Memory Limit Address Corresponds to A[31:20] of the memory address that corresponds to the upper limit of the range of memory accesses that will be passed by the PCI Express bridge
3:0	RV	0h	*Reserved.* (by PCI SIG)

4.8.10.18 PMBASE[7:1]: Prefetchable Memory Base

The Prefetchable Memory Base and Memory Limit registers define a memory mapped I/O prefetchable address range (32-bit addresses) which is used by the PCI Express bridge to determine when to forward memory transactions based on the following formula:

PREFETCH_MEMORY_BASE <= A[31:20] <= PREFETCH_MEMORY_LIMIT

The upper 12 bits of both the Prefetchable Memory Base and Memory Limit registers are read/write and corresponds to the upper 12 address bits, A[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, A[19:0], of the memory base address are zero. Similarly, the bridge assumes that the lower 20 address bits, A[19:0], of the memory limit address (not implemented in the Memory Limit register) are FFFFFh. Thus, the bottom of the defined memory address range will be aligned to a 1 MB boundary and the top of the defined memory address range will be one less than a 1 MB boundary.

Device: 7-1
Function: 0
Offset: 24h

Bit	Attr	Default	Description
15:4	RW	0h	PMBASE: Prefetchable Memory Base Address Corresponds to A[31:20] of the prefetchable memory address on the PCI Express port.
3:0	RO	1h	PMBASE_CAP: Prefetchable Memory Base Address Capability 0h – 32 bit Prefetchable Memory addressing 1h – 64bit Prefetchable Memory addressing, others - Reserved.

The bottom 4 bits of both the Prefetchable Memory Base and Prefetchable Memory Limit registers are read-only, contain the same value, and encode whether or not the bridge supports 64-bit addresses. If these four bits have the value 0h, then the bridge supports only 32 bit addresses. If these four bits have the value 01h, then the bridge supports 64-bit addresses and the Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers hold the rest of the 64-bit prefetchable base and limit addresses respectively.

4.8.10.19 PMLIM[7:1]: Prefetchable Memory Limit

This register controls the processor to PCI Express prefetchable memory access routing based on the following formula as described above:

PREFETCH_MEMORY_BASE <= A[31:20] <= PREFTCH_MEMORY_LIMIT

The upper 12 bits of the register are read/write and correspond to the upper 12 address bits A[31:20] of the 32 bit address. This register must be initialized by the configuration software. For the purpose of address decode address bits A[19:0] are assumed to be FFFFFh.

Device: 7-1
Function: 0
Offset: 26h

Bit	Attr	Default	Description
15:4	RW	0h	PMLIMIT: Prefetchable Memory Limit Address Corresponds to A[31:20] of the memory address on the PCI Express bridge
3:0	RO	1h	PMLIMIT_CAP: Prefetchable Memory Limit Address Capability 0h – 32 bit Prefetchable Memory addressing 1h – 64 bit Prefetchable Memory addressing, others - Reserved.

4.8.10.20 PMBU[7:1]: Prefetchable Memory Base (Upper 32 bits)

The Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers are extensions to the Prefetchable Memory Base and Prefetchable Memory Limit registers. If the Prefetchable Memory Base and Prefetchable Memory Limit registers indicate support for 32-bit addressing, then the Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers should return zero when read. If the Prefetchable Memory Base and Prefetchable Memory Limit registers indicate support for 64-bit addressing, then the Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers are implemented as read/write registers.

If a 64-bit prefetchable memory address range is supported, the Prefetchable Base Upper 32 Bits and Prefetchable Limit Upper 32 Bits registers specify the upper 32-bits, corresponding to A[63:32], of the 64-bit base and limit addresses which specify the prefetchable memory address range.

Device: 7-1
Function: 0
Offset: 28h

Bit	Attr	Default	Description
31:0	RW	0h	PUMBASE: Prefetchable Upper 32-bit Memory Base Address Corresponds to A[63:32] of the memory address that maps to the upper base of the prefetchable range of memory accesses that will be passed by the PCI Express bridge. OS should program these bits based on the available physical limits of the system.

4.8.10.21 PMLU[7:1]: Prefetchable Memory Limit (Upper 32 bits)

Device: 7-1
Function: 0
Offset: 2Ch

Bit	Attr	Default	Description
31:0	RW	0h	PUMLIM: Prefetchable Upper 32-bit Memory Limit Address Corresponds to A[63:32] of the memory address that maps to the upper limit of the prefetchable range of memory accesses that will be passed by the PCI Express bridge. OS should program these bits based on the available physical limits of the system.

4.8.10.22 IOB[7:1]: I/O Base Register (Upper 16 bits)

Not used since Intel® 7300 Chipset does not support upper 16-bit I/O addressing.

4.8.10.23 IOL[7:1]: I/O Limit Register (Upper 16 bits)

Not used since Intel® 7300 Chipset does not support upper 16-bit I/O addressing.

4.8.10.24 CAPPTR[7:0]: Capability Pointer

The CAPPTR is used to point to a linked list of additional capabilities implemented by this device.

It provides the offset to the first set of capabilities registers located in the PCI compatible space from 40h. Currently the first structure is located 50h to provide room for other registers.

Device: 7-0
Function: 0
Offset: 34h

Bit	Attr	Default	Description
7:0	RO	50h	CAPPTR: Capability Pointer Points to the first capability structure (PM) in PCI 2.3 compatible space at 50h

4.8.10.25 RBAR[7:1]: ROM Base Address Register

Not implemented since it is a virtual PCI-PCI bridge

4.8.10.26 INTL[7:0]: Interrupt Line Register

The Interrupt Line register is used to communicate interrupt line routing information between initialization code and the device driver.

Device: 7-0
Function: 0
Offset: 3Ch

Bit	Attr	Default	Description
7:0	RW	00h	**INTL: Interrupt Line** This field is kept as RW for compatibility reasons only

4.8.10.27 INTP[7:0]: 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 ESI port using the appropriate Assert_Intx commands.

Device: 7-0
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.

4.8.10.28 BCTRL[7:1]: Bridge Control Register

This register provides extensions to the PEXCMD register that are specific to PCI-PCI bridges. The BCTRL provides additional control for the secondary interface (i.e. PCI Express) as well as some bits that affect the overall behavior of the "virtual" PCI-PCI bridge embedded within the Intel® 7300 Chipset, e.g. VGA compatible address range mapping.

Device: 7-1
Function: 0
Offset: 3Eh

Bit	Attr	Default	Description
15:12	RV	0h	*Reserved.* (by PCI SIG)
11	RO	0	DTSS: Discard Timer SERR Status Not applicable to PCI Express. This bit is hardwired to 0.
10	RO	0	DTS: Discard Timer Status Not applicable to PCI Express. This bit is hardwired to 0.
9	RO	0	SDT: Secondary Discard Timer Not applicable to PCI Express. This bit is hardwired to 0.
8	RO	0	PDT: Primary Discard Timer Not applicable to PCI Express. This bit is hardwired to 0.
7	RO	0	FB2BEN: Fast Back-to-Back Enable Not applicable to PCI Express. This bit is hardwired to 0.
6	RW	0	SBUSRESET: Secondary Bus Reset 1: Setting this bit causes a hot reset on the link for the corresponding PCI Express port and the PCI Express hierarchy domain subordinate to the port. This sends the LTSSM into the Hot-Reset state, which necessarily implies a reset to the downstream device and all subordinate devices. The mechanism to reset the downstream device is utilizing the TS1/TS2 "link reset" bit (bit number 0 of symbol 5). It is recommended for software/BIOS that the SBUSRESET field be held asserted for a minimum of 2 ms to ensure that the Link enters the Hot-Reset state from L0 or L1/L2. Software can also poll the PEXLNKSTS.LNKTRG bit for a deasserted condition to determine if the hot-reset state has been entered at which point it can clear the SBUSRESET field to train the link. When this SBUSRESET bit is cleared after the MCH enters the "hot-reset" state, the Intel® 7300 Chipset will initiate operations to move to "detect" state and then train the link (polling, configuration, L0 (link-up)) after sending at least 2 TS1 and receiving 1 TS1 with the HotReset bit set in the training control field of TS1 and waiting for 2 ms in the Hot-reset state. The 2ms stay in the Hot-reset state is enforced by the chipset LTSSM for the PCI Express hierarchy to reset. If the SBUSRESET is held asserted even after the 2ms timeout has expired, the Intel® 7300 Chipset will continue to maintain the hot-reset state. Hence it is necessary for software to clear this register appropriately to bring the link back in training. Note also that a secondary bus reset will not in general reset the primary side configuration registers of the targeted PCI Express port. This is necessary to allow software to specify special training configuration, such as entry into loopback mode. 0: No reset happens on the PCI Express port.
5	RO	0	MAMODE: Master Abort Mode Not applicable to PCI Express. This bit is hardwired to 0.

Device: 7-1 Function: 0 Offset: 3Eh			
Bit	**Attr**	**Default**	**Description**
4	RW	0	VGA16bdecode: VGA 16-bit decode This bit enables the virtual PCI-to-PCI bridge to provide 16-bit decoding of VGA I/O address precluding the decoding of alias addresses every 1 KB. The I/O addresses decoded is in the range of 03B0h to 03BBh or 03C0h to 03DFh within the first 1KB I/O space. 0: execute 10-bit address decodes on VGA I/O accesses. 1: execute 16-bit address decodes on VGA I/O accesses. This bit only has meaning if bit 3 (VGAEN) of this register is also set to 1, enabling VGA I/O decoding and forwarding by the bridge. This read/write bit enables system configuration software to select between 10- and 16-bit I/O address decoding for all VGA I/O register accesses that are forwarded from the primary to secondary whenever the VGAEN is set to 1.
3	RW	0	VGAEN: VGA Enable Controls the routing of CPU initiated transactions targeting VGA compatible I/O and memory address ranges. This bit may only be set for one PCI Express port.
2	RW	0	ISAEN: ISA Enable Modifies the response by the Intel® 7300 Chipset to an I/O access issued by the CPU that target ISA I/O addresses. This applies only to I/O addresses that are enabled by the IOBASE and IOLIM registers. 1: The Intel® 7300 Chipset will not forward to PCI Express any I/O transactions addressing the last 768 bytes in each 1KB block even if the addresses are within the range defined by the IOBASE and IOLIM registers. Instead of going to PCI Express these cycles will be forwarded to ESI where they can be subtractively or positively claimed by the ISA bridge. 0: All addresses defined by the IOBASE and IOLIM for CPU I/O transactions will be mapped to PCI Express.
1	RW	0	BCSERRE: SERR Enable This bit controls forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL messages from the PCI Express port to the primary side. 1: Enables forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL messages. 0: Disables forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL. **Note** that BCSERRE is no longer a gating item for the recording of the SESCSTS.SRSE error. (See Section 4.8.10.15, "SECSTS[7:1]: Secondary Status" on page 125)
0	RW	0	PRSPEN: Parity Error Response Enable This bit controls the response to poisoned TLPs in the PCI Express port 1: Enables reporting of poisoned TLP errors. 0: Disables reporting of poisoned TLP errors

4.8.10.29 IQDPRES: DMA Engine Present Control Register

This register provides control for suppressing access to the configuration space of selected devices within Intel® 7300 Chipset. Specifically, the BIOS can enable/disable DMA Engine configuration and memory mapped operations to device 8, function 0 and device 8, function 1 respectively. This is a special register intended to suppress the "yellow-bang" warning for the DMA Engine device for Intel® 7300 Chipset customers who install non-standard Operating systems without associated drivers. It can also be used to disable Intel® QuickData technology from being used in the Intel® 7300 Chipset. Note that JTAG/SMBUS can access the region independent of the state of this register field.

Device: 0
Function: 0
Offset: 44h

Bit	Attr	Default	Description
15:1	RV	0h	***Reserved***.
0	RWO	0	IQD_CFG_ENABLE: Intel® QuickData Technology Configuration Enable 1: Enable Configuration/Memory mapped accesses to the Intel® QuickData Technology configuration space located in Device 8, Fn 0 and Fn 1 including configuration, fast/slow path for IQD_BAR MMIO (inbound/FSB). 0: Disable Intel® QuickData Technology configuration accesses to Device 8, Fn 0 and Fn 1. Intel® 7300 Chipset will master abort requests to Intel® QuickData Technology configuration/memory mapped space.

4.8.10.30 PEXCTRL[7:0]: PCI Express Control Register

This 32-bit register implements chipset specific operations for general control/accessibility such as device hiding, selective configuration cycles and interrupt signalling.

Device: 7-0
Function: 0
Offset: 48h

Bit	Attr	Default	Description
31	RW	0	**MSICOREN**: **MSI for Correctable errors enable** 1: Generate MSI for correctable errors 0: Do not generate MSI for correctable errors
30	RW	0	**MSINFATEN**: **MSI for Non Fatal errors enable** 1: Generate MSI for Non Fatal (uncorrectable) errors 0: Do not generate MSI for Non Fatal (uncorrectable) errors
29	RW	0	**EN_HP_INB_MSG: Enable Hotplug inband messages** 0: Disables in-band hot-plug button & indicator messages (default) 1: Enables in-band hot-plug button & indicator messages Since SIOM is not supported in Intel® 7300 Chipset and following the PCI Express Hotplug ECR for 1.0a, the feature need not be validated (inband message enable).
28	RO	0	**GPI**: **General Purpose Input** This status bit is intended to be used by platform VRM to indicate power good status to the external HPU. e.g. PCI Express card. If used 0: indicates no power good to HPU; 1: indicates power good to HPU. This field is applicable only to Hotplug capable ports 7-1. For ESI, it is always 0. This is to be used by Hotplug software to know when the power is cut-off to the external card.
27	RW	0	**DIS_TOGGLE_POP_PRI**:
26	RW	0	**Reserved**

Device: 7-0 Function: 0 Offset: 48h			
Bit	**Attr**	**Default**	**Description**
25:24	RW	3h	**COALESCE_MODE**: Used to increase the amount of combining for completions. 00: No restriction on coalescing_hint. The IOU will try to maximize completion combining. Since Intel® 7300 Chipset issues requests in order, it does not make sense to restrict the coalesce hint because there are few resources available at the time of fetch. By the time the hint is used, resources could be freed up and reused for the following requests **Note: This mode of "00" is the preferred setting for the MCH if COALESCE_EN=1 for software/BIOS** 01: #CPL_ENTRIES_FREE will restrict coalesce_hint 10: if set then #PF_PEND will restrict coalesce hint 11: Minimum of coalesce_hint obtained from settings "01" and "10"
23	RW	0	**TIMEOUT_ENABLE_CFG: Timeout enable for configuration transactions** 1: Config transactions can time out. 0: Config transactions cannot time out. Suggested value: 0 ***Note:*** In general, configuration timeouts on the PCI Express port should not be enabled. This is necessary to permit slow devices nested deep in the PCI hierarchy that may take longer to complete requests than the maximum timeout specified in Intel® 7300 Chipset. Software/BIOS should set this field based on the context and usage/platform configuration. For e.g. compliance testing with a known broken card should have this field set. ***Note:*** For the configuration timeout to take effect, the PEXCTRL.TIMEOUT_ENABLE (bit 22) has to be set.
22	RW	0	**TIMEOUT_ENABLE: Timeout enable for non-configuration transactions** 1: Non config transactions can time out. 0: Non config transactions cannot time out. Suggested value: 1 ***Note:*** When both TIMEOUT_ENABLE_CFG and TIMEOUT_ENABLE fields are set to 0, Intel® 7300 Chipset will assume an infinite completion time for the respective transactions. Hence the system is dependent on the end device returning the completion response at some point in time, else it will result in a hang.
21	RW	0	**MALTLP_EN:** 1: Check for certain malformed TLP types. 0: Do not check for certain malformed TLP types. Suggested value: 1 When this bit is set, it enables the following conditions to mark a packet as malformed: • 4DW header MEM_RD or MEM_WR and the address is less than 32 bits (address[39:32] = 0) • Byte enable check for mem/io/cfg requests. • Length > 1 DW and (first dword byte enables = 0 or last dword byte enables = 0) Length = 1 DW and last dword byte enables != 0 • I/O{rd,wr}/cfg{rd,wr}{0,1} and (traffic class != 0 or attributes != 0 or length != 1) • A configuration retry completion response (CRS) received for a non-cfg outbound request
20	RW	0	**DISINT_MA**: 1: Disables blocking memory writes to FEEh range. 0: Enables blocking memory writes to FEEh range.
19:13	RV	0h	*Reserved*

Device: 7-0
Function: 0
Offset: 48h

Bit	Attr	Default	Description
12	RW	0	**Max_rdcmp_lmt_EN: Maximum Read completion combining limit Enable** 1: Up to 256B return and **COALESCE_EN** = 1. 0: Up to 128B return if **COALESCE_EN** = 1
11	RW	0	**COALESCE_FORCE: Force coalescing of accesses.** When 1, forces Intel® 7300 Chipset to wait for all coalescable data before sending the transaction as opposed to forwarding as much as possible. 0: Normal operation 1: wait to coalesce data
10	RW	0	**COALESCE_EN: Read completion coalescing enable** **When 1, enables read return of >64B.** 1: Returns of >64B enabled. (See **Max_rdcmp_lmt_EN above)**. 0: Returns are 64B or less. ***Note:*** For optimal read completion combining, this field should be set to '1' along with **Max_rdcmp_lmt_EN being set**
9	RW	0	**PMEGPEEN: PME GPE Enable** 1: Enables "assert_pmegpe" (deassert_pmegpe) messages to be sent over the ESI from the root complex for PM interrupts. 0: Disables "assert_pmegpe" (deassert_pmegpe) messages for PM events to the root complex. This has an overriding effect to generate ACPI PM interrupts over traditional interrupts (MSI/intx).
8	RW	0	**HPGPEEN: Hotplug GPE Enable** 1: Enables "assert_hpgpe" (deassert_hpgpe) messages to be sent from the root complex for Hot-plug events. 0: Disables "assert_hpgpe" (deassert_hpgpe) messages for Hot-plug events from the root complex. This has an overriding effect to generate ACPI HP events over traditional interrupts. Refer to the Hotplug flow defined in Figure 4-5, "PCI Express Hot Plug Interrupt Flow" on page 162 for further reference.
7	RWO	1	DIS_INB_P2PCFG: Disable Inbound Peer to Peer Configuration This bit is used for controlling peer configuration request[a] accesses through the PCI Express port in the Intel® 7300 Chipset (security feature). 1: inbound Peer to Peer configuration accesses through the Intel® 7300 Chipset will be master aborted by the I/O cluster. 0: Allows inbound configuration (Peer to Peer) access through Intel® 7300 Chipset. This register bit is of type "write once" and is controlled by BIOS/special initialization firmware and applies only to peer-to-peer configuration. The default value is to disable inbound peer to peer configuration access. Note that inbound configuration accesses targeting Intel® 7300 Chipset internal registers is controlled separately by the PEXCTRL3.EN_INB_CSCFG bit in Section 4.8.10.32, "PEXCTRL3[7:0]: PCI Express Control Register 3" on page 138. Peer-to-peer configuration between ESI and other PCI Express ports are also allowed based on this register field setting.
6:3	RWST	0000	**VPP: Virtual Pin Port** [6:4] = SMBus Address, [3] =I/O Port defines the 8-bit I/O port that is used for routing power, attention, Hotplug, presence, MRL and other events defined in Section 4.8.13.10, "PEXSLOTCTRL[7:0]: PCI Express Slot Control Register" on page 157.

Device: 7-0 Function: 0 Offset: 48h			
Bit	**Attr**	**Default**	**Description**
2	RWST	1	DIS_VPP: Disable VPP The Intel® 7300 Chipset will use this bit to decide whether the VPP is valid or not for the given PCI Express port as set by configuration software. For example, to distinguish HP events for a legacy card or PCI Express port module, this bit can be used. 1: VPP is disabled for this PCI Express port. 0: VPP is enabled for this PCI Express port. Default value is to disable vpp for the PCI Express port
1	RW	0	DIS_APIC_EOI; Disable APIC EOI The Intel® 7300 Chipset will use this bit to decide whether end of interrupts (EOI) need to be sent to an APIC controller/bridge (e.g. PXH) through this PCI Express device. 1: no EOIs are sent (disabled). 0: EOIs are dispatched to the APIC Controller. Note: In general, EOI should be disabled for active ports that have a non-I/O APIC controller attached to them for performance considerations.
0	(port 7-1) = RWO (port 0) = RV	0	DEVHIDE: Device_hide The device hide bit is used to enable the Intel® 7300 Chipset to "hide" the PCI Express device from the Operating system and is applicable only to ports 7-1. Typically, an external I/O processor acts as its proxy by configuring it and claiming resources on behalf of it and then unhides. The hiding is done by changing the class code (CCR register) for this port to 0x0600. This will prevent the OS from attempting to probe or modify anything related to this device. 1: The PCI Express port CCR register has a value of 0600. 0: The PCI Express port CCR register has a value of 0604 (bridge) The default value is '0' (to make the device a bridge). The device hide bit does not apply to the ESI interface (port 0) and has no effect on its operation ***Note:*** Also refer to PEXCTRL4.PTHIDE register filed that sets the DID and VID to all 1's in order to remove the PCI Express port from BIOS/OS scan.

Notes:

a. Note that inbound MMCFG accesses are disallowed irrespective of the settings of these bits.

4.8.10.31 PEXCTRL2[7:0]: PCI Express Control Register 2

Device: 7-0 Function: 0 Offset: 4Ch			
Bit	**Attr**	**Default**	**Description**
7:1	RV	0	***Reserved***.
0	RW	0	NO_COMPLIANCE: Set by software to enable link operation in the presence of single wire failures on the link. If clear, then specified link behavior in the presence of a wire failure will be Polling.Compliance. **Note**: This is an One Generation Ahead (OGA) feature where the BIOS sets this field to inform it is a customer/target system and not a system under test. Intel® 7300 Chipset then degrades a link rather than take down all the lanes to Polling.Compliance as per the *PCI Express Base Specification*, Revision 1.0a for some fault models.

4.8.10.32 PEXCTRL3[7:0]: PCI Express Control Register 3

Device: 7-0
Function: 0
Offset: 4Dh

Bit	Attr	Default	Description
7	RV	0	***Reserved***.
6	RW	0	**COALESCE_64B_ALIGN: Enables coalescing to occur on the beginning of any 64B cacheline boundary.** 0: Coalescing can ONLY occur on multiple cacheline read requests that begin on a 128B aligned address boundaries. i.e. read requests whose address is aligned to 0x80 or 0x00. 1: Coalescing can occur on any multiple cacheline read request that begin on a 64B aligned address boundaries.
5	RW	0	**DIS_PFIBRD: Disable Prefetch for Inbound Reads** 0: Enable the PCI Express port for issuing prefetches for inbound reads (default mode for performance) 1: Disable the PCI Express port from issuing prefetches for inbound reads.
4:3	RV	port 0 = 0h ports (3-1) = 2h ports(7-4) = 3h	**Reserved**
2	RWO	(port 3-2) = 0 others = 1	DIS_IQD_BAR: Disable access to IQD_BAR Memory mapped space This bit is used for controlling IQD_BAR MMIO accesses from a PCI Express port in the Intel® 7300 Chipset (top level). 1: Disables IQD_BAR Memory mapped access and all requests to IQD_BAR will be Master aborted on the PCI Express interface. 0: Enables accesses to IQD_BAR region. This register bit is of type "write once" and is controlled by BIOS/special initialization firmware. The default value is "0" for ports 2 and 3 (i.e IQD_Bar access is allowed) while it is disabled for the rest of the ports. If this bit is enabled for the rest of the ports other than 2 and 3, then partial access to manipulate Intel® I/O Acceleration Technology resources (DMA only) is available to the end device/card connected to the enabled PCI Express port.
1	RWO	0	EN_INB_CSCFG: Enable Inbound Chipset configuration access This bit is used for controlling configuration request[a] accesses exclusively for the Intel® 7300 Chipset registers through the PCI Express port (security feature). Note that inbound configuration is illegal and disallowed as *PCI Express Base Specification*, Revision 1.0a. This is intended to be a special debug feature to allow alternate access to the Chipset configuration space and should not be disclosed to external customers. The default value is to disable inbound configuration access. 1: Allow inbound configuration accesses to Intel® 7300 Chipset registers. Note inbound AMB memory mapped DIMM registers are allowed as a special case in the absence of Jtag/SMBus configs 0: Disables inbound configuration to Intel® 7300 Chipset registers. and will be master aborted by the Intel® 7300 Chipset. This register bit is of type "write once" and is controlled by BIOS/special initialization firmware.
0	RW	0	**MSIRASERREN: MSI RAS Error Enable** 1: Enables MSI messages to be sent to the root complex for RAS Error events on PCI Express ports. 0: Disables sending of MSI messages for RAS Error events on PCI Express ports to the root complex. Note that for MSI RAS Error messages to be sent, both MSIRASERREN and MSICTRL[7:1].MSIEN bits defined in Section 4.8.12.3, "MSICTRL[7:0]: Message Control Register" on page 143 have to be set.

Notes:

a. Note that inbound MMCFG accesses to Intel® 7300 Chipset are disallowed irrespective of the settings of these bits. Peer to peer configuration is controlled separately through the PEXCTRL.DIS_INB_P2P_CFG bit defined in Section 4.8.10.30, "PEXCTRL[7:0]: PCI Express Control Register" on page 134.

4.8.10.33 PEXGCTRL: PCI Express Global Control Register

This 32-bit global register in the Intel® 7300 Chipset implements chipset specific operations for generalized control of all PCI Express events and activity such as Power Management, Hot-plug. There is only one register for all PCI Express ports and Intel® I/O Acceleration Technology device that controls related I/O operations.

Device: 19
Function: 0
Offset: 17Ch

Bit	Attr	Default	Description
31:18	RW	3FFFh	**Timeout: Completion Time out** Internal timer for handling Outbound NP completion timeouts. This varies based on the core clock frequency and the time at which the completion structure is loaded relative to the timeout timer which is free-running. The bounds of this roll over can be approximated as a Minimum of 6 or Max of 7 +/- few cycles) since there is a 3 bit counter whose roll over is tied to the timeout timer For 333 MHz, the granularity of this timer viz. each increment is in the range (9216 ns, 10,752 ns) giving a min/max value for a full face value of this register field as (150.99ms, 176.15 ms) For 266 MHz, the granularity of this timer viz. each increment is in the range (11520 ns, 13440 ns) giving a min/max value for a full face value of this register field as (188.73 ms, 220.19 ms) BIOS/Software needs to set this field as appropriate for handling various timeout conditions required by the system. ***Note:*** For example with Intel® 7300 Chipset running at 333 Mhz, for SMBUS protocols, the maximum value recommended for this field is 0x744 (or 1860 decimal) to achieve a 20ms timeout threshold(i.e 20 ms =~ 10,752 * 1860) such that it provides headroom to the chipset for the global SMBUS timeout of 25 ms.
17	RW	0h	**Reserved**
16	RW	0	IOU_CE_TXN_THROTTLE: IOU CE Transaction Throttling Mode 0: The IOU will utilize all 48 entries allocated by the CE for scheduling inbound transactions (default mode). 1: The IOU will ensure that the total number of inbound read/write transactions does not exceed 41 by reducing the number of write by 1 for each port. The following allocation will be used in Intel® 7300 Chipset in this mode: Ports 0,4,5,6,7 will use (3R+2W) while ports 2,3 will use (4R+3W). Total: 5x5+2x7=39 + 2 for interrupts = 41 entries.
15:13	RW	010	DMA_ARB_RATIO: DMA Arbitration Ratio Sets the number of successive cacheline requests made by the DMA unit of Intel® 7300 Chipset to the DM resource arbiter using a weighted round robin scheme. 010: 2 Cachelines (default) 100: 4 Cachelines (backup) Others - *Reserved*
12:10	RW	011	IOU0_ARB_RATIO: IOU0 Arbitration Ratio Sets the number of successive cacheline requests made by the IOU0 unit of Intel® 7300 Chipset to the DM resource arbiter. 010: 2 Cachelines 011: 3 Cachelines (default) 100: 4 Cachelines (backup) Others - *Reserved*

Device: 19 Function: 0 Offset: 17Ch			
Bit	**Attr**	**Default**	**Description**
9:7	RW	100	IOU1_ARB_RATIO: IOU1 Arbitration Ratio Sets the number of successive cacheline requests made by the IOU1 unit of Intel® 7300 Chipset to the DM resource arbiter. 100: 4 Cachelines (default) (for 4 GB/s) 010: 2 Cachelines (backup) Others - *Reserved*
6:4	RW	001	Reserved
3	RW	0	DISERDCMPHDR: Disable Extra Inbound Read completion header queue 0: Enables performance by providing extra Inbound Read completion structure header queues for stage buffering (default) 1: Disables extra Inbound Read completion structure.
2	RW	1	ENFDDOUT: Enable Fast Data Done Signal to Outbound layer 0: Disables the fast done signal to the outbound layer of the IOU clusters (default) 1: Enables the fast data done signal to be passed to the outbound transaction layer for indicating the start of the packet (for better performance) in the IOU clusters
1	RWST	0	PME_TURN_OFF: Send PME Turn Off Message[a] When set, the Intel® 7300 Chipset will issue a PME Turn Off Message to all enabled PCI Express ports excluding the ESI port.[b] The Intel® 7300 Chipset will clear this bit once the Message is sent.
0	RWC	0	PME_TO_ACK: Received PME Turn Off Acknowledge Message[a] Intel® 7300 Chipset sets this bit when it receives a PME_TO_ACK Message from all enabled PCI Express ports excluding the ESI port. Software will clear this bit when it handles the Acknowledge. Note that the Intel® 631xESB/632xESB I/O Controller Hub will not generate a PME_TO_Ack based on the flow described in the ESI spec. However, if a PME_TO_Ack is received at the Intel® 7300 Chipset ESI port, it will be Master Aborted.

Notes:

a. Note that PME_TURN_OFF and PME_TO_ACK bits are defined in the PEXGCTRL register (global) since the intent was to have one and only one of these status bits unique for all the seven PCI Express ports in Intel® 7300 Chipset.

b. The ESI port does not support PM_Turn_off, PM_TO_Ack messages and they are synchronized using the Go_Sx, Ack_Sx messages.

4.8.11 PCI Express Power Management Capability Structure

The Intel® 7300 Chipset PCI Express port provides basic power management capabilities to handle PM events for compatibility. The PCI Express ports can be placed in a pseudo D3hot state but it does have not real power savings and works as if it were in the D0 mode.

4.8.11.1 PMCAP[7:0]: 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.

Device: 7-0
Function: 0
Offset: 50h

Bit	Attr	Default	Description
31:27	RO	11001	PMES: PME Support Identifies power states in the Intel® 7300 Chipset which can send an "Assert_PMEGPE/Deassert PMEGPE" message. Bits 31, 30 and 27 must be set to '1' for PCI-PCI bridge structures representing ports on root complexes. The definition of these bits is taken from the PCI Bus Power Management Interface Specification Revision 1.1. XXXX1b - Assert_PMEGPE/Deassert PMEGPE can be sent from D0 XXX1Xb - Assert_PMEGPE/Deassert PMEGPE can be sent from D1 **(Not supported by the MCH)** XX1XXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D2 **(Not supported by the MCH)** X1XXXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D3 hot **(Supported by the MCH)** 1XXXXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D3 cold **(Supported by MCH)**
26	RO	0	D2S: D2 Support Intel® 7300 Chipset does not support power management state D2.
25	RO	0	D1S: D1 Support Intel® 7300 Chipset does not support power management state D1.
24:22	RO	0h	AUXCUR: AUX Current
21	RO	0	DSI: Device Specific Initialization
20	RV	0	*Reserved.*
19	RO	0	PMECLK: PME Clock This field is hardwired to 0h as it does not apply to PCI Express.
18:16	RO	010	VER: Version This field is set to 2h as version number from the *PCI Express Base Specification*, Revision 1.0a specification.
15:8	RO	58h	NXTCAPPTR: Next Capability Pointer This field is set to offset 58h for the next capability structure (MSI) in the PCI 2.3 compatible space.
7:0	RO	01h	CAPID: Capability ID Provides the PM capability ID assigned by PCI-SIG.

4.8.11.2 PMCSR[7:0]: Power Management Control and Status Register

This register provides status and control information for PM events in the PCI Express port of the Intel® 7300 Chipset.

Device: 7-0
Function: 0
Offset: 54h

Bit	Attr	Default	Description
31:24	RO	00h	Data: Data Data read out based on data select (DSEL). Refer to section 3.2.6 of PCI PM specification for details. This is not implemented in the Power Management capability for Intel® 7300 Chipset and is hardwired to 0h.
23	RO	0h	BPCCEN: Bus Power/Clock Control Enable This field is hardwired to 0h as it does not apply to PCI Express.

Device: 7-0 Function: 0 Offset: 54h			
Bit	**Attr**	**Default**	**Description**
22	RO	0h	B2B3S: B2/B3 Support This field is hardwired to 0h as it does not apply to PCI Express.
21:16	RV	0h	*Reserved.*
15	RWCST	0h	PMESTS: PME Status This PME Status is a sticky bit. When set, the PCI Express port generates a PME internally independent of the PMEEN bit defined below. Software clears this bit by writing a '1' when it has been completed. As a root port, the Intel® 7300 Chipset will never set this bit, because it never generates a PME internally independent of the PMEEN bit.
14:13	RO	0h	DSCL: Data Scale This 2-bit field indicates the scaling factor to be used while interpreting the "data_scale" field.
12:9	RO	0h	DSEL: Data Select This 4-bit field is used to select which data is to reported through the "data" and the "Data Scale" fields.
8	RWST	0h	PMEEN: PME Enable This field is a sticky bit and when set enables PMEs generated internally to appear at the Intel® 631xESB/632xESB I/O Controller Hub through the "Assert(Deassert)_PMEGPE"message. This has no effect on the Intel® 7300 Chipset since it does not generate PME events internally
7:2	RV	0h	*Reserved.*
1:0	RW	0h	PS: 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 (reserved) 10: D2 (reserved) 11: D3_hot If Software sets this to D1 or D2, then the power state will default to D0.

4.8.12 PCI Express Message Signalled Interrupts (MSI) Capability Structure

Message Signaled Interrupts (MSI) is an optional feature that enables a device to request service by writing a system-specified message to a system-specified address in the form of an interrupt message. The transaction address (e.g. FEEx_xxxxh) specifies the message destination and the transaction data specifies the message. The MSI mechanism is supported by the following registers: the MSICAPID, MSINXPTR, MSICTRL, MSIAR and MSIDR register described below.

4.8.12.1 MSICAPID[7:0]: MSI Capability ID

Device: 7-0 Function: 0 Offset: 58h			
Bit	**Attr**	**Default**	**Description**
7:0	RO	05h	CAPID: Capability ID Assigned by PCI-SIG for message signaling capability.

4.8.12.2 MSINXPTR[7:0]: MSI Next Pointer

Device: 7-0 Function: 0 Offset: 59h			
Bit	**Attr**	**Default**	**Description**
7:0	RO	6Ch	NXTPTR: Next Ptr This field is set to 6Ch for the next capability list (PCI Express capability structure - PEXCAP) in the chain.

4.8.12.3 MSICTRL[7:0]: Message Control Register

Device: 7-0 Function: 0 Offset: 5Ah			
Bit	**Attr**	**Default**	**Description**
15:8	RV	00h	*Reserved.*
7	RO	0	AD64CAP: 64-bit Address Capable This field is hardwired to 0h since the message writes addresses are only 32-bit addresses (e.g. FEEx_xxxxh).
6:4	RW	000	MMEN: Multiple Message Enable 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. See below for discussion on how the interrupts are handled if N is the number of messages by software. ***Note:*** If software writes a value greater than the limit specified by the MMCAP field in the MMEN field, it is considered as a programming error. The Intel® 7300 Chipset will only use the LSB of the MMEN (as a power of 2) to decode the Table 4-37, "IV Handling and Processing by Intel® 7300 Chipset" on page 145 for up to 2 messages.
3:1	RO	001	MMCAP: Multiple Message Capable Software reads this field to determine the number of requested messages. which is aligned to a power of two. It is set to 2 messages (encoding of 001). The Intel® 7300 Chipset is designed to handle MSI's for different events HP/PM events RAS Error events
0	RW	0	MSIEN: MSI Enable The software sets this bit to select legacy interrupts or transmit MSI messages. 0: Disables MSI from being generated. 1: Enables the MCH to use MSI messages to request context specific service through register bits defined in the Section 4.8.10.30, "PEXCTRL[7:0]: PCI Express Control Register" on page 134 for events such as Hot-plug, PM, RAS.

4.8.12.4 MSIAR[7:0]: MSI Address Register

The MSI Address Register (MSIAR) contains the system specific address information to route MSI interrupts and is broken into its constituent fields.

Device: 7-0 Function: 0 Offset: 5Ch			
Bit	**Attr**	**Default**	**Description**
31:20	RW	FEEh	AMSB: Address MSB This field specifies the 12 most significant bits of the 32-bit MSI address.

Device: 7-0 Function: 0 Offset: 5Ch			
Bit	**Attr**	**Default**	**Description**
19:12	RW	00h	ADSTID: Address Destination ID This field is initialized by software for routing the interrupts to the appropriate destination.
11:4	RW	00h	AEXDSTID: Address Extended Destination ID This field is not used by IA-32 processor.
3	RW	0h	ARDHINT: Address Redirection Hint 0: directed 1: redirectable
2	RW	0h	ADM: Address Destination Mode 0: physical 1: logical
1:0	RV	0h	***Reserved.*** Not used since the memory write is D-word aligned

4.8.12.5 MSIDR[7:0]: MSI Data Register

The MSI Data Register (MSIDR) contains all the data (interrupt vector) related information to route MSI interrupts.

Device: 7-0 Function: 0 Offset: 60h			
Bit	**Attr**	**Default**	**Description**
31:16	RV	0000h	*Reserved.*
15	RW	0h	TM: Trigger Mode This field Specifies the type of trigger operation 0: Edge 1: level
14	RW	0h	LVL: Level if TM is 0h, then this field is a don't care. Edge triggered messages are always treated as assert messages. For level triggered interrupts, this bit reflects the state of the interrupt input if TM is 1h, then 0: Deassert Messages 1: Assert Messages
13:11	RW	0h	These bits are don't care in IOxAPIC interrupt message data field specification.
10:8	RW	0h	**DM: Delivery Mode** 000: Fixed 001: Lowest Priority 010: SMI/HMI 011: *Reserved* 100: NMI 101: INIT 110: *Reserved* 111: ExtINT
7:0	RW	0h	**IV: Interrupt Vector** The interrupt vector (LSB) will be modified by the Intel® 7300 Chipset 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 in Section 4.8.12.3, "MSICTRL[7:0]: Message Control Register" on page 143, and Table 4-37 illustrates the breakdown.

Table 4-37. IV Handling and Processing by Intel® 7300 Chipset

Number of Messages enabled by Software (MSICTRL.MMEN)	Events	IV[7:0]
1	All	xxxxxxxx[a]
2	HP, PM	xxxxxxx0
	RAS errors	xxxxxxx1

Notes:

a. The term "xxxxxx" in the Interrupt vector denotes that software/BIOS initializes them and the Intel® 7300 Chipset will not modify any of the "x" bits except the LSB as indicated in the table as a function of MMEN

4.8.12.6 PEXCTRL4[7:0]: PCI Express Control Register 4

Device: 7-0
Function: 0
Offset: 64h

Bit	Attr	Default	Description
31:22	RV	0	***Reserved.***
21	(port 0) = RW (port 7-1) = RV	0	**DISP2PESI: Disable Peer to Peer requests targeting ESI** This register field is valid only on the ESI port and is used for blocking Peer-to-peer requests targeting the ESI. 0: Allow full Peer to Peer traffic from any PCI Express port to ESI (default operation). 1: Master abort Peer to Peer Requests going to ESI from any PCI Express port ***Note:*** This field affects only Memory, I/O requests targeting ESI. Peer to Peer Configuration requests are controlled separately by PEXCTRL.DIS_INB_P2P_CFG register bit.
20:15	RW	0h	**Reserved**
14	RW	0	**PTHIDE: Port Hide through DID and VID fields** 0:Normal operation DID and VID will return the default values as depicted in the configuration map for the PCI Express ports. See Section 4.8.1.1, "VID: Vendor Identification Register" on page 88 and Section 4.8.1.2, "DID: Device Identification Register" on page 88. (default) 1: The MCH will return 0xFFFF for each of the DID and VID fields. This process mimics a Master abort for that port and will prevent the device from being enumerated during PCI scan. However other register fields of the port will still be visible to software and will not get master aborted[a]. This field is independent of the PEXCTRL.DEVHIDE field ***Note:*** This field is only applicable to Ports7-1. For the ESI (port 0), this bit has no effect. ***Note:*** This register field is used to hide the device for example during Linux OS enumeration scan and prevent boot error messages caused by setting the PEXCTRL.DEVHIDE bit. (i.e header type mismatches for host bridge).
13	RW	0	**DISWRPERFOPT: Disable Write performance/ordering optimization** 0: Cache line writes of a single inbound multi-line write transaction to main memory are pipelined to the coherency engine, and these may be completed out of order by the coherency engine. 1: Disables write optimization: The IOU will send successive writes to memory following the strong ordering rules. This only affects the PCI Express to memory path.
12:8	RW	10000	**SB_MSG_THROTTLE_LIMIT: Sideband Message Throttle Limit** This field defines the threshold for the number of side band messages that is actively processed for each message type by the outbound unit before it backpressures the inbound unit. It is typically used to handle queue overflow and prevent loss of Assert_intx, deassert_intx, assert_GPE, assert_PME messages which may arrive asynchronously at any of the ports. The default value is set to 16 decimal (0x10). Any number between 1 and 31 is valid. A value of 0 would disable the throttling/backpressure scheme.

Device: 7-0 Function: 0 Offset: 64h			
Bit	**Attr**	**Default**	**Description**
7	RW	0	**LK_CMP_MA_EN: Lock Completion Master Abort enable** 0: The MCH IOU cluster expects a CplLk (unsuccessful) or CplDLk (successful) from the PCI Express device during the lock flow as specified in the *PCI Express Base Specification*, Revision 1.0a. (default golden behavior) 1: The MCH IOU cluster can expect a Cpl (unsuccessful) or a CplLk (unsuccessful) or CplDLk (successful) from the connected PCI Express device /bridge during the lock flow. **Note**: It is recommended that BIOS/Software program this field to '1' for those PCI Express ports in the MCH which is connected to the Intel® 631xESB/632xESB I/O Controller Hub.
6	RW	0	**DCAEN: DCA Enable** 0: Disable mode for DCA operation on the PCI Express/ESI port 1: Enable mode for DCA operation on the PCI Express/ESI port This bit controls whether the PCI Express/ESI port is allowed to process DCA and acts as a master override for DCA operations on that port. BIOS should enable this mode for those ports where a DCA capable end device is attached to the MCH/Intel® 631xESB/632xESB I/O Controller Hub.
5	RW	0	**en_abort_in_d3: Enable abort in D3** 0: Disable mode for aborting Read and Write request 1: Enable mode for aborting Read and Write request **Note:** The bits 5 to 0 in this register constitute debug hooks. The Power Management specification 1.1 says that power management states need to be exited based on config transactions. These bits provide workaround to handle cases where the MCH can abort posted transactions, completions in specific power states (non D0).
4	RW	0	**en_abort_in_d2: Enable abort in D2** 0: Disable mode for aborting Read/write request 1: Enable mode for aborting Read/Write request
3	RW	0	**en_abort_in_d1: Enable abort in D1** 0: Disable mode for aborting ReadWrite request 1: Enable mode for aborting Read/Write request
2	RW	1	**abort_p_if_non_d0: Abort posted requests in non d0** 0: No posted requests are aborted 1: Abort posted requests This is gated by bits[5:3] setting in this register
1	RW	1	**abort_np_if_non_d0: Abort non-posted requests in non d0** 0: Non-posted requests are aborted 1: Abort non-posted requests This is gated by bits[5:3] setting in this register
0	RW	0	**abort_c_if_non_d0: Abort completions in non d0** 0: Completions are not aborted 1: Completions are aborted

Notes:

a. It is the expectation that software will not probe the configuration register space beyond when it reads all 1's for the DID/VID fields.

4.8.12.7 PEXCTRL5[7:0]: PCI Express Control Register 5

Device: 7-1 Function: 0 Offset: 68h			
Bit	**Attr**	**Default**	**Description**
31:2	RV	0	***Reserved***.

Device: 7-1 Function: 0 Offset: 68h			
Bit	**Attr**	**Default**	**Description**
1	RW	0h	**DISINBPDS: Disable Inband Presence Detect** This register field will be used by the LTSSM to disable the "inband presence detect" mechanism using the State transitions described in Table 4-4 of the *PCI Express Base Specification*, Revision 1.0a. This field can be used by the Software for controlling the presence detect of a card using PEXSLOTCTRL.PDS bit. See Section 4.8.13.7, "PEXLNKCTRL[7:0]: PCI Express Link Control Register" on page 154 0: Enable inband presence detect (default) 1: Disable inband presence detect
0	RW	0	**DISASPML1Hang: Disable ASPM L1 Hang** This register field is used to prevent MCH from hanging during an ASPM L1 transition by a PCI Express Card (s500481) when enabled (value of '0').

4.8.13 PCI Express Capability Structure

The PCI Express capability structure describes PCI Express related functionality, identification and other information such as control/status associated with the port. It is located in the PCI 2.3 compatible space and supports legacy operating system by enabling PCI software transparent features.

4.8.13.1 PEXCAPL[7:0]: PCI Express Capability List Register

The PCI Express Capability List register enumerates the PCI Express Capability structure in the PCI 2.3 configuration space.

Device: 7-0 Function: 0 Offset: 6Ch			
Bit	**Attr**	**Default**	**Description**
15:8	RO	(port 0) = 00h others = F0h	NXTPTR: Next Ptr This field is set to 0xF0h pointer to point to the SVID/SID register.
7:0	RO	10h	CAPID: Capability ID Provides the PCI Express capability ID assigned by PCI-SIG.

4.8.13.2 PEXCAP[7:0]: PCI Express Capabilities Register

The PCI Express Capabilities register identifies the PCI Express device type and associated capabilities.

Device: 7-0 Function: 0 Offset: 6Eh			
Bit	**Attr**	**Default**	**Description**
15:14	RV	0h	*Reserved.*

Device: 7-0 Function: 0 Offset: 6Eh			
Bit	**Attr**	**Default**	**Description**
13:9	RO	00h	**IMN: Interrupt Message Number** This field indicates the interrupt message number that is generated from the PCI Express port. When there are more than one MSI interrupt Number, this register field is required to contain the offset between the base Message Data and the MSI Message that is generated when the status bits in the slot status register or root port status registers are set. The chipset is required to update the field if the number of MSI messages changes.
8	(port 7-1) = RWO (port 0) = RO	0	**SLOT_Impl: Slot Implemented** 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. For the ESI port, this value should be 0b always since it is not hot-pluggable and it is required for boot. Rest of the PCI_Express ports which are slotted/hot-pluggable, BIOS or Software can set this field to enable the slots. Also refer to PEXSLOTSTS.PDS field in Section 4.8.13.11, "PEXSLOTSTS[7:0]: PCI Express Slot Status Register" on page 159
7:4	RO	0100	DPT: Device/Port Type This field identifies the type of device. It is set to 0100 as defined in the spec since the PCI Express port is a "root port" in the Intel® 7300 Chipset. ***Note:*** This field should have been defined as "0000" for PEXCAP[0], the ESI port, since it is an end point rather than a P2P bridge. However since the CCR is set to 0600h (host bridge) for the ESI port, this should not cause an issue with Operting Systems.
3:0	RO	0001	VERS: Capability Version This field identifies the version of the PCI Express capability structure. Set to 0001 by PCI SIG.

4.8.13.3 PEXDEVCAP[7:0]: PCI Express Device Capabilities Register

The PCI Express Device Capabilities register identifies device specific information for the port.

Device: 7-0 Function: 0 Offset: 70h			
Bit	**Attr**	**Default**	**Description**
31:28	RV	0h	*Reserved.*
27:26	RO	0h	**CSPLS: Captured Slot Power Limit Scale** Specifies the scale used for the Captured Slot Power Limit Value. It does not apply to Intel® 7300 Chipset as it is a Root complex. Hardwired to 0h.
25:18	RO	00h	**CSPLV: Captured Slot Power Limit Value** This field specifies upper limit on power supplied by a slot in an upstream port. It does not apply to Intel® 7300 Chipset as it is a Root complex. Hardwired to 00h.
17:16	RV	0h	*Reserved*
15	RO	1	**RBER: Role Based Error Reporting** This bit indicates that the MCH supports this feature.
14	RO	0	**PIPD: Power Indicator Present on Device** This bit when set indicates that a Power Indicator is implemented. 0: PIPD is disabled in the MCH. 1: *Reserved*

Device: 7-0
Function: 0
Offset: 70h

Bit	Attr	Default	Description
13	RO	0	**AIPD: Attention Indicator Present** This bit when set indicates that an Attention Indicator is implemented. 0: AIPD is disabled in the MCH. 1: *Reserved*
12	RO	0	**ABPD: Attention Button Present** This bit when set indicates that an Attention Button is implemented. 0: ABPD is disabled in the MCH. 1: *Reserved*
11:9	RO	111	**EPL1AL: 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. 000: Less than 1µs 001: 1 µs to less than 2 µs 010: 2 µs to less than 4 µs 011: 4 µs to less than 8 µs 100: 8 µs to less than 16 µs 101: 16 µs to less than 32 µs 110: 32 µs to 64 µs 111: More than 64 µs **The MCH does not support EndpointL1 acceptable latency and is set to the maximum value for safety.**
8:6	RO	111	**EPL0AL: Endpoints L0s Acceptable Latency** This field indicates the acceptable latency that an Endpoint can withstand due to the transition from L0s state to the L0 state. 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 µs 101: 1 µs to less than 2 µs 110: 2 µs to 4 µs 111: More than 4 µs Note that the MCH does not support L0s implementation and for backup, this field is set to the maximum value.
5	RO	0	**ETFS: Extended Tag Field Supported** This field indicates the maximum supported size of the Tag field. 0: In the Intel® 7300 Chipset, only 5-bit Tag field is supported
4:3	RO	0h	**PFS: Phantom Functions Supported** This field indicates the number of most significant bits of the function number portion of Requester ID in a TLP that are logically combined with the Tag identifier. 0: For root ports, no function number bits for phantom functions are supported
2:0	RO	001	**MPLSS: Max Payload Size Supported** This field indicates the maximum payload size that the PCI Express port can support for TLPs. **001: 256B max payload size** Others - *Reserved* **Note that the MCH only supports up to a maximum of 256B payload (e.g. writes, read completions) for each TLP and violations will be flagged as PCI Express errors**

4.8.13.4 PEXDEVCTRL[7:0]: PCI Express Device Control Register

The PCI Express Device Control register controls PCI Express specific capabilities parameters associated with this port.

Device: 7-0
Function: 0
Offset: 74h

Bit	Attr	Default	Description
15	RV	0h	*Reserved.*
14:12	RW	000	MRRS: Max_Read_Request_Size This field sets maximum Read Request size generated by the MCH as a requestor. The corresponding IOU logic in the MCH associated with the PCI Express port must not generate read requests with size exceeding the set value. 000: 128B max read request size Others: Reserved The MCH will not generate read requests larger than 64B in general on the outbound side (CPU initiated, DMA or Peer to Peer). Hence the field is set to 000b encoding.
11	RW	1	**ENNOSNP: Enable No Snoop** When set, the PCI Express port is permitted to set the "No Snoop bit" in the Requester Attributes of transactions it initiates that do not require hardware enforced cache coherency. Typically the "No Snoop bit" is set by an originating PCI Express device down in the hierarchy. The MCH never sets or modifies the "No snoop bit" in the received TLP even if ENNOSNP is enabled. For outbound traffic, the MCH does not need to snoop.
10	RWST	0	**APPME: Auxiliary Power Management Enable** 1: Enables the PCI Express port to draw AUX power independent of PME AUX power. 0: Disables the PCI Express port to draw AUX power independent of PME AUX power. Devices that require AUX power on legacy operating systems should continue to indicate PME AUX power requirements. AUX power is allocated as requested in the AUX_Current field on the Power Management Capabilities Register (PMC), independent of the PMEEN bit in the Power Management Control & Status Register (PMCSR) defined in Section 4.8.11.2, "PMCSR[7:0]: Power Management Control and Status Register" on page 141.
9	RO	0	**PFEN: Phantom Functions Enable** This bit enables the PCI Express port to use unclaimed functions as Phantom Functions for extending the number of outstanding transaction identifiers. the MCH does not implement this bit (Root complex) and is hardwired to 0
8	RO	0h	**ETFEN: Extended Tag Field Enable** This bit enables the PCI Express port to use an 8-bit Tag field as a requester. The MCH does not use this field (Root complex) and is hardwired to 0.
7:5	RW	000	**MPS: 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 MCH must handle TLPs as large as the set value. As a transmitter, 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 (default) 001: 256B max payload size (max allowed in the MCH) others: Reserved ***Note:*** The MCH can support max payload sizes only up to 256B. If Software programs a value that exceeds 256B for the MPS field, then it will be considered as an error. For receive TLPs, it will be flagged as "unsupported request" and for transmit TLPs, it will be recorded as a Malformed TLP.

Device: 7-0 Function: 0 Offset: 74h			
Bit	**Attr**	**Default**	**Description**
4	RO	0	**ENRORD: Enable Relaxed Ordering** The MCH enforces only strict ordering only and hence this bit is initialized to '0'
3	RW	0	**URREN: Unsupported Request Reporting Enable** This bit controls the reporting of unsupported requests to the MCH in the PCI Express port. 0: Unsupported request reporting is disabled 1: Unsupported request reporting is enabled Note that the reporting of error messages (such as ERR_CORR, ERR_NONFATAL, ERR_FATAL) received by PCI Express port is controlled exclusively by the PCI Express Root Control register (PEXRTCTRL) described in Section 4.8.13.12, "PEXRTCTRL[7:0]: PCI Express Root Control Register" on page 163.
2	RW	0	**FERE: Fatal Error Reporting Enable** This bit controls the reporting of fatal errors internal to the MCH in the PCI Express port. 0: Fatal error reporting is disabled 1: Fatal error reporting is enabled
1	RW	0	**NFERE: Non Fatal Error Reporting Enable** This bit controls the reporting of non fatal errors internal to the MCH in the PCI Express port. 0: Non Fatal error reporting is disabled 1: Non Fatal error reporting is enabled
0	RW	0	**CERE: Correctable Error Reporting Enable** This bit controls the reporting of correctable errors internal to the MCH in the PCI Express port. 0: Correctable error reporting is disabled 1: Correctable Fatal error reporting is enabled

4.8.13.5 PEXDEVSTS[7:0]: PCI Express Device Status Register

The PCI Express Device Status register provides information about PCI Express device specific parameters associated with this port.

Device: 7-0 Function: 0 Offset: 76h			
Bit	**Attr**	**Default**	**Description**
15:6	RV	000h	*Reserved.*
5	RO	0h	**TP: Transactions Pending** 1: indicates that the PCI Express port has issued Non-Posted Requests which have not been completed. 0: A device reports this bit cleared only when all Completions for any outstanding Non-Posted Requests have been received. Since the MCH Root port that do not issue Non-Posted Requests on their own behalf, it is hardwired to 0b.
4	RO	0	**APD: AUX Power Detected** 1- AUX power is detected by the PCI Express port. 0: No AUX power is detected
3	RWC	0	**URD: Unsupported Request Detected** This bit indicates that the device received an Unsupported Request in the PCI Express port. 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 port This records the detection of receiving an unsupported request, error IO2.

Device: 7-0 Function: 0 Offset: 76h			
Bit	**Attr**	**Default**	**Description**
2	RWC	0	**FED: Fatal Error Detected** This bit indicates that status of a fatal (uncorrectable) error detected in the PCI Express port. 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	RWC	0	**NFED: Non Fatal Error Detected** This bit indicates status of non-fatal errors detected. This bit gets set if a non-fatal uncorrectable error is detected in the PCI Express port. 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	RWC	0	**CED: Correctable Error Detected** This bit indicates status of correctable errors detected. This bit gets set if a correctable error is detected in the PCI Express port. 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

4.8.13.6 PEXLNKCAP[7:0]: PCI Express Link Capabilities Register

The Link Capabilities register identifies the PCI Express specific link capabilities.

Device: 7-0 Function: 0 Offset: 78h			
Bit	**Attr**	**Default**	**Description**
31:24	RWO	(port 0) = 0 (port 1) = 01h (port 2) = 02h (port 3) = 03h (port 4) = 04h (port 5) = 05h (port 6) = 06h (port 7) = 07h	PN: Port Number This field indicates the PCI Express port number for the link and is initialized by software/BIOS. This will correspond to the device number for each port. port 0- device number of 0 (ESI) port 1- device number of 1 port 2 - device number of 2 port 3 - device number of 3 port 4 - device number of 4 port 5- device number of 5 port 6- device number of 6 port 7- device number of 7
23:21	RV	0h	*Reserved.*
20	RO	1	**DLLAEN: Data Link Layer Active Reporting Capable** The MCH is capable of reporting the DL Active State of the Link Control and Management State Machine.
19	RO	1	**SLNKDEN: Surprise Link Down Error Reporting Capable** The MCH is capable of detecting and reporting a surprise link down error condition.
18	RV	0h	*Reserved.*

Device: 7-0 Function: 0 Offset: 78h			
Bit	**Attr**	**Default**	**Description**
17:15	RO	7h	**L1EL: 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 1µs 001: 1 µs to less than 2 µs 010: 2 µs to less than 4 µs 011: 4 µs to less than 8 µs 100: 8 µs to less than 16 µs 101: 16 µs to less than 32 µs 110: 32 µs to 64 µs 111: More than 64 µs ***Note:***
14:12	RO	7h	L0sEL: 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 µs 101: 1 µs to less than 2 µs 110: 2 µs to 4 µs 111: More than 4 µs Note that the MCH does not support L0s exit Latency implementation and for safety, this field is set to the maximum value.
11:10	RO	01	ACTPMS: 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 The MCH does not initiate L0s active state Power Management but permits a downstream device from placing the link in L0s
9:4	RO	(port 0,1, 3,5,7) = 04h (port 2,4, 6) = 08h	MLW: Maximum Link Width This field indicates the maximum width of the given PCI Express Link attached to the port. 000001: x1 000100: x4 001000: x8 Others - *Reserved* See Table 4-38. The MCH will update this field based on values in "" on page 169 and the "PXPLWTCTRL: PCI Express Link Width and Training Control Register" on page 166. After link training is initiated, the number of PCI Express ports and their Maximum Link Width is locked by hardware. The value in this field may change before link training.
3:0	RO	0001	MLS: Maximum Link Speed This field indicates the maximum Link speed of the given PCI Express port. 0001: 2.5 Gb/s Others - *Reserved*

Table 4-38. Maximum Link Width Default Value for Different PCI Express Ports

Device/Port	Maximum Link Width	Value
0,1,3,5,7	x4	000100
2,4, 6	x8	001000

4.8.13.7 PEXLNKCTRL[7:0]: PCI Express Link Control Register

The PCI Express Link Control register controls the PCI Express Link specific parameters.

Device: 7-0
Function: 0
Offset: 7Ch

Bit	Attr	Default	Description
15:8	RV	00h	*Reserved.*
7	RW	0	Ext_Synch: Extended Synch This bit when set forces the transmission of 4096 FTS ordered sets in the L0s state followed by a single SKP ordered set prior to entering the L0 state, and the transmission of 1024 TS1 ordered sets in the L1 state prior to entering the Recovery state. This mode provides external devices (e.g., logic analyzers) monitoring the Link time to achieve bit and Symbol lock before the Link enters the L0 or Recovery states and resumes communication.
6	RW	0	CCCON: Common Clock Configuration 0: indicates that this PCI Express port and its counterpart at the opposite end of the Link are operating with an asynchronous reference clock. 1: indicates that this PCI Express port and its counterpart at the opposite end of the Link are operating with a distributed common reference clock. Components utilize this common clock configuration information to report the correct L0s and L1 Exit Latencies.
5	WO	0	RLNK: Retrain Link This bit, when set, initiates link retraining in the given PCI Express port. It always returns 0 when read.
4	RW	0	LNKDIS: Link Disable This field indicates whether the link associated with the PCI Express port is enabled or disabled. 0: Enables the link associated with the PCI Express port 1: Disables the link associated with the PCI Express port NOTE: This field should not be set for ESI port (Device 0) i.e. the ESI port should not be disabled. ***Note:*** It is recommended that software wait for at least 5us from the time it sets the LNKDIS bit to "disable" a link to when it re-enables again.
3	RO	0	RCB: Read Completion Boundary This field defines the read completion boundary for the PCI Express port. Defined encodings for RCB capabilities are: 0: 64 byte 1: 128 byte The MCH supports only 64B read completion boundary and is hardwired to 0.
2	RV	0	***Reserved.***
1:0	RW	00	ACTPMCTRL: Active State Link PM Control This field controls 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 It has no effect on the MCH.

4.8.13.8 PEXLNKSTS[7:0]: 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.

Device: 7-0
Function: 0
Offset: 7Eh

Bit	Attr	Default	Description
15:14	RV	0h	***Reserved.***
13	RO	0	**DLLA: Data Link Layer Active** This field is set by the MCH when the port's data Link Control and Management State Machine changes from/to DL_active state. 0: The port's Link Control and Management State Machine is not in DL_active state 1: The port's Link Control and Management State Machine is in DL_Active state
12	RWO	1	**SCCON: Slot Clock Configuration** This bit indicates that the component uses the same physical reference clock that the platform provides on the connector. If the device uses an independent clock irrespective of the presence of a reference on the connector, this bit must be clear. 1: indicates same physical clock in the PCI Express connector as in the platform 0: indicates independent clock on the PCI Express connector from that of the platform. The MCH initializes this bit to '1' because the expected state of the platform is to have one clock source shared between the MCH component and any down-devices or slot connectors. It is the responsibility of BIOS to be aware of the real platform configuration, and clear this bit if the reference clocks differ.
11	RO	0	LNKTRG: Link Training This field indicates the status of an ongoing link training session in the current PCI Express port and is controlled by the Hardware. 0: indicates that the LTSSM is neither in "Configuration" nor "Recovery" states. 1: indicates Link training in progress (Physical Layer LTSSM is in Configuration or Recovery state or the RLNK (retrain link) was set in Section 4.8.13.7, "PEXLNKCTRL[7:0]: PCI Express Link Control Register" on page 154 but training has not yet begun. Also refer to the BCTRL.SBUSRESET for details on how the Link training bit can be used for sensing Hot-reset states.
10	RO	0	TERR: Training Error This field indicates the occurrence of a Link training error. 0: indicates no Link training error occurred. 1: indicates Link training error occurred. The Link Training machine must make have made it out of detect into either Recovery or Configuration, before this bit can be set to a '1'. This bit is cleared by hardware upon successful training of the Link to the L0 Link state.
9:4	RO	000100	NLNKWD: Negotiated Link Width[a] This field indicates the negotiated width of the given PCI Express link after training is completed. Only x1, x2, x4 and x8 link width negotiations are possible in the MCH . Refer to Table 4-39 for the port and link width assignment after training is completed.
3:0	RO	1h	LNKSPD: Link Speed This field indicates the negotiated Link speed of the given PCI Express Link: 0001- 2.5 Gb/s PCI Express link Others - *Reserved*

Notes:

a. The NLNKWD field is set to a default value corresponding to x4 internally within the MCH. Note that this field is a don't care until training is completed for the link. Software should not use this field to determine whether a link is up (enabled) or not.

Table 4-39. Negotiated Link Width for Different PCI Express Ports after Training

Device/Port	Negotiated Link Width	Value
0,1,2,3,4,5,6,7	x1	000001
0,1,2,3,4,5,6,7	x2	000010
0,1,2,3,4,5,6,7	x4	000100
2,4,6	x8	001000[a]

Notes:

a. Ports 3, 5, and 7 report 000000 as appropriate for the slave ports of x8 connection

4.8.13.9 PEXSLOTCAP[7:0]: PCI Express Slot Capabilities Register

The Slot Capabilities register identifies the PCI Express specific slot capabilities.

Device: 7-0
Function: 0
Offset: 80h

Bit	Attr	Default	Description
31:19	RWO	0h	PSN: Physical Slot Number This field indicates the physical slot number connected to the PCI Express port. It should be initialized to 0 for ports connected to devices that are either integrated on the system board or integrated within the same silicon such as the Root port in the MCH.
18	RO	0	**No Command Completed Support -** This optional bit is not implemented, so it is hardwired.
17	RO	0	**Electromechanical Interlock Present -** This optional bit is not implemented, so it is hardwired.
16:15	RWO	0h	SPLS: Slot Power Limit Scale This field specifies the scale used for the Slot Power Limit Value. Range of Values: 00: 1.0x 01: 0.1x 10: 0.01x 11: 0.001x
14:7	RWO	00h	SPLV: 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
6	RWO	0h	HPC: 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
5	RWO	0h	HPS: Hot-plug Surprise This field indicates that a device in this slot may be removed from the system without prior notification. 0: indicates that hot-plug surprise is not supported 1: indicates that hot-plug surprise is supported

Device: 7-0 Function: 0 Offset: 80h			
Bit	**Attr**	**Default**	**Description**
4	RWO	0h	**PIP: Power Indicator Present** This bit indicates that a Power Indicator is implemented on the chassis for this slot. 0: indicates that Power Indicator is not present 1: indicates that Power Indicator is present
3	RWO	0h	**AIP: Attention Indicator Present** This bit indicates that an Attention Indicator is implemented on the chassis for this slot. 0: indicates that an Attention Indicator is not present 1: indicates that an Attention Indicator is present
2	RWO	0h	**MRLSP: 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
1	RWO	0h	**PCP: Power Controller Present** This bit indicates that a Power Controller is implemented on the chassis for this slot. 0: indicates that a Power Controller is not present 1: indicates that a Power Controller is present
0	RWO	0h	**ABP: Attention Button Present** This bit indicates that an Attention Button is implemented on the chassis for this slot. 0: indicates that an Attention Button is not present 1: indicates that an Attention Button is present

4.8.13.10 PEXSLOTCTRL[7:0]: PCI Express Slot Control Register

The Slot Control register identifies the PCI Express specific slot control specific parameters for operations such as Hot-plug and Power Management. Software issues a command to a hot-plug capable Port by issuing a write transaction that targets Slot Control Register fields viz, PWRCTRL, PWRLED, ATNLED described below. A single write to the Slot Control register is considered to be a single command, even if the write affects more than one field in the Slot Control register. In response to this transaction, the port must carry out the requested actions and then set the associated status field (PEXSLOTS.CMDCMP) for the command completed event. The PEXSLOTSTS.CMDCMP bit will be set only when there is a unique change to the state of the PWRCTRL, PWRLED, ATNLED in this register.

Device: 7-0 Function: 0 Offset: 84h			
Bit	**Attr**	**Default**	**Description**
31:19	RV	0h	***Reserved.***
18	RO	0	**Electromechanical Interlock Control -** This optional bit is not implemented, so it is hardwired.
17:13	RV	0h	***Reserved.***
12	RWST	0h	**DLLSCE:** 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	RV	0h	***Reserved.***

Device: 7-0 Function: 0 Offset: 84h			
Bit	**Attr**	**Default**	**Description**
10	RW	0h	**PWRCTRL:** Power Controller Control This bit indicates the current state of the Power applied to the slot of the PCI Express port. 0: Power On 1: Power Off
9:8	RW	11	**PWRLED:** Power Indicator Control This bit indicates the current state of the Power Indicator of the PCI Express port 00: *Reserved.* 01: On 10: Blink (The MCH drives 1.5 Hz square wave for Chassis mounted LEDs in the case of legacy card form factor for PCI Express devices) 11: Off Default is set to 11b (OFF) When this field is written, the MCH sends appropriate POWER_INDICATOR messages through the PCI Express port. For legacy card based PCI Express devices, the event is signaled via the virtual pins[a] of the MCH, in addition. For PCI Express modules with advanced form factor that incorporate LEDs and onboard decoding logic, the PCI Express messages are interpreted directly (No virtual pins).
7:6	RW	11	**ATNLED:** Attention Indicator Control This bit indicates the current state of the Attention Indicator of the PCI Express port 00: *Reserved.* 01: On 10: Blink (The MCH drives 1.5 Hz square wave) 11: Off Default is set to 11b (OFF) When this field is written, the MCH sends appropriate ATTENTION_INDICATOR messages through the PCI Express port. For legacy card based PCI Express devices, the event is signaled via the virtual pins of the MCH, in addition. For PCI Express modules with advanced form factor that incorporate LEDs and onboard decoding logic, the PCI Express messages are interpreted directly (No virtual pins).
5	RW	0h	**HPINTEN: Hot-plug Interrupt Enable** This field enables the generation of Hot-plug interrupts and events in the PCI Express port. 0: disables Hot-plug events and interrupts 1: enables Hot-plug events and interrupts
4	RW	0h	**CCIEN: Command Completed Interrupt Enable** This field enables the generation of Hot-plug interrupts 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	**PRSINTEN: 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.

Device: 7-0 Function: 0 Offset: 84h			
Bit	**Attr**	**Default**	**Description**
2	RW	0h	**MRLINTEN: 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	**PWRINTEN: 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	**ATNINTEN: 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.

Notes:

a. More information on Virtual pins and Hot plug can be found in Section 7.12.6, "PCI Express Hot-Plug Support, VPP SMBus" .

4.8.13.11 PEXSLOTSTS[7:0]: PCI Express Slot Status Register

The PCI Express Slot Status register defines important status information for operations such as Hot-plug and Power Management.

Device: 7-0 Function: 0 Offset: 86h			
Bit	**Attr**	**Default**	**Description**
31:19	RV	0h	***Reserved.***
18	RO	0	**Electromechanical Interlock Status -** This optional bit is not implemented, so it is hardwired.
17:9	RV	0h	***Reserved.***
8	RWC	0h	**DLLSCS:** Data Link layer State Changed Status 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 has changed. Software must read Data Link Layer Active field to determine the link state before initiating configuration cycles to the hot plugged device.
7	RV	0h	***Reserved.***

Device: 7-0
Function: 0
Offset: 86h

Bit	Attr	Default	Description

6	RO	1	**PDS: Presence Detect State**
			This field conveys the Presence Detect status determined via an in-band mechanism or through the Present Detect pins and shows the presence of a card in the slot.
			0: Slot Empty
			1: Card Present in slot
			This field is set to 1 by the MCH following reset for all the ports since it is related to the complement of the PEXCAP.SLOT bit whose default is 0 implying that the card is connected to the MCH on the motherboard. Software should ensure that for those standard PCI Express ports which are hot-plug capable, it sets the PEXCAP.SLOT field to '1' such that PDS becomes 0. Then any subsequent hotplug action will ensure that this field is set.
			
			The PDS can also be set by inband presence detect LTSSM feature in the MCH which is controlled through PEXCTRL5.DISINBPDS register field.
			
			Refer to the algorithm given below:
			if (PEXCAP.SLOT == FALSE) {
			// always set to 1b and ignore the input from inband/OOB presence detect
			PEXSLOTSTS.PDS = 1b
			}
			else {//
			if (PEXCAP.SLOT == TRUE) {
			// Slot Empty = 0, Card Present = 1
			PEXSLOTSTS.PDS reflects slot status of side-band SMBUS Presence Detect signal OR set through the inband presence from LTSSM
			}

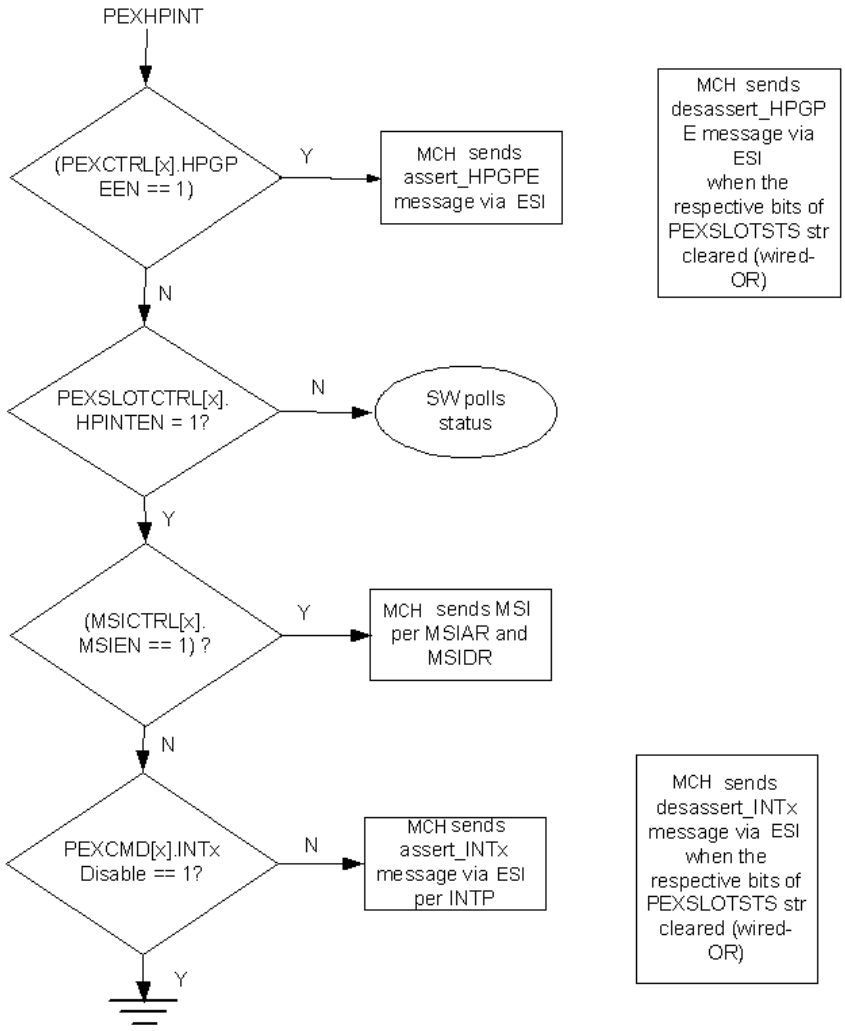
5	RO	1h	**MRLSS: MRL Sensor State** This bit reports the status of an MRL sensor if it is implemented. 0: MRL Closed 1: MRL Open
4	RWC	0h	**CMDCOMP: Command Completed** This bit is set by the MCH when the hot-plug controller completes an issued command and is ready to accept a new command. It is subsequently cleared by software after the field has been read and processed.
3	RWC	0h	**PRSINT: Presence Detect Changed** This bit is set by the MCH when a Presence Detect Changed event is detected. It is subsequently cleared by software after the field has been read and processed.
2	RWC	0h	**MRLSC: MRL Sensor Changed** This bit is set by the MCH when an MRL Sensor Changed event is detected. It is subsequently cleared by software after the field has been read and processed.
1	RWC	0h	**PWRINT: Power Fault Detected** This bit is set by the MCH 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.
0	RWC	0h	**ABP: Attention Button Pressed** This bit is set by the MCH 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 to this bit inactive if the FF/system does not support out-of-band presence detect. ***Note:*** The MCH design implements this as an edge detection logic in the chipset/platform to recognize a 0 -> 1 transition as a ABP event. Due to the long time frames for human interaction, software could potentially have serviced and cleared this event while the operator is pressing the attention button. Hardware must only generate 1 interrupt for every ABP event.

Note that the Assert_intx/Assert_HPGPE message is sent to ESI port when any of the events defined in bits[4:0] (CMDCOMP,PRSINT, MRLSC, PWRINT, ABP) of the PEXSLOTSTS register are set provided the corresponding events in bits [4:0] of the

Section 4.8.13.10, “PEXSLOTCTRL[7:0]: PCI Express Slot Control Register” on page 157 and HPINTEN are enabled. Software writes to clear these bits and the MCH will send a Deassert_HPGPE message to ESI port (wired-OR).

For the case when MSI is enabled, any new event that sets these bits (e.g ABP, PRSINT etc.) will cause an MSI message to be sent to the FSB for each occurrence. i.e. each bit is considered unique. In the case of Legacy interrupts, a wired-OR approach is used to mimic the level sensitive behavior and only one assert_intx/assert_GPE (deassert_intx/deassert_GPE) is sent even when multiple interrupt generating bits of the register get set. Refer to Figure 4-5, “PCI Express Hot Plug Interrupt Flow” on page 162.

Figure 4-5. PCI Express Hot Plug Interrupt Flow



HPGPEEN	HPINTEN	MSIEN	INTx Disable	Output
1	x	x	x	assert_hpgpe
0	1	1	x	MSI
0	1	0	0	assert_intx
0	1	0	1	--
0	0	x	x	--

4.8.13.12 PEXRTCTRL[7:0]: PCI Express Root Control Register

The PCI Express Root Control register specifies parameters specific to the root complex port.

Device: 7-0
Function: 0
Offset: 88h

Bit	Attr	Default	Description
15:4	RV	0h	*Reserved.*
3	RW	0h	**PMEINTEN: PME Interrupt Enable** This field controls the generation of interrupts for PME messages. 1: Enables interrupt generation upon receipt of a PME message as reflected in the PME Status bit defined in the PEXRTSTS register. A PME interrupt is generated if the PMESTATUS register bit defined in Section 4.8.13.13, "PEXRTSTS[7:0]: PCI Express Root Status Register" on page 164, is set when this bit is set from a cleared state. 0: Disables interrupt generation for PME messages.
2	RW	0h	**SEFEEN: System Error on Fatal Error Enable** This field controls generation of system errors in the PCI Express port hierarchy for fatal errors. 1: indicates that a System Error should be generated if a fatal error (ERR_FATAL) is reported by any of the devices in the hierarchy associated with and including this PCI Express port. 0: No System Error should be generated on a fatal error (ERR_FATAL) reported by any of the devices in the hierarchy.
1	RW	0h	**SENFEEN: System Error on Non-Fatal Error Enable** This field controls generation of system errors in the PCI Express port hierarchy for non-fatal errors. 1: indicates that a System Error should be generated if a non-fatal error (ERR_NONFATAL) is reported by any of the devices in the hierarchy associated with and including this PCI Express port. 0: No System Error should be generated on a non-fatal error (ERR_NONFATAL) reported by any of the devices in the hierarchy.
0	RW	0h	**SECEEN: System Error on Correctable Error Enable** This field controls generation of system errors in the PCI Express port hierarchy for correctable errors. 1: indicates that a System Error should be generated if a correctable error (ERR_COR) is reported by any of the devices in the hierarchy associated with and including this PCI Express port 0: No System Error should be generated on a correctable error (ERR_COR) reported by any of the devices in the hierarchy associated with and including this PCI Express port.

4.8.13.13 PEXRTSTS[7:0]: PCI Express Root Status Register

The PCI Express Root Status register specifies parameters specific to the root complex port.

Device: 7-0
Function: 0
Offset: 8Ch

Bit	Attr	Default	Description
31:18	RV	0h	*Reserved.*
17	RO	0h	**PMEPEND: PME Pending** This field indicates that another PME is pending when the PME Status bit is set. When the PME Status bit is cleared by software; the pending PME is delivered by hardware by setting the PME Status bit again and updating the Requestor ID appropriately. The PME pending bit is cleared by hardware if no more PMEs are pending. ***Note:*** The MCH can handle two outstanding PM_PME messages in its internal queues of the Power Management controller per port. If the downstream device issues more than 2 PM_PME messages successively, it will be dropped.
16	RWC	0h	**PMESTATUS: PME Status**[a] This field indicates status of a PME that is underway in the PCI Express port. 1: PME was asserted by a requester as indicated by the PMEREQID field This bit is cleared by software by writing a '1'. Subsequent PMEs are kept pending until the PME Status is cleared.
15:0	RO	0000h	**PMEREQID: PME Requester ID** This field indicates the PCI requester ID of the last PME requestor.

Notes:

a. PMEINTEN defined in PEXRTCTRL has to be set for PM interrupts to be generated.

For non-MSI PM interrupts, the PMESTATUS bit in each of the PEXRTSTS[7:1] registers are wired OR together and when set, the MCH will send the "Assert_PMEGPE" message to the Intel® 631xESB/632xESB I/O Controller Hub for power management. When all the bits are clear, it will send the "Deassert_PMEGPE" message. PMEINTEN defined in PEXRTCTRL has to be set for PM interrupts to be generated.

PM_PME events that generate MSI will depend on the MSIEN field in Section 4.8.12.3, "MSICTRL[7:0]: Message Control Register" on page 143. Refer to the PM interrupt flow in Power Management Chapter.

4.8.13.14 ESICTRL[0]: ESI Control Register

The ESICTRL register holds control information and defeature bits pertaining to the ESI interface for power management.

Device: 0
Function: 0
Offset: D4h

Bit	Attr	Default	Description
31:25	RV	0h	*Reserved.*
24	RW	0	**DISESIRST: Disable ESI Reset Sequence** 0: The ESI port will perform the normal ESI reset sequence by constraining the Link Detect state machines and other polling states to match the ESI requirements. (default) 1: The ESI port will skip the shortened ESI reset sequence and revert to the PCI Express reset/training specification. This mode will be used for debug. Refer to the Intel ESI Specification
23:17	RV	0h	*Reserved.*

Device: 0
Function: 0
Offset: D4h

Bit	Attr	Default	Description
16	RW	0	**DCFD: Defeature CE Flush Done** 0: Normal operation. 1: The MCH's CE has flushed all transactions and is done. The PM logic can continue its operation under the assumption that CE has flushed all its transactions when this bit is set. ***Note:*** For Sx Power management in H/W or S/W mode
15	RW	0	**DMQE: Defeature MC Queue Empty** 0: Normal operation. 1: The MCH's MC queue are flushed (i.e. empty). The PM logic can continue its operation under the assumption that MC Queue is empty when this bit is set. ***Note:*** For Sx Power management in H/W or S/W mode
14	RW	0	**DL23R: Defeature L23 Ready** 0: Normal operation. 1: The MCH has entered L23 state (electrical idle, low power state) on all the standard PCI Express[7:1] ports. The PM logic can continue its operation under the assumption that all the PCI Express ports have entered the L23 state. ***Note:*** For Sx Power management in H/W or S/W mode
13	RW	0	**DPTOA: Defeature PME_TO_Ack** 0: Normal operation. 1: The MCH has received "PME_TO_Ack" message from all the standard PEX port[7:1]. The PM logic can continue its operation under the assumption that all PCI Express ports have received the "PME_TO_ACK" message when this bit is set. ***Note:*** For Sx Power management in H/W mode only
12	RW	0	**DSRA: Defeature Self Refresh Acknowledge** 0: Normal operation. 1: The MC has issued self refresh to Gold Bridge. The PM logic can continue its operation under the assumption that the MC has issued a self-refresh to GB when this bit is set. ***Note:*** For Sx Power management in H/W or S/W mode
11	RWC	0	**PTE: PME_TO_Ack Time Expired** 0: Default mode where the MCH hardware broadcasts PME_turn_off message to all enabled PCI Express ports 1: Signal that time expiration has occurred when the PTOV field described below crosses the threshold in the MCH . ***Note:*** For Sx Power management in H/W mode only
10:9	RW	0h	**PTOV: PME_TO_Ack Time Out Value** 00: 1 ms (default) 01: 10 ms 10: 50 ms 11: test mode This register field provides the timer limit for the MCH to keep track of the elapsed time from sending "PME_Turn_off" to receiving a "PME_TO_Ack". The test mode bypasses the timer i.e no timer is invoked for the "PME_Turn_off" message. ***Note:*** For Sx Power management in H/W mode only

Device: 0 Function: 0 Offset: D4h			
Bit	**Attr**	**Default**	**Description**
8	RW	1	**S3HM: S3 Hardware Mode** 0: Software will control the broadcast of PME_Turn_Off before entry to S3 through the PME_TURN_OFF register field in Section 4.8.10.33, "PEXGCTRL: PCI Express Global Control Register" on page 139. 1: Hardware mode where the MCH hardware broadcasts PME_turn_off message to all enabled PCI Express ports (excluding the ESI). The MCH will initially set the PEXGCTRL.PME_TURN_OFF bit and then clears it when all the PME_TURN_OFFs have been sent to the PCI Express ports. (default) ***Note:*** For Sx Power management in H/W mode only ***Note:*** The MCH does not stay in the S3 state. It changes to S3 before transitioning into deeper Sleep states.
7:6	RV	0h	*Reserved.*
5:0	RW	0h	**SAC: STOPGRANT ACK COUNT** This field tracks the number of Stop Grant acks received from the FSBs. THe MCH will forward the last StopGrantAck received from the FSB to the Intel® 631xESB/632xESB I/O Controller Hub using the "Req_C2" command. Software is expected to set this field to "THREADs-1" where the variable "THREAD" is the total number of logical threads present in the system (currently can handle up to 64). Typically each CPU thread will issue a StopGrantAck in response to a STPCLK# assertion from the Intel® 631xESB/632xESB I/O Controller Hub. When the final StopGrantAck is received from the FSB and the internal counter hits the value of SAC+1 (which is equal to THREAD), the MCH will initiate the "Req_C2" command on the ESI. It is illegal for the CPU to send more Stop Grant Acks than that specified in the "THREAD" variable. ***Note:*** For Sx Power management in H/W or S/W mode

4.8.13.15 PXPLWTCTRL: PCI Express Link Width and Training Control Register

This register provides the ability for software to configure the width of PCI Express ports 1-7. Intel® 7300 Chipset does not have any external hardware strapping pins to indicate how the PCI Express ports should be configured in the system. This register, along with the DEVPRES register in Section 4.8.13.16, provides the primary method for BIOS to enforce a specific setting for a given platform. BIOS must ensure that both PXPLWTCTRL and DEVPRES registers are programmed to reflect identical settings for all ports.

The following sequence is required by BIOS to determine the number of ports and the maximum port widths for PCI Express ports 1-7:

1. BIOS must first determine the active PCI Express ports by programming the DEVPRES register. Table 4-40. PCI Express Port and Width Selection shows how the DEVPRES register values determine the port width. BIOS must also program PXPLWTCTRL register fields accordingly, to match exactly the information contained in the DEVPRES register. PCI Express port devices that are "not present" will not be shutdown.
2. If ports 2, 4, and 6 are active, BIOS will need to ensure that the port width register fields are correctly programmed in the PXPLWTCTRL register before initiating link training.
3. BIOS will initiate link training by setting PXPLWTCTRL.ILNKTRN1 and PXPLWTCTRL.ILNKTRN0.

Device: 0 Function: 0 Offset: E0h			
Bit	**Attr**	**Default**	**Description**
31:28	RV	0h	***Reserved***.
27:26	RW	01	**PXPWID6: Port 6 Maximum Width**[2] This field is used to determine the port width of port 6 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 6 is "not present", this field has no effect. Refer to Table 4-40 to determine when HW will use this field to determine the width of port 6. After link training has been initiated by the ILNKTRN1 field in this register, any update of this field has no effect. 00: x4 maximum port width 01: x8 maximum port width Others: Reserved
25:24	RW	01	**PXPWID4: Port 4 Maximum Width**[1] This field is used to determine the port width of port 4 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 4 is "not present", this field has no effect. Refer to Table 4-40 to determine when HW will use this field to determine the width of port 4. After link training has been initiated by the ILNKTRN1 field in this register, any update of this field has no effect. 00: x4 maximum port width 01: x8 maximum port width Others: Reserved
23:17	RV	0	***Reserved***.
16	RW	0	**ILNKTRN1: Initiate Link Training IOU1** This bit controls PCI Express port link training for IOU1 ports. A value of '1' initiates link training on ports 4, 5, 6 and 7 if present as determined by "DEVPRES: Device Present Control Register". IOU1 ports will not automatically initiate link training after reset. A write of '0' has no effect. A write of '1' will lock this register down and initiate link training. 0: The IOU1 PCI Express ports have not initiated link training. 1: The IOU1 PCI Express ports are initiating link training or have trained.
15:12	RV	0	***Reserved.***
11:10	RW	01	**PXPWID2: Port 2 Maximum Width**[2] This field is used to determine the port width of port 2 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 2 is "not present", this field has no effect. Refer to Table 4-40 to determine when HW will use this field to determine the width of port 2. After link training has been initiated by the ILNKTRN0 field in this register, any update of this field has no effect. 00: x4 maximum port width 01: x8 maximum port width Others: Reserved
9:8	RO	00	**PXPWID1: Port 1 Maximum Width** This field is hardwired to "00" to indicate that port 1 can be only connected to a x4 device.
7:1	RV	0h	***Reserved***.
0	RW	0h	**ILNKTRN0: Initiate Link Training IOU0** This bit controls PCI Express port link training for IOU0 ports. A value of '1' initiates link training on ports 1, 2 and 3 if present as determined by "DEVPRES: Device Present Control Register". IOU0 ports will not automatically initiate link training after reset. A write of '0' has no effect. A write of '1' will lock this register down and initiate link training. 0: The IOU0 PCI Express ports have not initiated link training. 1: The IOU0 PCI Express ports are initiating link training or have trained.

Notes:

1. The PXPLWTCTRL.PXPWID4 field determines the port's maximum capable width. Software/BIOS must make sure that this field is not programmed to any Reserved value. The only supported values are 00b and 01b for x4 and x8 maximum port widths, respectively.
2. The PXPLWTCTRL.PXPWID2 field (or PXPLWTCTRL.PXPWID6 field) determines the port's maximum capable width. Software/BIOS must make sure that this field is not programmed to any Reserved value. The only supported values are 00b and 01b for x4 and x8 maximum port widths, respectively.

4.8.13.16 DEVPRES: Device Present Control Register

This register allows BIOS to inform hardware which of the following devices in Intel® 7300 Chipset are "Not Present". A value of '0' indicates a device is not present.

BIOS should program the value of DEVPRES to match the desired Port configuration specific to each platform. In some instances involving slot Lane reversal where the platform Max Link Width Capability is smaller than the MCH capability, width information for Ports 2, 4 or 6 may have to be programmed. BIOS will then initiate PCIe Training after the platform Port configuration and Max Link Width Cap information is programmed. Refer to Section 4.8.13.15, "PXPLWTCTRL: PCI Express Link Width and Training Control Register" on page 166 and Table 4-40, "PCI Express Port and Width Selection" on page 169 for more details.

For Devices 1-7, setting of a previously cleared DEVPRES bit is not allowed after PCIe Training is initiated. After link training, SW may only remove devices by clearing the associated bit, but any additional functionality that would be normally be made available to other ports will not be restored.

Device: 0
Function: 0
Offset: E4h

Bit	Attr	Default	Description
31:8	RV	00h	***Reserved***.
7	RW	1	**PCIE Port 7 Device**
6	RW	1	**PCIE Port 6 Device**
5	RW	1	**PCIE Port 5Device**
4	RW	1	**PCIE Port 4 Device**
3	RW	1	**PCIE Port 3 Device**
2	RW	1	**PCIE Port 2 Device**
1	RW	1	**PCIE Port 1 Device**
0	RO	1	**ESI Port Device**

Table 4-40. PCI Express Port and Width Selection

Port 7	Port 6	DEVPRES[7:6]	PXPLWTCTRL.PXPWID6
Not Present	Not Present	00b	00b
Not Present	x4	01b	00b
Not Present	x8	01b	01b
x4	Not Present	10b	00b
x4	x4	11b	00b
Port 5	**Port 4**	**DEVPRES[5:4]**	**PXPLWTCTRL.PXPWID4**
Not Present	Not Present	00b	00b
Not Present	x4	01b	00b
Not Present	x8	01b	01b
x4	Not Present	10b	00b
x4	x4	11b	00b
Port 3	**Port 2**	**DEVPRES[3:2]**	**PXPLWTCTRL.PXPWID2**
Not Present	Not Present	00b	00b
Not Present	x4	01b	00b
Not Present	x8	01b	01b
x4	Not Present	10b	00b
x4	x4	11b	00b
Port 1	**Port 0**	**DEVPRES[1:0]**	
Not Present	x4	01b	
x4	x4	11b	

Notes:
1. Only combinations listed in the table above are supported configurations.
2. Port 0 (ESI Port) cannot be disabled.

4.8.13.17 SVID_CAPID: Capability ID Register for SVID/SID Capability

Device: 1-7
Function: 0
Offset: F0h

Bit	Attr	Default	Description
7:0	RO	0Dh	Subsystem Vendor Identification Number

4.8.13.18 SVID_NXTP: Next Capability Pointer Register for SVID/SID Capability

Device: 1-7
Function: 0
Offset: F1h

Bit	Attr	Default	Description
7:0	RO	0000h	This is the last capability in the MCH capability chain

4.8.13.19 SVID: Subsystem Vendor ID Register

Device: 1-7
Function: 0
Offset: F4h

Bit	Attr	Default	Description
15:0	RWO	8086	**Subsystem Vendor Identification Number:** The default value specifies Intel. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.8.13.20 SID: Subsystem Device ID Register

Device: 1-7
Function: 0
Offset: F6h

Bit	Attr	Default	Description
15:0	RWO	0000h	**Subsystem Device Identification Number:** Uniquely identifies each device in the subsystem from the subsystem vendor. Each byte of this register will be writable once. Second and successive writes to a byte will have no effect.

4.8.14 PCI Express Advanced Error Reporting Capability

4.8.14.1 PEXENHCAP[7:0]: PCI Express Enhanced Capability Header

This register identifies the capability structure and points to the next structure.

Device: 7-0
Function: 0
Offset: 100h

Bit	Attr	Default	Description
31:20	RO	0h	**NCAPOFF: Next Capability Offset** This field points to the next Capability in extended configuration space. This field is initialized to 0h to terminate the extended capability.
19:16	RO	1h	**CV: Capability Version** Set to 1h for this version of the PCI Express logic
15:0	RO	0001h	**PEXCAPID: PCI Express Extended CAP_ID** Assigned for advanced error reporting

4.8.14.2 UNCERRSTS[7:1]: Uncorrectable Error Status

This register identifies uncorrectable errors detected. for PCI Express Port. If an error occurs and is unmasked in the detect register (EMASK_UNCOR_PEX), the appropriate error bit will be recorded in this register. If an error is recorded in the UNCERRSTS register and the appropriate bit (along with the severity bit of the UNCERRSEV register) determines which bit in the PEX_FAT_FERR, PEX_NF_COR_FERR, PEX_FAT_NERR, PEX_NF_COR_NERR registers gets recorded. These error log registers are described starting from Section 4.8.14.24, "PEX_FAT_FERR[7:0]: PCI Express First Fatal Error Register" on page 183.

Device: 7-1
Function: 0
Offset: 104h

Bit	Attr	Default	Description
31:21	RV	0h	Reserved
20	RWCST	0	IO2Err: Received an Unsupported Request
19	RV	0	Reserved
18	RWCST	0	IO9Err: Malformed TLP Status
17	RWCST	0	IO10Err: Receiver Buffer Overflow Status
16	RWCST	0	IO8Err: Unexpected Completion Status
15	RWCST	0	IO7Err: Completer Abort Status
14	RWCST	0	IO6Err: Completion Time-out Status
13	RWCST	0	IO5Err: Flow Control Protocol Error Status
12	RWCST	0	IO4Err: Poisoned TLP Status
11:6	RV	0h	Reserved
5	RWCST	0h	IO19Err: Surprise Link Down Error Status
4	RWCST	0	IO0Err: Data Link Protocol Error Status
3:1	RV	0h	Reserved

Device: 7-1 Function: 0 Offset: 104h			
Bit	**Attr**	**Default**	**Description**
0	RWCST	0	IO3Err:Training Error Status ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error. ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.

4.8.14.3 UNCERRSTS[0]: Uncorrectable Error Status

This register identifies uncorrectable errors detected on ESI Port. If an error occurs and is unmasked in the detect register (EMASK_UNCOR_PEX), the appropriate error bit will be recorded in this register. If an error is recorded in the UNCERRSTS register and the appropriate bit (along with the severity bit of the UNCERRSEV register) determines which bit in the PEX_FAT_FERR, PEX_NF_COR_FERR, PEX_FAT_NERR, PEX_NF_COR_NERR registers gets recorded. These error log registers are described starting from Section 4.8.14.24, "PEX_FAT_FERR[7:0]: PCI Express First Fatal Error Register" on page 183.

Device: 0 Function: 0 Offset: 104h			
Bit	**Attr**	**Default**	**Description**
31:22	RV	0h	Reserved
21	RWCST	0	IO18Err: ESI Reset timeout
20	RWCST	0	IO2Err: Received an Unsupported Request
19	RV	0	Reserved
18	RWCST	0	IO9Err: Malformed TLP Status
17	RWCST	0	IO10Err: Receiver Buffer Overflow Status
16	RWCST	0	IO8Err: Unexpected Completion Status
15	RWCST	0	IO7Err: Completer Abort Status
14	RWCST	0	IO6Err: Completion Time-out Status
13	RWCST	0	IO5Err: Flow Control Protocol Error Status
12	RWCST	0	IO4Err: Poisoned TLP Status
11:6	RV	0h	Reserved
5	RWCST	0h	IO19Err: Surprise Link Down Error Status
4	RWCST	0	IO0Err: Data Link Protocol Error Status
3:1	RV	0h	Reserved
0	RWCST	0	IO3Err:Training Error Status ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error. ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.

4.8.14.4 UNCERRMSK[7:1]: Uncorrectable Error Mask

This register masks uncorrectable errors from the UNCERRSTS[7:1] register from being signaled. For a description of the I/O Errors (IOxx) refer to Table 7-37, "Intel® 7300 Chipset Chipset Error List".

Device: 7-1
Function: 0
Offset: 108h

Bit	Attr	Default	Description
31:21	RV	0h	Reserved
20	RWST	0	IO2Msk: Received an Unsupported Request
19	RV	0	Reserved
18	RWST	0	IO9Msk: Malformed TLP Status
17	RWST	0	IO10Msk: Receiver Buffer Overflow Mask
16	RWST	0	IO8Msk: Unexpected Completion Mask
15	RWST	0	IO7Msk: Completer Abort Status
14	RWST	0	IO6Msk: Completion Time-out Mask
13	RWST	0	IO5Msk: Flow Control Protocol Error Mask
12	RWST	0	IO4Msk: Poisoned TLP Mask
11:6	RV	0h	Reserved
5	RWST	0h	IO19Msk: Surprise Link Down Error Mask
4	RWST	0	IO0Msk: Data Link Layer Protocol Error Mask
3:1	RV	000	Reserved
0	RWST	0	IO3Msk:Training Error Mask ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error. ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.

4.8.14.5 UNCERRMSK[0]: Uncorrectable Error Mask

This register masks uncorrectable errors from the UNCERRSTS[0] register (ESI port) from being signaled. For a description of the I/O Errors (IOxx) refer to Table 7-37, "Intel® 7300 Chipset Chipset Error List".

Device: 0
Function: 0
Offset: 108h

Bit	Attr	Default	Description
31:22	RV	0h	Reserved
21	RWST	0	IO18Msk: ESI Reset timeout
20	RWST	0	IO2Msk: Received an Unsupported Request
19	RV	0	Reserved
18	RWST	0	IO9Msk: Malformed TLP Status
17	RWST	0	IO10Msk: Receiver Buffer Overflow Mask
16	RWST	0	IO8Msk: Unexpected Completion Mask
15	RWST	0	IO7Msk: Completer Abort Status
14	RWST	0	IO6Msk: Completion Time-out Mask

Device: 0 Function: 0 Offset: 108h			
Bit	Attr	Default	Description
13	RWST	0	IO5Msk: Flow Control Protocol Error Mask
12	RWST	0	IO4Msk: Poisoned TLP Mask
11:6	RV	0h	Reserved
5	RWST	0h	IO19Msk: Surprise Link Down Error Mask
4	RWST	0	IO0Msk: Data Link Layer Protocol Error Mask
3:1	RV	000	Reserved
0	RWST	0	IO3Msk:Training Error Mask ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error. ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.

4.8.14.6 UNCERRSEV[0]: Uncorrectable Error Severity

This register indicates the severity of the uncorrectable errors for the ESI port. An error is reported as fatal when the corresponding error bit in the severity register is set. If the bit is cleared, the corresponding error is considered non-fatal. If an error is recorded in the UNCERRSTS register, the appropriate bit of UNCERRSEV determines if the error gets reflected as a device fatal or nonfatal error in the PEX_FAT_FERR, PEX_NF_COR_FERR, PEX_FAT_NERR, PEX_NF_COR_NERR registers starting from Section 4.8.14.24, "PEX_FAT_FERR[7:0]: PCI Express First Fatal Error Register" on page 183. For a description of the I/O Errors, refer to Table 7-37, "Intel® 7300 Chipset Chipset Error List".

Device: 0 Function: 0 Offset: 10Ch			
Bit	Attr	Default	Description
31:22	RV	0h	Reserved
21	RWST	1	IO18Severity: ESI Reset timeout
20	RWST	0	IO2Severity: Received an Unsupported Request
19	RV	0	Reserved
18	RWST	1	IO9Severity: Malformed TLP Severity
17	RWST	1	IO10Severity: Receiver Buffer Overflow Severity
16	RWST	0	IO8Severity: Unexpected Completion Severity
15	RWST	0	IO7Severity: Completer Abort Status
14	RWST	0	IO6Severity: Completion Time-out Severity
13	RWST	1	IO5Severity: Flow Control Protocol Error Severity
12	RWST	0	IO4Severity: Poisoned TLP Severity
11:6	RV	0h	Reserved
5	RWST	1	IO19Severity: Surprise Link Down Severity
4	RWST	1	IO0Severity: Data Link Protocol Error Severity (See Figure 3-17 in *PCI Express Base Specification*, Revision 1.0a)
3:1	RV	000	Reserved
0	RWST	1	IO3Severity:Training Error Severity

4.8.14.7 UNCERRSEV[7:1]: Uncorrectable Error Severity

This register indicates the severity of the uncorrectable errors. An error is reported as fatal when the corresponding error bit in the severity register is set. If the bit is cleared, the corresponding error is considered non-fatal. If an error is recorded in the UNCERRSTS register, the appropriate bit of UNCERRSEV determines if the error gets reflected as a device fatal or nonfatal error in the PEX_FAT_FERR, PEX_NF_COR_FERR, PEX_FAT_NERR, PEX_NF_COR_NERR registers starting from Section 4.8.14.24, "PEX_FAT_FERR[7:0]: PCI Express First Fatal Error Register" on page 183. For a description of the I/O Errors, refer to Table 7-37, "Intel® 7300 Chipset Chipset Error List".

Device: 7-1
Function: 0
Offset: 10Ch

Bit	Attr	Default	Description
31:21	RV	0h	Reserved
20	RWST	0	IO2Severity: Received an Unsupported Request
19	RV	0	Reserved
18	RWST	1	IO9Severity: Malformed TLP Severity
17	RWST	1	IO10Severity: Receiver Buffer Overflow Severity
16	RWST	0	IO8Severity: Unexpected Completion Severity
15	RWST	0	IO7Severity: Completer Abort Status
14	RWST	0	IO6Severity: Completion Time-out Severity
13	RWST	1	IO5Severity: Flow Control Protocol Error Severity
12	RWST	0	IO4Severity: Poisoned TLP Severity
11:6	RV	0h	Reserved
5	RWST	1	IO19Severity: Surprise Link Down Severity
4	RWST	1	IO0Severity: Data Link Protocol Error Severity (See Figure 3-17 in *PCI Express Base Specification*, Revision 1.0a)
3:1	RV	000	Reserved
0	RWST	1	IO3Severity:Training Error Severity ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error. ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.

4.8.14.8 CORERRSTS[7:0]: Correctable Error Status

This register identifies which unmasked correctable error has been detected and they feed the respective device correctable error bit in the PEX_NF_COR_FERR, PEX_NF_COR_NERR registers starting from Section 4.8.14.25, "PEX_NF_COR_FERR[7:0]: PCI Express First Non-Fatal or Correctable Error Register" on page 184 (if the error is unmasked in the CORERRMSK register defined in Section 4.8.14.9, "CORERRMSK[7:0]: Correctable Error Mask" on page 176).

Device: 7-0
Function: 0
Offset: 110h

Bit	Attr	Default	Description
31:14	RV	0h	Reserved
13	RWCST	0	Advisory Non-fatal Error Status
12	RWCST	0	IO16Err: Replay Timer Time-out Status
11:9	RV	0h	Reserved
8	RWCST	0	IO15Err: Replay_Num Rollover Status
7	RWCST	0	IO14Err: Bad DLLP Status
6	RWCST	0	IO13Err: Bad TLP Status
5:1	RV	0h	Reserved
0	RWCST	0	IO12Err: Receiver Error Status

4.8.14.9 CORERRMSK[7:0]: Correctable Error Mask

This register masks correctable errors from being not signalled. They are still logged in the CORERRSTS register.

Device: 7-0
Function: 0
Offset: 114h

Bit	Attr	Default	Description
31:14	RV	0h	Reserved
13	RWST	0	Advisory Non-fatal Error Mask
12	RWST	0	IO16Msk: Replay Timer Time-out Mask
11:9	RV	0h	Reserved
8	RWST	0	IO15Msk: Replay_Num Rollover Mask
7	RWST	0	IO14Msk: Bad DLLP Mask
6	RWST	0	IO13Msk: Bad TLP Mask
5:1	RV	0h	Reserved
0	RWST	0	IO12Msk: Receiver Error Mask

4.8.14.10 AERRCAPCTRL[7:0]: Advanced Error Capabilities and Control Register

This register identifies the capability structure and points to the next structure.

Device: 7-0
Function: 0
Offset: 118h

Bit	Attr	Default	Description
31:9	RV	0h	Reserved
8	RO	0	ECRCCHKEN: ECRC Check Enable This bit when set enables ECRC checking.
7	RO	0	ECRCCHKCAP: ECRC Check Capable The MCH does not support ECRC.
6	RO	0	ECRCGENEN: ECRC Generation Enable The MCH does not generate ECRC.
5	RO	0	ECRCGENCAP: ECRC Generation Capable The MCH does not generate ECRC.
4:0	ROST	0h	FERRPTR: First error pointer The First Error Pointer is a read-only register that identifies the bit position of the first error reported in the Uncorrectable Error status register. Left most error bit if multiple bits occurred simultaneously.

4.8.14.11 HDRLOG0[7:0]: Header Log 0

This register contains the first 32 bits of the header log locked down when the first uncorrectable error occurs. Headers of the subsequent errors are not logged.

Device: 7-0
Function: 0
Offset: 11Ch

Bit	Attr	Default	Description
31:0	ROST	0h	HDRLOGDW0: Header of TLP (DWORD 0) associated with first uncorrectable error

4.8.14.12 HDRLOG1[7:0]: Header Log 1

This register contains the second 32 bits of the header log.

Device: 7-0
Function: 0
Offset: 120h

Bit	Attr	Default	Description
31:0	ROST	0h	HDRLOGDW1: Header of TLP (DWORD 1) associated with uncorrectable error

4.8.14.13 HDRLOG2[7:0]: Header Log 2

This register contains the third 32 bits of the header log.

Device: 7-0
Function: 0
Offset: 124h

Bit	Attr	Default	Description
31:0	ROST	0h	HDRLOGDW2: Header of TLP (DWORD 2) associated with uncorrectable error

4.8.14.14 HDRLOG3[7:0]: Header Log 3

This register contains the fourth 32 bits of the header log.

Device: 7-0
Function: 0
Offset: 128h

Bit	Attr	Default	Description
31:0	ROST	0h	HDRLOGDW3: Header of TLP (DWORD 3) associated with uncorrectable error

4.8.14.15 RPERRCMD[7:0]: Root Port Error Command

This register controls behavior upon detection of errors. Refer to the RAS Chapter for details on the Root port error interrupt handling.

Device: 7-0
Function: 0
Offset: 12Ch

Bit	Attr	Default	Description
31:8	RV	0h	Reserved
7:3	RV	0h	Reserved
2	RW	0	**EN_FAT_ERR: FATAL Error Reporting Enable** Enable interrupt on fatal errors when set.
1	RW	0	**EN_NONFAT_ERR: Non-FATAL Error Reporting Enable** Enable interrupt on a non-fatal (uncorrectable) error when set
0	RW	0	**EN_CORR_ERR: Correctable Error Reporting Enable** Enable interrupt on correctable errors when set

4.8.14.16 RPERRSTS[7:0]: Root Error Status Register

The Root Error Status register reports status of error Messages (ERR_COR, ERR_NONFATAL, and ERR_FATAL) received by the Root Complex in Intel® 7300 Chipset, and errors detected by the Root Port itself (which are treated conceptually as if the Root Port had sent an error Message to itself). The ERR_NONFATAL and ERR_FATAL Messages are grouped together as uncorrectable. Each correctable and uncorrectable (Non-fatal and Fatal) error source has a first error bit and a next error bit associated with it respectively. When an error is received by a Root Complex, the respective first error bit is set and the Requestor ID is logged in the Error Source Identification register. A set individual error status bit indicates that a particular error category occurred; software may clear an error status by writing a 1 to the respective bit. If software does not clear the first reported error before another error Message is received of the same category (correctable or uncorrectable), the corresponding next error status bit will be set but the Requestor ID of the subsequent error Message is discarded. The next error status bits may be cleared by software by writing a 1 to the respective bit as well. This register is updated regardless of the settings of the Root Control register in Section 4.8.13.12, "PEXRTCTRL[7:0]: PCI Express Root Control Register" on page 163 and the Root Error Command register defined in Section 4.8.14.15, "RPERRCMD[7:0]: Root Port Error Command" on page 178. Refer to the PCI_Express error flow in the RAS Chapter.

Device: 7-0 Function: 0 Offset: 130h			
Bit	**Attr**	**Default**	**Description**
31:27	RO	0h	ADVERR_INT_MSG_NUM: Advanced Error Interrupt Message Number Advanced Error Interrupt Message Number offset between base message data and the MSI message if assigned more than one message number to be used of any status in this capability.
26:7	RV	0h	Reserved
6	RWCST	0	FAT_ERR_Rcvd: Fatal Error Messages Received Set when one or more Fatal Uncorrectable error Messages[a] have been received.
5	RWCST	0	NFAT_ERR_Rcvd: Non-Fatal Error Messages Received Set when one or more Non-Fatal Uncorrectable error Messages have been received.
4	RWCST	0	FRST_UNCOR_FATAL: First Uncorrectable Fatal Set when the first Uncorrectable error message received is for a FATAL error.
3	RWCST	0	MULT_ERR_NOFAT_ERR: Multiple ERR_FATAL NO FATAL_Received Set when either a fatal or a non-fatal error message is received and ERR_FAT_NONFAT_RCVD is already set, i.e log from the 2nd Fatal or No fatal error message onwards
2	RWCST	0	ERR_FAT_NOFAT_RCVD: ERROR FATAL NOFATAL 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
1	RWCST	0	MULT_ERR_COR_RCVD: Multiple Correctable Error Received Set when either a correctable error message is received and ERR_CORR_RCVD is already set, i.e log from the 2nd Correctable error message onwards
0	RWCST	0	ERR_CORR_RCVD: First 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

Notes:

a. This applies to both internal generated Root port errors and those messages received from an external source.

4.8.14.17 RPERRSID: Error Source Identification Register

The Error Source Identification register identifies the source (Requestor ID) of first correctable and uncorrectable (Non-fatal/Fatal) errors reported in the Root Error Status register defined in Section 4.8.14.16, "RPERRSTS[7:0]: Root Error Status Register" on page 178. This register is updated regardless of the settings of the Root Control register defined in Section 4.8.13.12, "PEXRTCTRL[7:0]: PCI Express Root Control Register" on page 163 and the Root Error Command register defined in Section 4.8.14.15, "RPERRCMD[7:0]: Root Port Error Command" on page 178.

Device: 7-0 Function: 0 Offset: 134h			
Bit	**Attr**	**Default**	**Description**
31:16	ROST	0h	**ERR_FAT_NOFAT_SID: Fatal No Fatal Error Source ID** Requestor ID of the source when an Fatal or No Fatal error is received and the **ERR_FAT_NOFAT_RCVD bit** is not already set. i.e log ID of the first Fatal or Non Fatal error
15:0	ROST	0h	**ERR_CORR_SID: Correctable Error Source ID** Requestor ID of the source when a correctable error is received and the **ERR_CORR_RCVD** is not already set. i.e log ID of the first correctable error.

4.8.14.18 SPCAPID[7:0]: Intel® 7300 Chipset Specific Capability ID

This register identifies the Intel® 7300 Chipset specific extended capability structure and is a dummy holder. The set of registers defined below this offset can only used by BIOS/Software that is aware of the MCH configuration map for Advanced Error reporting.

Device: 7-0
Function: 0
Offset: 140h

Bit	Attr	Default	Description
31:20	RO	0h	**NXTCAPOFF: Next Capability Offset** It is set 000h (terminal chain)
19:16	RO	0h	**VN: Version Number** Version number for this capability structure
15:0	RO	0h	**EXTCAPID: Extended CAP_ID**

4.8.14.19 PEX_ERR_DOCMD[7:0]: PCI Express Error Do Command Register

Link Error Commands for doing the various signaling: ERR[2:0] and MCERR.

Device: 7-0
Function: 0
Offset: 144h

Bit	Attr	Default	Description
31:8	RV	0h	***Reserved***.
7:6	RW	00	**PEX_RP_FAT_MAP: Root Port steering for fatal errors** 00: ERR[0] 01: ERR[1] 10: ERR[2] 11: MCERR The Root Port Fatal errors are routed to one of the ERR[2:0] pins or MCERR.
5:4	RW	00	**PEX_RP_NF_MAP: Root Port steering for non-fatal errors** 00: ERR[0], 01: ERR[1] 10: ERR[2] 11: MCERR The Root Port Non Fatal (uncorrectable) errors are routed to one of the ERR[2:0] pins or MCERR.
3:2	RW	00	**PEX_RP_CORR_MAP: Root Port steering for correctable errors** 00: ERR[0], 01: ERR[1] 10: ERR[2] 11: MCERR The Root Port correctable errors are routed to one of the ERR[2:0] pins or MCERR.
1:0	RW	00	**PEX_DEV_UNSUP_MAP:** Report steering for unsupported request errors (master aborts) for legacy devices 00: ERR[0] 01: ERR[1] 10: ERR[2] 11: MCERR Unsupported request error report enable is in the Device control register. This is Error IO2.

4.8.14.20 EMASK_UNCOR_PEX[0]: Uncorrectable Error Detect Mask

This register masks (blocks) the detection of the selected error bits for the ESI port. When a specific error is blocked, it does NOT get reported or logged. For a description of the I/O Errors refer to Table 7-37, "Intel® 7300 Chipset Chipset Error List".

Device: 0
Function: 0
Offset: 148h

Bit	Attr	Default	Description
31:22	RV	0h	Reserved
21	RW	0	IO18ESIRstDetMsk: ESI Reset timeout
20	RW	0	IO2DetMsk: Received an Unsupported Request
19	RV	0	Reserved
18	RW	0	IO9DetMsk: Malformed TLP Status
17	RW	0	IO10DetMsk: Receiver Buffer Overflow Status
16	RW	0	IO8DetMsk: Unexpected Completion Status
15	RW	0	IO7DetMsk: Completer Abort Status
14	RW	0	IO6DetMsk: Completion Time-out Status
13	RW	0	IO5DetMsk: Flow Control Protocol Error Status
12	RW	0	IO4DetMsk: Poisoned TLP Status
11:6	RV	0h	Reserved
5	RW	0	IO19DetMsk: Surprise Link Down Status
4	RW	0	IO0DetMsk: Data Link Protocol Error Status
3:1	RV	0h	Reserved
0	RW	0	IO3DetMsk:Training Error Status ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error. ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.

4.8.14.21 EMASK_UNCOR_PEX[7:1]: Uncorrectable Error Detect Mask

This register masks (blocks) the detection of the selected error bits. When a specific error is blocked, it does NOT get reported or logged. For a description of the I/O Errors refer to Table 7-37, "Intel® 7300 Chipset Chipset Error List".

Device: 7-1
Function: 0
Offset: 148h

Bit	Attr	Default	Description
31:21	RV	0h	Reserved
20	RW	0	IO2DetMsk: Received an Unsupported Request
19	RV	0	Reserved
18	RW	0	IO9DetMsk: Malformed TLP Status
17	RW	0	IO10DetMsk: Receiver Buffer Overflow Status
16	RW	0	IO8DetMsk: Unexpected Completion Status
15	RW	0	IO7DetMsk: Completer Abort Status
14	RW	0	IO6DetMsk: Completion Time-out Status

Device: 7-1 Function: 0 Offset: 148h			
Bit	**Attr**	**Default**	Description
13	RW	0	IO5DetMsk: Flow Control Protocol Error Status
12	RW	0	IO4DetMsk: Poisoned TLP Status
11:6	RV	0h	Reserved
5	RW	0	IO19DetMsk: Surprise Link Down Status
4	RW	0	IO0DetMsk: Data Link Protocol Error Status
3:1	RV	0h	Reserved
0	RW	0	IO3DetMsk:Training Error Status ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error. ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.

4.8.14.22 EMASK_COR_PEX[7:0]: Correctable Error Detect Mask

This register masks (blocks) the detection of the selected bits. Normally all are detected. But software can choose to disable detecting any of the error bits.

Device: 7-0 Function: 0 Offset: 14Ch			
Bit	**Attr**	**Default**	**Description**
31:14	RV	0h	Reserved
13	RW	0	Advisory Non-fatal Error Detect Mask
12	RW	0	IO16DetMsk: Replay Timer Time-out Mask
11:9	RV	0h	Reserved
8	RW	0	IO15DetMsk: Replay_Num Rollover Mask
7	RW	0	IO14DetMsk: Bad DLLP Mask
6	RW	0	IO13DetMsk: Bad TLP Mask
5:1	RV	0h	Reserved
0	RW	0	IO12DetMsk: Receiver Error Mask

4.8.14.23 EMASK_RP_PEX[7:0]: Root Port Error Detect Mask

This register masks (blocks) the detection of the selected bits associated with the root port errors. Normally, all are detected.

Device: 7-0 Function: 0 Offset: 150h			
Bit	**Attr**	**Default**	**Description**
31:3	RV	0h	Reserved
2	RW	0	IO1DetMsk: Fatal Message Detect Mask
1	RW	0	IO11DetMsk: Uncorrectable Message Detect Mask
0	RW	0	IO17DetMsk: Correctable Message Detect Mask

4.8.14.24 PEX_FAT_FERR[7:0]: PCI Express First Fatal Error Register

This register records the occurrence of the first unmasked PCI Express FATAL error and written by Intel® 7300 Chipset if the respective bits are not set prior. After the recording of the 1st PCI Express fatal error of any given type, the MCH will lock down this register. Subsequent Fatal errors will be placed in the PEX_FAT_NERR register. The classification of uncorrectable errors into FATAL is based on the severity level of the UNCERRSEV register defined in Section 4.8.14.7, "UNCERRSEV[7:1]: Uncorrectable Error Severity" on page 175.

Device: 7-0
Function: 0
Offset: 154h

Bit	Attr	Default	Description
31:13	RV	0h	Reserved
12	RWCST	0	First_FAT_Err_IO19: Surprise Link Down
11	RWCST	0	First_FAT_Err_IO18: ESI Reset timeout
10	RWCST	0	First_FAT_Err_IO9: PEX - Malformed TLP
9	RWCST	0	First_FAT_Err_IO10: PEX - Receive Buffer Overflow Error
8	RWCST	0	First_FAT_Err_IO8: PEX - Unexpected Completion Error
7	RWCST	0	First_FAT_Err_IO7: PEX - Completer Abort
6	RWCST	0	First_FAT_Err_IO6: PEX - Completion Timeout
5	RWCST	0	First_FAT_Err_IO5: PEX - Flow Control Protocol Error
4	RWCST	0	First_FAT_Err_IO4: PEX - Poisoned TLP
3	RWCST	0	First_FAT_Err_IO3: PEX - Training Error ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error. ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.
2	RWCST	0	First_FAT_Err_IO2: PEX - Received Unsupported Request
1	RWCST	0	First_FAT_Err_IO1: PEX - Received Fatal Error Message
0	RWCST	0	First_FAT_Err_IO0: PEX - Data Link Layer Protocol Error

4.8.14.25 PEX_NF_COR_FERR[7:0]: PCI Express First Non-Fatal or Correctable Error Register

This register records the occurrence of the first unmasked PCI Express NON-FATAL (Uncorrectable) and CORRECTABLE errors. These errors are written by Intel® 7300 Chipset if the respective bits are not set prior. After the recording of the 1st PCI Express Non-fatal/Correctable error of any given type, the MCH will lock down this register. Subsequent Fatal errors will be placed in the PEX_FAT_NERR register. The classification of uncorrectable errors into FATAL or Non-Fatal is based on the UNCERRSEV register defined in Section 4.8.14.7, "UNCERRSEV[7:1]: Uncorrectable Error Severity" on page 175.

Device: 7-0
Function: 0
Offset: 158h

Bit	Attr	Default	Description
31:18	RV	0h	Reserved
17	RWCST	0	First_NFAT_Corr_Err_IO19: PEX - Surprise Link Down (uncorrectable)
16	RWCST	0	First_NFAT_COR_Err_IO17: PEX - Received Correctable Error Message
15	RWCST	0	First_NFAT_COR_Err_IO16: PEX - Replay Timer Timeout (correctable)
14	RWCST	0	First_NFAT_COR_Err_IO15: PEX - Replay_Num Rollover (correctable)
13	RWCST	0	First_NFAT_COR_Err_IO14: PEX - BAD DLLP Error (correctable)
12	RWCST	0	First_NFAT_COR_Err_IO13: PEX - Bad TLP Error (correctable)
11	RWCST	0	First_NFAT_COR_Err_IO12: PEX - Receiver Error (correctable)
10	RWCST	0	First_NFAT_COR_Err_IO11: PEX - Received Non Fatal (uncorrectable) Error Message
9	RWCST	0	First_NFAT_COR_Err_IO10: PEX - Receive Buffer Overflow Error (uncorrectable)
8	RWCST	0	First_NFAT_COR_Err_IO9: PEX -Malformed TLP (uncorrectable)
7	RWCST	0	First_NFAT_COR_Err_IO8: no error logged by this bit. See note below
6	RWCST	0	First_NFAT_COR_Err_IO7: no error logged by this bit. See note below
5	RWCST	0	First_NFAT_COR_Err_IO6: PEX - Completion Timeout (uncorrectable)
4	RWCST	0	First_NFAT_COR_Err_IO5: PEX - Flow Control Protocol Error (uncorrectable)
3	RWCST	0	First_NFAT_COR_Err_IO4: no error logged by this bit. See note below
2	RWCST	0	First_NFAT_COR_Err_IO3: PEX - Training Error (uncorrectable) ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.
1	RWCST	0	First_NFAT_COR_Err_IO2: no error logged by this bit. See note below
0	RWCST	0	First_NFAT_COR_Err_IO0: PEX - Data Link Layer Protocol Error (uncorrectable)

Note: This register does not log the following Advisory Errors: IO2 (Unsupported request error), IO4 (Poison TLP Error), IO7 (Completer Abort), IO8 (Unexpected Completion Error). These errors are still logged in the PCI/PCIe architected registers: UNCERRSTS[7:0] and SECSTS[7:0] as expected. These errors are still reported on ERR_N[2:0] pins or FSB{3:0}MCERR_N based on PECX_ERR_DOCMD[7:0] register setting. BIOS should read the standard PCI/PCIe registers mentioned above to determine when these errors occur.

4.8.14.26 PEX_FAT_NERR[7:0]: PCI Express Next Fatal Error Register

This register records the subsequent occurrences after the first unmasked PCI Express FATAL errors and written by the Intel® 7300 Chipset if the respective bits are set prior in the PEX_FAT_FERR register.

Device: 7-0
Function: 0
Offset: 15Ch

Bit	Attr	Default	Description
31:13	RV	0h	Reserved
12	RWCST	0	Next_FAT_Err_IO19: Surprise Link Down
11	RWCST	0	Next_FAT_Err_IO18: ESI - ESI Reset timeout
10	RWCST	0	Next_FAT_Err_IO9: PEX - Malformed TLP
9	RWCST	0	Next_FAT_Err_IO10: PEX - Receive Buffer Overflow Error
8	RWCST	0	Next_FAT_Err_IO8: PEX - Unexpected Completion Error
7	RWCST	0	Next_FAT_Err_IO7: PEX - Completer Abort
6	RWCST	0	Next_FAT_Err_IO6: PEX - Completion Timeout
5	RWCST	0	Next_FAT_Err_IO5: PEX - Flow Control Protocol Error
4	RWCST	0	Next_FAT_Err_IO4: PEX - Poisoned TLP
3	RWCST	0	Next_FAT_Err_IO3: PEX - Training Error ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error. ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.
2	RWCST	0	Next_FAT_Err_IO2: PEX - Received Unsupported Request
1	RWCST	0	Next_FAT_Err_IO1: PEX - Received Fatal Error Message
0	RWCST	0	Next_FAT_Err_IO0: PEX - Data Link Layer Protocol Error

4.8.14.27 PEX_NF_COR_NERR[7:0]: PCI Express Non Fatal or Correctable Next Error Register

These errors are written by the Intel® 7300 Chipset if the respective bits are set prior in PEX_NF_COR_FERR register. This register records the subsequent occurrences of unmasked PCI Express NON-FATAL (Uncorrectable) and CORRECTABLE errors.

Device: 7-0
Function: 0
Offset: 160h

Bit	Attr	Default	Description
31:18	RV	0h	Reserved
17	RWCST	0	Next_NFAT_Corr_Err_IO19: PEX - Surprise Link Down (uncorrectable)
16	RWCST	0	Next_NFAT_COR_Err_IO17: PEX - Received Correctable Error Message
15	RWCST	0	Next_NFAT_COR_Err_IO16: PEX - Replay Timer Timeout (correctable)
14	RWCST	0	Next_NFAT_COR_Err_IO15: PEX - Replay_Num Rollover (correctable)
13	RWCST	0	Next_NFAT_COR_Err_IO14: PEX - BAD DLLP Error (correctable)
12	RWCST	0	Next_NFAT_COR_Err_IO13: PEX - Bad TLP Error (correctable)
11	RWCST	0	Next_NFAT_COR_Err_IO12: PEX - Receiver Error (correctable)
10	RWCST	0	Next_NFAT_COR_Err_IO11: PEX - Received Non Fatal (uncorrectable) Error Message

Device: 7-0 Function: 0 Offset: 160h			
Bit	**Attr**	**Default**	**Description**
9	RWCST	0	Next_NFAT_COR_Err_IO10: PEX - Receive Buffer Overflow Error (uncorrectable)
8	RWCST	0	Next_NFAT_COR_Err_IO9: PEX -Malformed TLP (uncorrectable)
7	RWCST	0	Next_NFAT_COR_Err_IO8: no error logged by this bit. See note below
6	RWCST	0	Next_NFAT_COR_Err_IO7: no error logged by this bit. See note below
5	RWCST	0	Next_NFAT_COR_Err_IO6: PEX - Completion Timeout (uncorrectable)
4	RWCST	0	Next_NFAT_COR_Err_IO5: PEX - Flow Control Protocol Error (uncorrectable)
3	RWCST	0	Next_NFAT_COR_Err_IO4: no error logged by this bit. See note below
2	RWCST	0	Next_NFAT_COR_Err_IO3: PEX - Training Error (uncorrectable) ***Note:*** This field should not be used for obtaining Training error status due to a recent *PCI Express Base Specification*, Revision 1.0a Errata Dec 2003 to remove training error ***Note:*** However, refer to PEXLNKSTS.TERR for basic training error indication in Link Status register as an alternative.
1	RWCST	0	Next_NFAT_COR_Err_IO2: no error logged by this bit. See note below
0	RWCST	0	Next_NFAT_COR_Err_IO0: PEX - Data Link Layer Protocol Error (uncorrectable)

Note: This register does not log the following Advisory Errors: IO2 (Unsupported request error), IO4 (Poison TLP Error), IO7 (Completer Abort), IO8 (Unexpected Completion Error). These errors are still logged in the PCI/PCIe architected registers: UNCERRSTS[7:0] and SECSTS[7:0] as expected. These errors are still reported on ERR_N[2:0] pins or FSB{3:0}MCERR_N based on PECX_ERR_DOCMD[7:0] register setting. BIOS should read the standard PCI/PCIe registers mentioned above to determine when these errors occur.

4.8.14.28 PEX_UNIT_FERR[7:1]: PCI Express First Unit Error Register

This register records the occurrence of the first unit errors that are specific to this PCI Express port caused by external activities. e.g. VPP error due to a malfunctioning port on the SMBUS that did not receive acknowledge due to a PCI Express hot-plug event. The unit errors are sent to the Coherency Engine to classify as to which port cluster it came from (ports 1-3 (IOU0) or ports 4-7 (IOU1) and the errors are recorded in Coherence Engine and appropriate interrupts generated through ERR pins.

Device: 7-1 Function: 0 Offset: 168h			
Bit	**Attr**	**Default**	**Description**
31:1	RV	0h	Reserved
0	RWCST	0	**First_FAT_VPP_Err: VPP Error for PCI Express port** Records the occurrence of the first VPP error if this bit is not set prior. Software clears this when the error has been serviced.

4.8.14.29 PEX_UNIT_NERR[7:1]: PCI Express Next Unit Error Register

This register records the occurrence of the subsequent unit errors that are specific to this PCI Express port caused by external activities. e.g. VPP error due to a malfunctioning port on the SMBUS that did not receive acknowledge due to a PCI Express hot-plug event. The next error unit errors are sent to the Coherency Engine where the errors are further recorded and appropriate interrupts generated through ERR pins. Refer to the RAS chapter for the VPP Error flow diagram.

Device: 7-1
Function: 0
Offset: 16Ch

Bit	Attr	Default	Description
31:1	RV	0h	Reserved
0	RWCST	0	**Next_FAT_VPP_Err: VPP Error for PCI Express port** Records the occurrence of subsequent VPP errors after the PEX_UNIT_FERR.First_FAT_VP_ERR is set. Software clears this when the error has been serviced.

4.8.15 PCI Express IBIST Registers

4.8.15.1 PEX[7:0]IBCTL: PCIe IBIST Control Register

This register contains the control bits and status information necessary to operate the Fixed and Open modes of the IBIST logic. The default settings allow the CMM logic to operate with link width of a PEX port. Usage of control characters undefined in the PCI Express specification may have unpredictable results, and is therefore not recommended.

Device: 7-0
Function: 0
Offset: 380h

Bit	Attr	Default	Description
31	RW	0	**SYMTYPSEL3: Symbol[3] Type Select** 1: selects Symbol [3] as a control character 0: selects Symbol [3] to a data character
30	RW	1	**SYMTYPSEL2: Symbol[2] Type Select** 1: selects Symbol [2] as a control character 0: selects Symbol [2] to a data character
29	RW	0	**SYMTYPSEL1: Symbol[1] Type Select** 1: selects Symbol [1] as a control character 0: selects Symbol [1] to a data character
28	RW	1	**SYMTYPSEL0: Symbol[0] Type Select** 1: selects Symbol [0] as a control character 0: selects Symbol [0] to a data character
27:23	RV	0	**Reserved**
22:14	RWCST	0h	**ERRVAL: Error Value** This is the raw 9-bit error value captured on the lane that asserted the Error Detected bit (ERRDET) or the Global Error status bit if this register is implemented. The value must be extracted in the datapath before the 10b/8b decoder in order to examine its contents for debugging potential link errors.

Device: 7-0
Function: 0
Offset: 380h

Bit	Attr	Default	Description
13:9	RWCST	0h	**ERRLNNUM: Error Lane Number** This field indicates which lane reported the error that was detected when ERRDET was asserted. Note: When the number of lanes reporting exceeds 32, this field will show an aliased error lane number and cannot be used to indicate the errant lane. Larger lane indications will require an extended register to display accurate information.
8	RWCST	0	**ERRDET: Error Detected** A mis-compare between the transmitted symbol and the symbol received on link indicates an error condition occurred. Refer to Error Value, Error Symbol Pointer and Error Symbol Type bit fields for further information about fault locations. This bit is cleared by writing a logic '1' and it remains asserted through reset (sticky). 0: No error detected 1: Error Detected Note: The error signal that causes this bit to be set should be made available externally to the IBIST logic. It is implementation specific as to how this is accomplished. The purpose is for symbol (bit) error rate testing. It is assumed that this signal is either sent to a performance counter or an external pin for signal assertion accumulation. There is not any error counting resources available in this spec.
7	RW	0	**SUPSKP: Suppress Skips** 0: Skips are still inserted in the IBIST data stream during IBIST test operations. 1: Skip insertion is suppressed
6:4	RW	000	**DSYMINJLNUM: Delay Symbol Injection Lane Number** This selects the Lane number to inject the delay symbol pattern. All 8 values could be valid depending on the setting of the IBEXTCTL.LNMODUEN bit field. This is true regardless of whether this IBIST engine is instantiated for a x4 or a x8 port.
3	RW	1	**AUTOSEQEN: Automatic Sequencing Enable of Delay Symbol** 0: Disable delay symbol auto-sequence. IBIST does not automatically sequence the delay symbol across the width of the link. 1: Enable delay symbol auto-sequencing.
2	RV	0	**Reserved**
1	RW	0	**INITDISP: Initial Disparity** This bit sets the disparity of the first IBIST data pattern symbol. The default is negative meaning that the first symbol transmitted by Tx will have a negative disparity regardless of what the running disparity is. This allows a deterministic pattern set to be transmitted on the link for every IBIST run. If IBIST causes a discontinuous disparity error in the receiver this error can be ignored in the reporting register. It will not affect the operation of the IBIST since it is outside of its domain. Higher levels of software management must be aware that side effects from running IBIST could cause other errors and should they be ignored. 0: Disparity starts as negative 1: Disparity starts as positive
0	RW	0	**IBSTR: IBIST Start** This bit is OR'ed with the global start bit. 0: Stop IBIST 1: Start IBIST

Note only valid PCI Express control characters/symbols are allowed for IBIST registers for testing.

4.8.15.2 PEX[7:0]IBSYMBUF: PCIe IBIST Symbol Buffer

This register contains the character symbols that are transmitted on the link. Note only valid PCI Express control characters/symbols are allowed for IBIST registers for testing. Usage of control characters undefined in the PCI Express specification may have unpredictable results, and is therefore not recommended.

Device: 7-0
Function: 0
Offset: 384h

Bit	Attr	Default	Description
31:24	RW	4Ah	**CHARSYM3: Character Symbol [3]** This character is symbol [3] of the four-symbol pattern buffer. The default value is the 8-bit encoding for D10.2.
23:16	RW	BCh	**CHARSYM2: Character Symbol [2]** This character is symbol [3] of the four-symbol pattern buffer. The default value is the 8-bit encoding for K28.5.
15:8	RW	B5h	**CHARSYM1: Character Symbol [1]** This character is symbol [3] of the four-symbol pattern buffer. The default value is the 8-bit encoding for D21.5.
7:0	RW	BCh	**CHARSYM0: Character Symbol [0]** This character is symbol [3] of the four-symbol pattern buffer. The default value is the 8-bit encoding for K28.5.

4.8.15.3 PEX[7:0]IBEXTCTL: PCIe IBIST Extended Control Register

This register extends the functionality of the IBIST with pattern loop counting, skip character injection, and symbol management. A bit is provided to ignore the count value and loop continuously for port testing.

Device: 7-0
Function: 0
Offset: 388h

Bit	Attr	Default	Description
31:29	RV	0h	**Reserved**
28	RW	0	**FRCENT: Forced Entry:** Setting this bit forces entry into master IBIST loopback state when the Start bit is asserted. No TS1s are sent when enabled. The IBIST is granted direct control of the transmitted path regardless of which state the LTSSM is in. The IBIST state machine sends the contents of the pattern buffer until the stop condition is reached. The receiver isn't expected to perform any error checking and must ignore input symbols. 0: Execute normally 1: Force to Loopback state as a 'master' condition.
27:26	RW	01	**LNMODUEN: Lane Modulo Enable for Delay Symbol Injection** 00: No symbols sent on lanes 01: Delay symbols sent on modulo 4 group of lanes across the width of the port. 10: Delay symbols sent on modulo 8 group of lanes across the width of the port. 11: Reserved
25	RW	0	**DISSTOP: Disable Stop on Error** 0: Enable Stop on Error 1: Disable Stop on Error. The IBIST engine continues to run in its current mode in the presence of an error. If an error occurs overwrite the error status collected from a previous error event.

Device: 7-0 Function: 0 Offset: 388h			
Bit	**Attr**	**Default**	**Description**
24	RW	0	**LPCON: Loop Continuously** 0: Use loop counter. Test terminates at the end of the global count. 1: Loop symbols continuously.
23:12	RW	000h	**SKPCNTINT: Skip Count Interval** This register indicates when a skip order sequence is sent on the transmitter. Upon reaching this count the transmitter sends an SOS then clears the skip counter and counting resumes until the next match on the skip count interval. 000: No Skip Ordered Sets are sent on TX. nnn: The number of 8 symbol sets transmitted before a Skip Ordered Set is sent.
11:0	RW	07Fh	**LOOPCNTLIM: Loop Count Limit** This register indicates the number of times the data symbol buffer is looped as a set of 8 symbol times. If LOOPCON is set then this count limit is ignored. 00: No symbols are sent from symbol buffer unless LOOPCON is set. If LOOPCON is cleared and this value is 000h then the transmitter immediately exits out of loopback state by sending EIOS without sending a pattern buffer payload. 01-FFF: 1 to 4095 sets of symbols from the symbol buffer. One set of symbols is defined as either two copies of the contents of the buffer or the modified delayed symbol set.

4.8.15.4 PEX[7:0]IBDLYSYM: PCIe IBIST Delay Symbol

This register stores the value of the delay symbol used in lane inversion cross-talk testing. Usage of control characters undefined in the PCI Express specification may have unpredictable results, and is therefore not recommended.

Device: 7-0 Function: 0 Offset: 38Ch			
Bit	**Attr**	**Default**	**Description**
15:9	RV	0h	***Reserved***
8:0	RW	1BCh	**DLYSYM: Delay Symbol** This is the 9-bit delay symbol value used (default is K28.5).

4.8.15.5 PEX[7:0]IBLOOPCNT: PCIe IBIST Loop Counter

This register stores the current value of the loop counter.

Device: 7-2, 0
Function: 0
Offset: 38Eh

Bit	Attr	Default	Description
15:12	RV	0h	***Reserved***
11:0	RWCST	000h	**LOOPCNTVAL: Loop Count Value** Once the IBIST is engaged, loop counts are incremented when a set of 8 symbols has been received. If an error occurs, this register reflects the loop count value of the errant Rx lane. If there is no error then this register reads 00h. Note: Since each receiver is not deskewed with respect to the IBIST pattern generator we cannot have a coherent loop count value with N number of receivers and only one loop counter. It would require additional logic to select which receiver indicates the count.

4.8.15.6 PEX[7:0]IBLNS[3:0]: PCIe IBIST Lane Status

This register stores the error status for individual lanes of the link. The status indicates the error, the pointer to the symbol of the miss-compare and which set of symbols the receiver was comparing against (i.e. the set of 8 data symbols or the delay symbol set). The Lane Start Enable bit allows selective testing of lanes by enabling which lane(s) are executing the IBIST test sequence. This register is implemented per lane of the PEX link which is controlled by the IOIBCTL register.

Device: 7-2, 0
Function: 0
Offset: 393h, 392h, 391h, 390h

Bit	Attr	Default	Description
7	RWCST	0	**ERRPTRTYP: Error Symbol Pointer Type** This bit indicates whether or not the errant symbol pointer was a delay symbol set. If an IBIST engine is implemented with the MISR compare method then this field is reserved. 0: Errant symbol pointer was a DATA symbol set 1: Errant symbol pointer was a DELAY symbol set
6:4	RWCST	0h	**ERRPTR: Error Symbol Pointer** This value indicates which symbol of the 8 possible symbols sent on the lane, as a set of characters, failed. The value corresponds to position of the set of 8 symbols. If an IBIST engine is implemented with the MISR compare method then this field is reserved.
3	RWCST	0	**IBLOOPSTAT: IBIST Loopback State Status:** This bit is set when the Rx received a TS1 with the loopback bit set. Write a logic '1' to clear. 0: IBIST did not receive a TS1 with loopback bit set. 1: IBIST received a TS1 with loopback bit set. **NOTE:** This bit is set even for lanes that are not participating in the loopback negotiation and hence will not receive TS1 (for a x1 configuration, for instance). In other words, this bit is set for all lanes regardless of link width.
2	RV	0	**Reserved.**
1	RWCST	0	**ERRLNSTAT: Error Lane Status** Error assertion for this lane. Writing a logic '1' will clear this bit. This bit is sticky. 0: No error on this lane 1: Error has occurred on this lane

Device: 7-2, 0 Function: 0 Offset: 393h, 392h, 391h, 390h			
Bit	**Attr**	**Default**	**Description**
0	RW	1	**LNSTREN: Lane Start Enable** When the lane is disabled, the receiver's (Rx) IBIST error reporting is suppressed. 0: This lane is disabled from IBIST testing, no Tx transmissions and Rx error reporting is suppressed. 1: Lane enabled. Allows the port start bit to begin IBIST symbol operations on this lane.

4.8.15.7 DIO0IBSTAT: PCIe IBIST Completion Status Register

Device: 0 Function: 0 Offset: 394h			
Bit	**Attr**	**Default**	**Description**
7	RO	0	**IBSTAT7: IBIST Status port 7** 0: IBIST either has not started the first time or it is still running. 1: IBIST is done. This bit will be cleared by hardware when the start bit is asserted.
6	RO	0	**IBSTAT6: IBIST Status port 6** See above.
5	RO	0	**IBSTAT5: IBIST Status port 5** See above.
4	RO	0	**IBSTAT4: IBIST Status port 4** See above.
3	RO	0	**IBSTAT3: IBIST Status port 3** See above.
2	RO	0	**IBSTAT2: IBIST Status port 2** See above.
1	RO	0	**IBSTAT1: IBIST Status port 1** See above.
0	RO	0	**IBSTAT0: IBIST Status port 0** See above.

4.8.15.8 DIO0IBERR: PCIe IBIST Error Register

Device: 0 Function: 0 Offset: 395h			
Bit	**Attr**	**Default**	**Description**
7	RWCST	0	**P7ERRDET: Error Detected on port 7**
6	RWCST	0	**P6ERRDET: Error Detected on port 6**
5	RWCST	0	**P5ERRDET: Error detected on port 5**
4	RWCST	0	**P4ERRDET: Error detected on port 4**
3	RWCST	0	**P3ERRDET: Error Detected on port 3**
2	RWCST	0	**P2ERRDET: Error Detected on port 2**
1	RWCST	0	**P1ERRDET: Error detected on port 1**
0	RWCST	0	**P0ERRDET: Error detected on port 0**

4.8.15.9 DIOIBSTR: PCIe IBIST Global Start/Status Register

This register contains the global start for all the ports in the Intel® 7300 Chipset component simultaneously. One start bit is placed in the register for each port. IBIST will start at approximately the same time on all ports written to with a 1 in the same write access.

Device: 0
Function: 0
Offset: 398h

Bit	Attr	Default	Description
7	RW	0	**START7:** Writing a 1 starts IBIST on port 7.
6	RW	0	**START6:** Writing a 1 starts IBIST on port 6.
5	RW	0	**START5:** Writing a 1 starts IBIST on port 5.
4	RW	0	**START4:** Writing a 1 starts IBIST on port 4.
3	RW	0	**START3:** Writing a 1 starts IBIST on port 3.
2	RW	0	**START2:** Writing a 1 starts IBIST on port 2.
1	RW	0	**START1:** Writing a 1 starts IBIST on port 1.
0	RW	0	**START0:** Writing a 1 starts IBIST on port 0.

4.8.16 Error Registers

This section describes the registers that record the first and next errors, logging, detection masks, signalling masks, and error injection control.

Definitions:

- Hard Error: An error caused by a hardware malfunction. Detectable and correctable hard errors include dead DRAM chips and opens and shorts on DRAM data bus or FBD bus.
- Soft Error: An error caused by a transient event such as an alpha particle strike. The hardware functions correctly but has been pushed beyond its margins by the transient condition.
- DRAM hard error: A failure of a DRAM chip or a failure in a DRAM data bus trace.
- DRAM soft error: A bit-flip of a DRAM cell or a transient error on a DRAM data bus.
- FBD hard error: A failure of an FBD link trace on northbound or southbound direction.
- FBD soft error: A transient error on an FBD link

Philosophy:

The system in all configurations must provide some level of protection against soft errors. This is because by design the DRAM cells and FBD link are allowed a maximum error rate. This error rate is low enough to assume that each error happens in isolation and does not interact with other errors in the system. Therefore, while the system is designed to continue to operate in the face of an isolated soft error, it may fail if faced with multiple soft errors that occur close enough in time or space. Except for a few cases, occurrences of double soft errors are detected and do not lead to data corruption.

Protection is provided to improve reliability of the systems against hardware malfunction. In particular, protection is provided against failing DRAM chips and the same mechanisms provide protection against failing DRAM data bus traces. Optional

protection is also provided against failing FBD link traces. End users concerned with Availability should replace the failed component within a reasonable period to minimize the likelihood that another component will fail. The concern is with providing protection against a single component failing to avoid system failure or data corruption due to multiple component failures.

It may not be possible to repair a system with a failed component prior to the occurrence of a soft error. Therefore, it is critical to detect all cases of a soft error occurring while there is a failed component, and when possible allow for correction of the soft error and continued operation.

Recommended Configurations:

1. Basic RAS: ECC, FBD CRC, FBD Retry, scrubbing This mode provides basic protection against soft errors. The system should continue to operate in the presence of transient errors on the FBD link as long as:
   a. Any single error only affects a single FBD trace
   b. The duration of any single error is less than 12 UI (one lane) or less than 6 random bitsc. The errors are separated by at least 1 ms
   c. The errors are separated by at least 1 ms

The system should continue to operate in the presence of bit errors on the DRAM as long as:

   a. Any single error only affects a single bit per ECC codeword
   b. Errors to the same ECC codeword are spaced by 2x the scrub period

This configuration will also provide some protection against DRAM hard errors. However, a DRAM soft error to the same DIMM may cause an uncorrectable ECC error and result in a system failure. To minimize the likelihood of system failure, it must be repaired immediately.

2. Basic RAS + DIMM Sparing

This configuration provides improved protection against DRAM hard errors. It provides the soft error protection above, plus continued operation in the presence of DRAM hard errors as long as:

   a. There are no soft errors in the DIMM that have not been scrubbed at the time the hard error occurs, and no new soft error occurs at the time the hard error occurs, and no new soft error occurs to the DIMM before the sparing operation completes

3. Basic RAS + Mirroring

This configuration provides an alternate mechanism to protect against hard errors. It provides protection against FBD Hard errors (and some other hard errors such as AMB device failures) as long as:

   a. There is no DRAM hard error and no other FBD hard error in the system
   b. There is no FBD soft error detected within 1 ms of when the FBD hard error was detected and the branch degrades.

Note: If an FBD hard error that causes a degrade operation to happen, the additional protections provided by mirroring are lost. It provides protection against DRAM hard errors as long as:

   a. The assumptions in configuration #1 for soft error rate are met
   b. There is only a single DRAM hard error in the system

Guidelines for Error Injection:

To test error correction (i.e. continuing operation in presence of errors):

- Select a supported configuration
- Select a sequence of errors supported by that configuration

— One hard error and a sequence of soft errors in any order

- Inject errors, separating each error injection by 1 ms.
- Do not inject multiple errors to the same ECC codeword
- For DIMM sparing, do not inject any DRAM soft error to the same DIMM as the DRAM hard error before the sparing operation completes

To test Error Detection & Containment:

- Select a configuration
- Select a sequence of errors:

— (1 hard error) or (1 hard error + 1 soft error) or (2 soft errors)

- Inject errors with ANY timing except:

— All bits of a lane failing if injecting hard error in FBD link

— Two soft errors on FBD link within 12 UI (or any pattern within the capability of the CRC detection scheme)

- Ensure that system either continues without fault or machine checks

4.8.16.1 FERR_GLOBAL_HI: Global High First Error Register

The first error is recorded in this register. All subsequent errors, regardless of type, are recorded in NERR_GLOBAL.

Device: 16
Function: 2
Offset: 48h

Bit	Attr	Default	Description
31-4	RV	0	Reserved
3	RWCST	0	Global_FERR_35 FSB3 Fatal Error
2	RWCST	0	Global_FERR_34 FSB 2 Fatal Error
1	RWCST	0	Global_FERR_33 FSB3 non - Fatal Error
0	RWCST	0	Global_FERR_32 FSB 2 non - Fatal Error

4.8.16.2 FERR_GLOBAL_LO: Global Low First Error Register

The first error is recorded in this register. All subsequent errors, regardless of type, are recorded in NERR_GLOBAL.

Device: 16 Function: 2 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
31	RWCST	0	Global_FERR_31 Internal MCH Fatal Error
30	RWCST	0	Global_FERR_30 Intel® QuickData Technology Device Fatal Error
29	RWCST	0	Global_FERR_29 FSB1 Fatal Error
28	RWCST	0	Global_FERR_28 FSB 0 Fatal Error
27	RWCST	0	Global_FERR_27 FBD Channel 3 Fatal Error
26	RWCST	0	Global_FERR_26 FBD Channel 2 Fatal Error
25	RWCST	0	Global_FERR_25 FBD Channel 1 Fatal Error
24	RWCST	0	Global_FERR_24 FBD Channel 0 Fatal Error
23	RWCST	0	Global_FERR_23 PCI Express Device 7Fatal Error
22	RWCST	0	Global_FERR_22 PCI Express Device 6 Fatal Error
21	RWCST	0	Global_FERR_21 PCI Express Device 5 Fatal Error
20	RWCST	0	Global_FERR_20 PCI Express Device 4 Fatal Error
19	RWCST	0	Global_FERR_19 PCI Express Device 3 Fatal Error
18	RWCST	0	Global_FERR_18 PCI Express Device 2 Fatal Error
17	RWCST	0	Global_FERR_17 PCI Express Device 1 Fatal Error
16	RWCST	0	Global_FERR_16 ESI Fatal Error
15	RWCST	0	Global_FERR_15 Internal MCH Non-Fatal Error
14	RWCST	0	Global_FERR_14 Intel® QuickData Technology Device Non Fatal Error
13	RWCST	0	Global_FERR_13 FSB1 Non-Fatal Error
12	RWCST	0	Global_FERR_12 FSB 0 Non-Fatal Error
11	RWCST	0	Global_FERR_11 FBD Channel 3 Non-Fatal Error

Device: 16 Function: 2 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
10	RWCST	0	Global_FERR_10 FBD Channel 2 Non-Fatal Error
9	RWCST	0	Global_FERR_09 FBD Channel 1 Non-Fatal Error
8	RWCST	0	Global_FERR_08 FBD Channel 0 Non-Fatal Error
7	RWCST	0	Global_FERR_07 PCI Express Device 7Non-Fatal Error
6	RWCST	0	Global_FERR_06 PCI Express Device 6 Non-Fatal Error
5	RWCST	0	Global_FERR_05 PCI Express Device 5 Non-Fatal Error
4	RWCST	0	Global_FERR_04 PCI Express Device 4 Non-Fatal Error
3	RWCST	0	Global_FERR_03 PCI Express Device 3 Non-Fatal Error
2	RWCST	0	Global_FERR_02 PCI Express Device 2 Non-Fatal Error
1	RWCST	0	Global_FERR_01 PCI Express Device 1Non-Fatal Error
0	RWCST	0	Global_FERR_00 ESI Non-Fatal Error

4.8.16.3 NERR_GLOBAL: Global Next Error Register

Once an error of any type has been logged in the FERR_GLOBAL_HI and FERR_GLOBAL_LO registers, subsequent errors, regardless of type, are logged in the NERR_GLOBAL register. This allows the ability to know which error happened first.

Device: 16 Function: 2 Offset: 44h			
Bit	**Attr**	**Default**	**Description**
31	RWCST	0	Global_NERR_31 Internal MCH Fatal Error
30	RWCST	0	Global_NERR_30 Intel® QuickData Technology Device Fatal Error
29	RWCST	0	Global_NERR_29 FSB1 Fatal Error
28	RWCST	0	Global_NERR_28 FSB 0 Fatal Error
27	RWCST	0	Global_NERR_27 FSB2 Fatal Error
26	RWCST	0	Global_NERR_26 FSB 3 Fatal Error
25	RV	0	*Reserved*

Device: 16
Function: 2
Offset: 44h

Bit	Attr	Default	Description
24	RWCST	0	Global_NERR_24 FBD Channel 0,1,2 or 3 Fatal Error
23	RWCST	0	Global_NERR_23 PCI Express Device 7Fatal Error
22	RWCST	0	Global_NERR_22 PCI Express Device 6 Fatal Error
21	RWCST	0	Global_NERR_21 PCI Express Device 5 Fatal Error
20	RWCST	0	Global_NERR_20 PCI Express Device 4 Fatal Error
19	RWCST	0	Global_NERR_19 PCI Express Device 3 Fatal Error
18	RWCST	0	Global_NERR_18 PCI Express Device 2 Fatal Error
17	RWCST	0	Global_NERR_17 PCI Express Device 1Fatal Error
16	RWCST	0	Global_NERR_16 ESI Fatal Error
15	RWCST	0	Global_NERR_15 Internal MCH Non-Fatal Error
14	RWCST	0	Global_NERR_14 Intel® QuickData Technology Device Non Fatal Error
13	RWCST	0	Global_NERR_13 FSB1 Non-Fatal Error
12	RWCST	0	Global_NERR_12 FSB 0 Non-Fatal Error
11	RWCST	0	Global_NERR_11 FSB2 Non-Fatal Error
10	RWCST	0	Global_NERR_10 FSB 3 Non-Fatal Error
9	RV	0h	Reserved
8	RWCST	0	Global_NERR_08 FBD Channel 0,1, 2 or 3 Non-Fatal Error
7	RWCST	0	Global_NERR_07 PCI Express Device 7 Non-Fatal Error
6	RWCST	0	Global_NERR_06 PCI Express Device 6 Non-Fatal Error
5	RWCST	0	Global_NERR_05 PCI Express Device 5 Non-Fatal Error
4	RWCST	0	Global_NERR_04 PCI Express Device 4 Non-Fatal Error
3	RWCST	0	Global_NERR_03 PCI Express Device 3 Non-Fatal Error
2	RWCST	0	Global_NERR_02 PCI Express Device 2 Non-Fatal Error
1	RWCST	0	Global_NERR_01 PCI Express Device 1 Non-Fatal Error

Device: 16 Function: 2 Offset: 44h			
Bit	**Attr**	**Default**	**Description**
0	RWCST	0	Global_NERR_00 ESI Non-Fatal Error

4.8.16.4 FERR_FAT_FSB[3:0]: FSB First Fatal Error Register

Device: 17 Function: 0,3 Offset: 40h, C0h			
Bit	**Attr**	**Default**	**Description**
7:6	RV	00	*Reserved*
5	RWCST	0	**F9Err: FSB protocol Error**
4	RV	0h	*Reserved*
3	RWCST	0	**F2Err: Unsupported Processor Bus Transaction**
2:1	RV	0h	*Reserved*
0	RWCST	0	**F1Err: Request/Address Parity Error**

4.8.16.5 FERR_NF_FSB[3:0]: FSB First Non-Fatal Error Register

Device: 17 Function: 0,3 Offset: 41h, C1h			
Bit	**Attr**	**Default**	**Description**
7:3	RV	00000	*Reserved*
2	RWCST	0	**F7Err: Detected MCERR from a processor**
1	RWCST	0	**F8Err: Detected BINIT from a processor**
0	RWCST	0	**F6Err: Parity Error in Data from FSB Interface**

4.8.16.6 NERR_FAT_FSB[3:0]: FSB Next Fatal Error Register

This register logs all FSB subsequent errors after the FERR_FAT_FSB has logged the 1st fatal error.

Device: 17 Function: 0,3 Offset: 42h, C2h			
Bit	**Attr**	**Default**	**Description**
7:6	RV	00	*Reserved*
5	RWCST	0	**F9Err: FSB protocol Error**
4	RV	0h	*Reserved*
3	RWCST	0	**F2Err: Unsupported Processor Bus Transaction**
2:1	RV	0h	*Reserved*
0	RWCST	0	**F1Err: Request/Address Parity Error**

4.8.16.7 NERR_NF_FSB[3:0]: FSB Next Non-Fatal Error Register

This register logs all FSB subsequent errors after the FERR_NF_FSB has logged the 1st fatal error.

Device: 17
Function: 0,3
Offset: 43h, C3h

Bit	Attr	Default	Description
7:3	RV	00000	*Reserved*
2	RWCST	0	**F7Err: Detected MCERR from a processor**
1	RWCST	0	**F8Err: Detected BINIT from a processor**
0	RWCST	0	**F6Err: Parity Error in Data from FSB Interface**

4.8.16.8 NRECFSB[3:0]: NonRecoverable FSB Error Log Register

FSB Log registers for non recoverable errors when a fatal error is logged in its corresponding FERR_FAT_FSB Register.

Device: 17
Function: 0,3
Offset: 44h, C4h

Bit	Attr	Default	Description
31:29	RV	000	*Reserved*
28:24	ROST	00000	**REQA: REQa[4:0]# fields of the FSB**
23:21	ROST	000	**REQB: REQb[2:0]# fields of the FSB**
20:16	ROST	00000	**EXF: EXF[4:0]# fields of the FSB**
15:8	ROST	00h	**ATTR: ATTR[7:0]# fields of the FSB**
7:0	ROST	00h	**DID: DID[7:0]# fields of the FSB**

4.8.16.9 RECFSB[3:0]: Recoverable FSB Error Log Register

The following error log registers captures the FSB fields on the logging of an error in the corresponding FERR_NF_FSB Register.

Device: 17
Function: 0,3
Offset: 48h, C8h

Bit	Attr	Default	Description
31:29	RV	000	*Reserved*
28:24	ROST	00000	**REQA: REQa[4:0]# fields of the FSB**
23:21	ROST	000	**REQB: REQb[2:0]# fields of the FSB**
20:16	ROST	00000	**EXF: EXF[4:0]# fields of the FSB**
15:8	ROST	00h	**ATTR: ATTR[7:0]# fields of the FSB**
7:0	ROST	00h	**DID: DID[7:0]# fields of the FSB**

4.8.16.10 NRECADDRL[3:0]: NonRecoverable FSB Address Low Error Log Register

This register captures the lower 32 bits of the FSB address for non recoverable errors when a fatal error is logged in its corresponding FERR_FAT_FSB Register. This register is only valid for Request FSB Errors.

Device: 16
Function: 0,3
Offset: 4Ch, CCh

Bit	Attr	Default	Description
31:4	ROST	0h	**A31DT4: FSB Address [31:4]**
3	ROST	0	**A3: FSB Address [3]**
2:0	RV	000	*Reserved*

4.8.16.11 NRECADDRH[3:0]: NonRecoverable FSB Address High Error Log Register

This register captures the upper 8 bits of the FSB address for non recoverable errors when a fatal error is logged in its corresponding FERR_FAT_FSB Register. This register is only valid for Request FSB Errors.

Device: 17
Function: 0,3
Offset: D0h, 50h

Bit	Attr	Default	Description
7:0	ROST	00h	**A39DT32: FSB Address [39:32]**

4.8.16.12 EMASK_FSB[3:0]: FSB Error Mask Register

A '0' in any field enables that error.

Device: 17
Function: 0,3
Offset: D2h, 52h

Bit	Attr	Default	Description
15:9	RV	0h	*Reserved*
8	RWST	1	**F9Msk:FSB Protocol Error**
7	RWST	1	**F8Msk: B-INIT**
6	RWST	1	**F7Msk: Detected MCERR**
5	RWST	1	**F6Msk: Data Parity Error**
4	RV	0h	*Reserved*
3	RV	0h	*Reserved*
2	RV	0h	*Reserved*
1	RWST	1	**F2Msk: Unsupported Processor Bus Transaction**
0	RWST	1	**F1Msk: Request/Address Parity Error**

4.8.16.13 ERR2_FSB[3:0]: FSB Error 2 Mask Register

This register enables the signaling of Err[2] when an error flag is set. Note that one and only one error signal should be enabled ERR2_FSB, ERR1_FSB, ERR0_FSB, and MCERR_FSB for each of the corresponding bits.

Device: 17
Function: 0,3
Offset: D8h, 58h

Bit	Attr	Default	Description
15:9	RV	0h	*Reserved*
8	RWST	1	**F9Msk:FSB Protocol Error**
7	RWST	1	**F8Msk: B-INIT**
6	RWST	1	**F7Msk: Detected MCERR**
5	RWST	1	**F6Msk: Data Parity Error**
4	RV	0	*Reserved*
3	RV	0h	*Reserved*
2	RV	0h	*Reserved*
1	RWST	1	**F2Msk: Unsupported Processor Bus Transaction**
0	RWST	1	**F1Msk: Request/Address Parity Error**

4.8.16.14 ERR1_FSB[3:0]: FSB Error 1 Mask Register

This register enables the signaling of Err[1] when an error flag is set. Note that one and only one error signal should be enabled ERR2_FSB, ERR1_FSB, ERR0_FSB, and MCERR_FSB for each of the corresponding bits.

Device: 17
Function: 0,3
Offset: D6h, 56h

Bit	Attr	Default	Description
15:9	RV	0h	*Reserved*
8	RWST	1	**F9Msk:FSB Protocol Error**
7	RWST	1	**F8Msk: B-INIT**
6	RWST	1	**F7Msk: Detected MCERR**
5	RWST	1	**F6Msk: Data Parity Error**
4	RV	0h	*Reserved*
3	RV	0h	*Reserved*
2	RV	0h	*Reserved*
1	RWST	1	**F2Msk: Unsupported Processor Bus Transaction**
0	RWST	1	**F1Msk: Request/Address Parity Error**

4.8.16.15 ERR0_FSB[3:0]: FSB Error 0 Mask Register

This register enables the signaling of Err[0] when an error flag is set. Note that one and only one error signal should be enabled ERR2_FSB, ERR1_FSB, ERR0_FSB, and MCERR_FSB for each of the corresponding bits.

Device: 17
Function: 0,3
Offset: D4h, 54h

Bit	Attr	Default	Description
15:9	RV	0h	*Reserved*
8	RWST	1	**F9Msk:FSB Protocol Error**
7	RWST	1	**F8Msk: B-INIT**
6	RWST	1	**F7Msk: Detected MCERR**
5	RWST	1	**F6Msk: Data Parity Error**
4	RV	0h	*Reserved*
3	RV	0h	*Reserved*
2	RV	0h	*Reserved*
1	RWST	1	**F2Msk: Unsupported Processor Bus Transaction**
0	RWST	1	**F1Msk: Request/Address Parity Error**

4.8.16.16 MCERR_FSB[3:0]: FSB MCERR Mask Register

This register enables the signaling of MCERR when an error flag is set. Note that one and only one error signal should be enabled ERR2_FSB, ERR1_FSB, ERR0_FSB, and MCERR_FSB for each of the corresponding bits.

Note: In the event of error F7, Detected MCERR, the MCH will not re-asserted MCERR on the FSB that was detected as having asserted MCERR.

Device: 17
Function: 0,3
Offset: DAh, 5Ah

Bit	Attr	Default	Description
15:9	RV	0h	*Reserved*
8	RWST	1	**F9Msk:FSB Protocol Error**
7	RWST	1	**F8Msk: B-INIT**
6	RWST	1	**F7Msk: Detected MCERR**
5	RWST	1	**F6Msk: Data Parity Error**
4	RV	0h	*Reserved*
3	RV	0h	*Reserved*
2	RV	0h	*Reserved*
1	RWST	1	**F2Msk: Unsupported Processor Bus Transaction**
0	RWST	1	**F1Msk: Request/Address Parity Error**

4.8.16.17 NRECSF: Non-Recoverable Error Control Information of Snoop Filter

Device: 16
Function: 2
Offset: B0h

Bit	Attr	Default	Description
63:38	RV	0h	*Reserved*
37	ROST	0	**Hit(1), Miss(0)**
36:16	ROST	0h	**Tag(A[39:19])**
15:4	ROST	000h	**Set(A[18:7])**
3	ROST	0	**Interleave(A[6])**
2	ROST	0	**State**
1:0	ROST	00	**Presence Vector**

4.8.16.18 RECSF: Recoverable Error Control Information of Snoop Filter

Device: 16
Function: 2
Offset: B8h

Bit	Attr	Default	Description
63:38	RV	0h	*Reserved*
37	ROST	0	**Hit(1), Miss(0)**
36:16	ROST	0h	**Tag(A[39:19])**
15:4	ROST	000h	**Set(A[18:7])**
3	ROST	0	**Interleave(A[6])**
2	ROST	0	**State**
1:0	ROST	00	**Presence Vector**

4.8.16.19 FERR_FAT_INT: Internal First Fatal Error Register

FERR_FAT _INT latches the first MCH internal fatal error. All subsequent errors get logged in the NERR_FAT_INT.

Device: 16
Function: 2
Offset: C0h

Bit	Attr	Default	Description
7:5	RV	000	*Reserved*
4	RWCST	0	B7Err: Multiple ECC error in any of the ways during SF lookup
3	RWCST	0	B4Err: Virtual Pin Error (VPP_PEX)
2	RWCST	0	B3Err: Coherency Violation Error for EWB
1	RWCST	0	B2Err: Multi-Tag Hit SF
0	RWCST	0	B1Err: DM Parity Error

4.8.16.20 FERR_NF_INT: Internal First Non-Fatal Error Register

Device: 16
Function: 2
Offset: C1h

Bit	Attr	Default	Description
7:3	RV	0h	*Reserved*
2	RWCST	0	**B8Err: SF Coherency Error for BIL (SF)**
1	RWCST	0	**B6Err: Single ECC error on SF lookup (SF)**
0	RWCST	0	**B5Err: Address Map Error (COH)**

4.8.16.21 NERR_FAT_INT: Internal Next Fatal Error Register

Device: 16
Function: 2
Offset: C2h

Bit	Attr	Default	Description
7:5	RV	000	*Reserved*
4	RWCST	0	**B7Err: Multiple ECC error in any of the ways during SF lookup (SF)**
3	RWCST	0	**B4Err: Virtual Pin Port Error (VPP_PEX)**
2	RWCST	0	**B3Err: Coherency Violation Error (COH) for EWB**
1	RWCST	0	**B2Err: Multi-Tag Hit SF (SF)**
0	RWCST	0	**B1Err: DM Parity Error (DM)**

4.8.16.22 NERR_NF_INT: Internal Next Non-Fatal Error Register

Device: 16
Function: 2
Offset: C3h

Bit	Attr	Default	Description
7:3	RV	0h	*Reserved*
2	RWCST	0	**B8Err: SF Coherency Error for BIL (SF)**
1	RWCST	0	**B6Err: Single ECC error on SF lookup (SF)**
0	RWCST	0	**B5Err: Address Map Error (COH)**

4.8.16.23 NRECINT: Non Recoverable Internal Intel® 7300 Chipset Error Log Register

This register will log non-recoverable errors (Fatal and Non Fatal) based on the internal MCH errors that originate from the FERR_FAT_INT, FERR_NF_INT described starting from Section 4.8.16.19, "FERR_FAT_INT: Internal First Fatal Error Register" on page 204. For debugging VPP errors in this register, e.g. if VPP_PEX_PORT1-3 is set, then software can scan the PCI Express configuration space for unit errors logged in the device 2,3 for PEX_UNIT_FERR/NERR register as defined in Section 4.8.14.28, "PEX_UNIT_FERR[7:1]: PCI Express First Unit Error Register" on page 186 to determine the failing port.

Device: 16
Function: 2
Offset: C4h

Bit	Attr	Default	Description
31:21	RV	0h	*Reserved*
20:13	ROST	0h	DM entry
12:11	RV	00	*Reserved*
10:8	ROST	000	**Internal Block that detected the Failure** 001: VPP_PEX_PORT1-3 010: VPP_PEX_PORT4-7 100: COH 101:DM Others: *Reserved* For details on VPP_PEX errors, refer to the FERR/NERR log registers in the PCI Express (Section 4.8.14.28, "PEX_UNIT_FERR[7:1]: PCI Express First Unit Error Register" on page 186).
7	RV	0h	*Reserved*
6:0	ROST	00h	COH Entry of Failed Location

4.8.16.24 RECINT: Recoverable Internal Intel® 7300 Chipset Data Log Register

This register is not currently used as there are no correctable errors in the internal data path of the MCH.

Device: 16
Function: 2
Offset: C8h

Bit	Attr	Default	Description
31:21	RV	0h	*Reserved1*
20:13	ROST	0h	DM entry
12:11	RV	00	*Reserved*
10:8	ROST	000	**Internal Block that detected the Failure** 001: VPP_PEX_PORT2-3 010: VPP_PEX_PORT4-7 100: COH 101:DM Others: *Reserved*
7	RV	0h	*Reserved*
6:0	ROST	00h	COH Entry of Failed Location

4.8.16.25 EMASK_INT: Internal Error Mask Register

A '0' in any field enables that error.

Device: 16 Function: 2 Offset: CCh			
Bit	**Attr**	**Default**	**Description**
7	RWST	1	**B8Msk: SF Coherency Error for BIL**
6	RWST	1	**B7Msk: Multiple ECC error in any of the ways during SF lookup**
5	RWST	1	**B6Msk:Single ECC error on SF lookup**
4	RWST	1	**B5Msk: Address Map Error**
3	RWST	1	**B4Msk: Virtual Pin Port Error**
2	RWST	1	**B3Msk: Coherency Violation Error for EWB**
1	RWST	1	**B2Msk: Multi-Tag Hit SF**
0	RWST	1	**B1Msk: DM Parity Error**

4.8.16.26 ERR2_INT: Internal Error 2 Mask Register

This register enables the signaling of Err[2] when an error flag is set. Note that one and only one error signal should be enabled. Note that one and only one error signal should be enabled in the ERR2_INT, ERR1_INT, ERR0_INT, and MCERR_INT for each of the corresponding bits. A value of '0' enables the error while a value of '1' suppresses it.

Device: 16 Function: 2 Offset: D2h			
Bit	**Attr**	**Default**	**Description**
7	RWST	1	**B8Err2Msk: SF Coherency Error for BIL**
6	RWST	1	**B7Err2Msk: Multiple ECC error in any of the ways during SF lookup**
5	RWST	1	**B6Err2Msk:Single ECC error on SF lookup**
4	RWST	1	**B5Err2Msk: Address Map Error**
3	RWST	1	**B4Err2Msk: SMBus Virtual Pin Error**
2	RWST	1	**B3Err2Msk: Coherency Violation Error for EWB**
1	RWST	1	**B2Err2Msk: Multi-Tag Hit SF**
0	RWST	1	**B1Err2Msk: DM Parity Error**

4.8.16.27 ERR1_INT: Internal Error 1 Mask Register

This register enables the signaling of Err[1] when an error flag is set. Note that one and only one error signal should be enabled. Note that one and only one error signal should be enabled in the ERR2_INT, ERR1_INT, ERR0_INT, and MCERR_INT for each of the corresponding bits. A value of '0' enables the error while a value of '1' suppresses it.

Device: 16
Function: 2
Offset: D1h

Bit	Attr	Default	Description
7	RWST	1	**B8Err1Msk: SF Coherency Error for BIL**
6	RWST	1	**B7Err1Msk: Multiple ECC error in any of the ways during SF lookup**
5	RWST	1	**B6Err1Msk:Single ECC error on SF lookup**
4	RWST	1	**B5Err1Msk: Address Map Error**
3	RWST	1	**B4Err1Msk: SMBus Virtual Pin Error**
2	RWST	1	**B3Err1Msk: Coherency Violation Error**
1	RWST	1	**B2Err1Msk: Multi-Tag Hit SF**
0	RWST	1	**B1Err1Msk: DM Parity Error**

4.8.16.28 ERR0_INT: Internal Error 0 Mask Register

This register enables the signaling of Err[0] when an error flag is set. Note that one and only one error signal should be enabled. Note that one and only one error signal should be enabled in the ERR2_INT, ERR1_INT, ERR0_INT, and MCERR_INT for each of the corresponding bits. A value of '0' enables the error while a value of '1' suppresses it.

Device: 16
Function: 2
Offset: D0h

Bit	Attr	Default	Description
7	RWST	1	**B8Err0Msk: SF Coherency Error for BIL**
6	RWST	1	**B7Err0Msk: Multiple ECC error in any of the ways during SF lookup**
5	RWST	1	**B6Err0Msk:Single ECC error on SF lookup**
4	RWST	1	**B5Err0Msk: Address Map Error**
3	RWST	1	**B4Err0Msk: SMBus Virtual Pin Error**
2	RWST	1	**B3Err0Msk: Coherency Violation Error for EWB**
1	RWST	1	**B2Err0Msk: Multi-Tag Hit SF**
0	RWST	1	**B1Err0Msk: DM Parity Error**

4.8.16.29 MCERR_INT: Internal MCERR Mask Register

This register enables the signaling of MCERR when an error flag is set. Note that one and only one error signal should be enabled. Note that one and only one error signal should be enabled in the ERR2_INT, ERR1_INT, ERR0_INT, and MCERR_INT for each of the corresponding bits. A value of '0' enables the error while a value of '1' suppresses it.

Device: 16
Function: 2
Offset: D3h

Bit	Attr	Default	Description
7	RWST	1	**B8McErrMsk: SF Coherency Error for BIL**
6	RWST	1	**B7McErrMsk: Multiple ECC error in any of the ways during SF lookup**
5	RWST	1	**B6McErrMsk:Single ECC error on SF lookup**
4	RWST	1	**B5McErrMsk: Address Map Error**
3	RWST	1	**B4McErrMsk: SMBus Virtual Pin Error**
2	RWST	1	**B3McErrMsk: Coherency Violation Error for EWB**
1	RWST	1	**B2McErrMsk: Multi-Tag Hit SF**
0	RWST	1	**B1McErrMsk: DM Parity Error**

4.8.17 Debug and Error Injection Registers

4.8.17.1 FEINJCTL[3:0]: FSB Error Injection Control Register

These registers contain the error injection masks to determine which bits get corrupted for error detection testing.

Device: 16
Function: 3, 0
Offset: 49Ch, 19Ch

Bit	Attr	Default	Description
31:20	RV	0h	**Reserved**
19:16	RW	0	**DATAPERR1: Data Parity Error** This bit field selects which parity bit will be inverted when EINJEN is enabled and when the DINJ[1:0] response function is asserted. 1xxx: DP3 x1xx: DP2 xx1x: DP1 xxx1: DP0
15	RW	0	**EINJEN: Error injection enable** 0: Disable error injection 1: Enable error injection
14	RW	0	**EINJFUNCTSEL: Error Injection Function Select** 0: Select DINJ0 response function. 1: Select DINJ1 response function.
13:0	RW	0h	**Unused:** Writing to these bits has no effect.

4.8.17.2 SFERRINJ[1:0]: Snoop Filter (SF) Error Injection Register

Device: 17 Function: 0 Offset: A8h			
Bit	**Attr**	**Default**	**Description**
31:29	RW	0	Reserved
28	RWC	0h	WRMATCH: Indicates hardware has injected an error.
27:16	RW	0h	SET[11:0]: Indicates the set that should be matched against writes.
15	RW	1	ONCE: If Set error is injected only once. 1: Inject one error 0: inject infinite error
14	RW	0	ENABLE: Enables SF tag array error injection.
13	RW	0	ALL: Indicates to corrupt the column indicated in SFERRINJ.COLUMN in EVERY way. 1: Error will be injected in all ways 0: use bit [6:0] of this register to program the WAY[6:0] value for error injection
12:7	RW	000001b	COLUMN[5:0]: Indicates the column to corrupt on every write. Values range from 6'd0->6'd34 inclusive. The injection happens after the redundancy muxes.
6:0	RW	0011100b	WAY[6:0]: Indicates the way to inject errors into

4.8.17.3 ERRINJCON[7:0]: PCI Express Error Injection Control Register

This register controls injection of received (detected) errors for the purposes of testing error signaling hardware and software. It can create the appearance of a detected correctable or uncorrectable error.

Device: 0-7 Function: 0 Offset: F8h			
Bit	**Attr**	**Default**	**Description**
15:3	RV	0h	***Reserved.***
2	RW	0	**Cause a Completion Timeout Error:** When this bit is written, one and only one error assertion pulse is produced on the error source signal. This error will appear equivalent to an actual error assertion because this event is OR'd into the existing error reporting structure. ***Note:*** This bit is used for an uncorrectable error test.
1	RW	0	**Cause a Receiver Error:** When this bit is written, one and only one error assertion pulse is produced on the error source signal. This error will appear equivalent to an actual error assertion because this event is OR'd into the existing error reporting structure. ***Note:*** This bit is used for a correctable error test.
0	RWO	0	**Error Injection Disable:** This bit field disables the use of the PCIe error injection bits (bits [2:1]). This is a write-once bit.

4.8.17.4 PEX[7:0]EINJCTL: PCI Express Outbound Error Injection Control Register

This register contains the error injection mask register to determine which bits get corrupted for error detection testing.

Device: 7-0
Function: 0
Offset: 278h

Bit	Attr	Default	Description
31:25	RV	0h	**Reserved**
24:23	RW	00	**LNSLCT:** Select one of the 4 lanes to inject error on.
22	RW	0	**OTHRINJ:** Inject errors in symbols other than STP, END, SDP 0: Do not inject error on type Others 1: Inject error on type Others when EINJEN is set.
21	RW	0	**SDPINJ:** Inject error in SDP symbol. 0: Do not inject error on SDP 1: Inject error on SDP when EINJEN is set.
20	RW	0	**ENDINJ:** Inject error in a END symbol 0: Do not inject error on END 1: Inject error on END when EINJEN is set.
19	RW	0	**STPINJ:** Inject error in a STP symbol 0: Do not inject error on STP 1: Inject error on STP when EINJEN is set.
18	RW	0	**CRCERRINJ: CRC error injection** Selects which type of injection will be performed when DINJ0,1 is asserted. 0: Data bit corruption. Based ETRANS and EBITPOS values. 1: CRC bit corruption. Based on values in PEX[6:0]EINJMSK.
17	RW	0	**EINJEN: Error injection enable** 0: Disable error injection 1: Enable error injection
16	RW	0	**EINJFUNCTSEL: Error Injection Function Select** 0: Select DINJ0 response function. 1: Select DINJ1 response function.
15:10	RW	0h	**ETRANS: Symbol transfer number** This field indicates which symbol contains the corrupted bit. A maximum depth of 64 is where the corruption could occur.
9:0	RW	0h	**EBITPOS: Error injection bit position** Bit field can contain more than one bit for multi-bit error injection.

4.8.17.5 PEX[7:0]EINJMSK: PEX Outbound Error Injection Mask Register

This register contains the error injection mask register to determine which bits get corrupted for error detection testing.

Device: 7-0
Function: 0
Offset: 27Ch

Bit	Attr	Default	Description
31:0	RW	0h	**CRCMSK: CRC Mask.** A logic '1' in any location inverts the CRC for an outbound transaction.

4.8.17.6 FBD[1:0]PLLCTRL: FBD PLL Control Register

This register defines the ability to control the PLL (on/off) for FBD branches 0 and 1. This is an additional feature besides JTAG providing BIOS an extra degree of control on the FBD branches.

Device: 22, 21
Function: 0
Offset: 1C0h

Bit	Attr	Default	Description
7:1	RV	0	**Reserved**
0	RWST	0h	**DISPLL: Disable PLL for the FDB Branch** 0: Normal operation i.e PLL is enabled and FBD memory can be accessed. 1: Disable PLL for the FBD Branch. This implies that the selected FBD branch is disabled and no operations to memory are possible.

4.8.17.7 FBD[3:2]EINJCTL: FBD Outbound Error Injection Control Register

This register contains the error injection mask register to determine which bits get corrupted for error detection testing.

Device: 22
Function: 0
Offset: 20Ch, 10Ch

Bit	Attr	Default	Description
31:24	RV	0	**Reserved**
23:22	RW	0	**EINJEN: Error injection enable** 00: No error injection 01: Flit transfer error injection 10: Stuck at 1 error injection 11: Reserved
21	RW	0	**EINJSEL: Error Injection Select** 0: Select DINJ0 response function. 1: Select DINJ1 response function.
20:18	RW	0	**DDRCMD: Select the DDR command types** For selecting which transaction to corrupt. FBD Command encoding bit field [20:18]
17:14	RW	0	**FBDCHCMD: Select channel command to corrupt OP[3:0] (Thermal WRite Configs/Updates are excluded from corruption)**
13:10	RW	0h	**FTRANS: Flit transfer number to corrupt** This field indicates which flit contains the corrupted bit. A maximum depth of 12 is where the corruption could occur. 0-Bh: Flit number C-Fh: Reserved
9:0	RW	0h	**EBITPOS: Error injection bit position** Bit field can contain more than one bit for multi-bit error injection.

4.8.17.8 FBD[1:0]EINJCTL: FBD Outbound Error Injection Control Register

This register contains the error injection mask register to determine which bits get corrupted for error detection testing.

Device: 21
Function: 0
Offset: 20Ch, 10Ch

Bit	Attr	Default	Description
31:24	RV	0	**Reserved**
23:22	RW	0	**EINJEN: Error injection enable** 00: No error injection 01: Flit transfer error injection 10: Stuck at 1 error injection 11: Reserved
21	RW	0	**EINJSEL: Error Injection Select** 0: Select DINJ0 response function. 1: Select DINJ1 response function.
20:18	RW	0	**DDRCMD: Select the DDR command types** For selecting which transaction to corrupt. FBD Command encoding bit field [20:18]
17:14	RW	0	**FBDCHCMD: Select channel command to corrupt OP[3:0] (Thermal WRite Configs/Updates are excluded from corruption)**
13:10	RW	0h	**FTRANS: Flit transfer number to corrupt** This field indicates which flit contains the corrupted bit. A maximum depth of 12 is where the corruption could occur. 0-Bh: Flit number C-Fh: Reserved
9:0	RW	0h	**EBITPOS: Error injection bit position** Bit field can contain more than one bit for multi-bit error injection.

4.8.17.9 FBD[3:2]STUCKL: Stuck "ON" FBD Lanes

This register selects FBD Lanes to be stuck at "1" when error injection is enabled and Stuck on Lanes is selected.

Device: 22
Function: 0
Offset: 211h, 111h

Bit	Attr	Default	Description
7:4	RV	0	**Reserved**
3:0	RW	Fh	**SBSTUCK:** SB Lane to be driven to "1" 0h: lane 0 9h: lane 9 and >9h: NOP

4.8.17.10 FBD[1:0]STUCKL: Stuck "ON" FBD Lanes

This register selects FBD Lanes to be stuck at "1" when error injection is enabled and Stuck on Lanes is selected.

Device: 21
Function: 0
Offset: 211h, 111h

Bit	Attr	Default	Description
7:4	RV	0	**Reserved**
3:0	RW	Fh	**SBSTUCK:** SB Lane to be driven to "1" 0h: lane 0 9h: lane 9 and >9h: NOP

4.8.17.11 MEM[1:0]EINJMSK0: Branch Memory Error Injection Mask0 Register

This register contains the error injection mask register to determine which bits get corrupted for error detection testing.

Notes for register usage:

1. In mirror mode both sets of registers MEM[1:0]EINJMSK0 and MEM[1:0]EINJMSK1 must be programmed with the same values.
2. When the first SRAM location and the second SRAM location are the same value, then only the first causes and ECC mask corruption. The second will have no effect.
3. After the error injection feature is activated, the masks for both first and second corruption are consistently active. In other words, there isn't an enable bit for each of them. The bit field is zero for a valid selection. For these features to have an effect, the user is required to make the mask bit field as zero.

Device: 16
Function: 1
Offset: 108h, 100h

Bit	Attr	Default	Description
31:30	RV	00	**Reserved**
29:28	RW	01	**HLINESEL: Half cache line selection** 00: Reserved 01: Selects lower half cache line for error injection on transfer 0 and 1 using First and/or Second device pointers. 10: Selects upper cache line for error injection on transfer 2 and 3 using First and/or Second device pointers. 11: Select both upper and lower cache lines to inject errors based on the First and Second device pointers. The same masks are applied to both halves.
27	RW	0	**EINJEN: Error injection enable** 0: Disable error injection 1: Enable error injection
26	RW	0	**EINJFUNCTSEL: Error Injection Function Select** 0: Select DINJ0 response function. 1: Select DINJ1 response function.
25:10	RW	0h	**XORMSK: XOR mask bit for first device pointer** [25:18]: XOR mask for transfer 1 (lower half cache line) or 3 (upper half cache line). [17:10]: XOR mask for transfer 0 (lower half cache line) or 2 (upper half cache line).
9:5	RW	0h	**SEC2RAM: Second device pointer location [17:0]**

Device: 16 Function: 1 Offset: 108h, 100h			
Bit	**Attr**	**Default**	**Description**
4:0	RW	0h	**FIR2RAM: First device pointer location [17:0]**

4.8.17.12 MEM[1:0]EINJMSK1: Branch Memory Error Injection Mask1 Register

This register contains the error injection mask register to determine which bits get corrupted for error detection testing.

Device: 16 Function: 1 Offset: 106h, 104h			
Bit	**Attr**	**Default**	**Description**
15:0	RW	0h	**XORMSK: XOR mask bit for second device location** [15:8]: XOR mask for transfer 1 (lower half cache line) or 3 (upper half cache line). [7:0]: XOR mask for transfer 0 (lower half cache line) or 2 (upper half cache line).

4.8.17.13 FBD[3:0]CHCFG: Channel Configuration Register

This register controls channel configuration information for TS3 for individual lanes.

Device: 21, 22 Function: 0 Offset: 1D4h, 2D4h			
Bit	**Attr**	**Default**	**Description**
15:9	RV	0h	**Reserved**
8	RWST	0h	**REGVAL1:** Set to '1'.
7:4	RWST	0Fh	**REGVAL2:** Set to the value 'Fh'.
3:0	RWST	0Fh	**REGVAL3:** Set to the value 'Fh'.

4.8.17.14 FBD[3:0]TS1PARM: FBD TS1 Training Set Parameter Register

This register contains the parameter values that are sent as part of the TS1 training set.

Device: 22, 21 Function: 0 Offset: 2B8h, 1B8h			
Bit	**Attr**	**Default**	**Description**
31:24	RV	0	**Reserved**
23:12	RWST	0h	**GRP3PARM: Group 3 parameters** The contents of this bit field is sent as part of the TS1 training set.
11:0	RWST	0h	**GRP2PARM: Group 2 parameters** The contents of this bit field is sent as part of the TS1 training set.

4.8.17.15 DINJ0: Error Injection 0 Response Function

This register and the DINJ1 register determines which event starts and stops the error injection mechanisms distributed throughout the chip. Error injection verifies a downstream component's error detection capabilities. When the start source is asserted the start signal will arrive to all the clusters within a few clocks. An enabled error injector will cause one error each time an opportunity is available, this will vary depending on the interface. Injection continues until the Stop_sel condition is satisfied. If Stop_sel is set to 0x0A, exactly one error will be injected on each interface selected by the mask registers.

Device: 19
Function: 0
Offset: 9Ah

Bit	Attr	Default	Description
7:4	RWST	Ch	**Stop_sel: Response Function Stop Select** 1010: Deactivate one clock after activation. Error injection on one transaction must occur on the interface that is enabled. 1011: Response is never deactivated Other: Reserved
3:0	RWST	Ah	**Start_sel: Response Function Start Select** 1010: Response never occurs 1011: Response occurs immediately Other: Reserved

4.8.17.16 DINJ1: Error Injection 1 Response Function

This register and the DINJ0 register determines which event starts and stops the error injection mechanisms distributed throughout the chip. Error injection verifies a downstream component's error detection capabilities. When the start source is asserted the start signal will arrive to all the clusters within a few clocks. An enabled error injector will cause one error each time an opportunity is available, this will vary depending on the interface. Injection continues until the Stop_sel condition is satisfied. If Stop_sel is set to 0x0A, exactly one error will be injected on each interface selected by the mask registers.

Device: 19 Function: 0 Offset: 9Bh			
Bit	**Attr**	**Default**	**Description**
7:4	RWST	Ch	**Stop_sel: Response Function Stop Select** Selects the source which causes the deactivation of the response in the cluster. 0000: Global Event 0 (GE[0]) 0001: Global Event 1 (GE[1]) 0010: Global Event 2 (GE[2]) 0011: Global Event 3 (GE[3]) 0100: Artificial Starvation Counter 0 (ASC[0]) 0101: Artificial Starvation Counter 1 (ASC[1]) 0110: Artificial Starvation Counter 2 (ASC[2]) 0111: Artificial Starvation Counter 3 (ASC[3]) 1000: Global Trigger 0 1001: Global Trigger 1 1010: Deactivate one clock after activation. Error injection on one transaction must occur on the interface that is enabled. 1011: Response is never deactivated 1100: Deactivate the response when the activating signal becomes false. 1101-1111: Reserved
3:0	RWST	Ah	**Start_sel: Response Function Start Select** Selects the source which causes the activation of the response in the cluster. 0000: Global Event 0 (GE[0]) 0001: Global Event 1 (GE[1]) 0010: Global Event 2 (GE[2]) 0011: Global Event 3 (GE[3]) 0100: Artificial Starvation Counter 0 (ASC[0]) 0101: Artificial Starvation Counter 1 (ASC[1]) 0110: Artificial Starvation Counter 2 (ASC[2]) 0111: Artificial Starvation Counter 3 (ASC[3]) 1000: Global Trigger 0 1001: Global Trigger 1 1010: Response never occurs 1011: Response occurs immediately 1100-1111: Reserved

4.8.18 Memory Control Registers

4.8.18.1 MC: Memory Control Settings

Miscellaneous controls not implemented in other registers. Updating the DIMM Electrical Throttling Limit field in the Memory Controller Settings register, MC.ETHROT, after BOOT time may cause the system to hang. Refer to bits [19:18] in the updated register description below for additional programming details.

Device: 16 Function: 1 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
31	RV	0	**Reserved**
30	RW	0	**RETRY: Retry Enable** '1' = enables retry. '0' = disables retry.
29	RW	0	**FLUSH: Flush Writes** '1' = Flush pending writes to memory before servicing pending reads. '0' = Allow normal operation

Device: 16
Function: 1
Offset: 40h

Bit	Attr	Default	Description
28:25	RW	0h	**BADRAMTH: BADRAM Threshold** Number of consecutive instances of adjacent symbol errors required to mark a bad device in a rank. Number of patrol scrub cycles required to decrement a non-saturated BADCNT. If Software desires to enable the "enhanced mode" and use the BADRAMTH, it needs to set a non-zero value to this register field prior. Otherwise, a value of 0 is considered illegal and memory RAS operations may lead to indeterministic behavior.
24	RW	0	**FBDALIGN: FBD Frame Alignment Mode** '1' = Capped alignment. "Pad" tRRL by one frame after transition from hotplug "Ready" or "Recovery Ready" to "Active" or "Resilver". Do not increase tRRL through fast reset. '0' = Minimized alignment. Minimize tRRL. Allow tRRL to increase through fast reset.
23	RO	0	**SYSADDRFuseVal: System Address fuse value** 0: The MCH will use a 40-bit system address with up to 128GB of physical DDR DIMMs. The MC will use a 40-bit decoder internally. 1: The MCH will use 36-bit system address with up to 64GB of physical DDR DIMMs. The MC will use a 36-bit decoder internally. This capability field can be used by BIOS/Software to determine whether the system is using 40-bits or 36-bits address space and then use it to initialize the "**Enable40bADDR" (bit 22)** field in this register for the appropriate application.
22	RW	0	**Enable40bADDR: Enable 40-bit ADDRess** 0: The MCH will use 36-bit system address with up to 64GB of DDR DIMMs. The MC will be set in the 36-bit address decoder mode. (default) 1: The MCH will use a 40-bit system address with up to 128GB of DDR DIMMs. The MC will be set in the 40-bit address decoder mode. BIOS/Software can program this value as desired for the OEM platforms.
21	RW	0	**INITDONE: Initialization Complete.** This scratch bit communicates software state from the MCH to BIOS. BIOS sets this bit to 1 after initialization of the DRAM memory array is complete. This bit has no effect on the MCH operation.
20	RW	0	**FSMEN: FSM enable.** '1' = Enables operation of DDR protocol. This can be used as a synchronous reset to the FSM. (normal) '0' = Inhibits processing of enqueued transactions. Disables all DRAM accesses which means that the FBD link comes up, trains, goes to L0, sends NOP's, does alerts, syncs, fast resets, AMB configs, etc. but it DOES NOT perform a) memory reads b) memory writes c) refreshes] Not preserved by SAVCFG.
19:18	RV	1h	**ETHROT: DIMM Electrical Throttling Limit** Electrical throttling is required to prevent data corruption by limiting the number of activates within a specific time interval and is enabled by MTR.ETHROTTLE register bit. For each rank in the DIMM, Maximum number of activates is four per sliding electrical throttle window as defined below: 00: 10 clocks (DDR533) 01: 13 clocks (DDR667) (default) 10: reserved 11: 20 clocks (safe/conservative setting) The Memory controller should stop sending more than 4 activates for each sliding electrical throttle window. When the sliding window boundary crosses, the counter is reset and the process repeats. The value of ETHROT should be programmed at BOOT time after memory initialization is done and should not be changed while normal memory traffic is in progress.

Device: 16 Function: 1 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
17	RW	0	**GTW_MODE: Global Throttling window mode** This register field is used to reduce the Global throttling window size for the purposes of debug/validation. 0: Global/open-loop turtling window of 16384*1344 (default, normal working mode). If global throttling is enabled in this normal window, it will be held active for 16 global throttling windows without any DIMM exceeding GBLACT. 1: Global/open-loop throttling window of 4*1344 (debug, validation). If global throttling is enabled in this debug window, it will be held active for 2 global throttling windows without any DIMM exceeding GBLACT.
16	RW	0	**MIRROR: Mirror mode enable** '1' = mirroring enabled '0' = mirroring disabled. FBDHPC.NEXTSTATE defines other characteristics of mirrored mode. The MCH does not support mirroring while sparing is enabled: this bit should not be set if SPCPC.SPAREN is set. Note: When MIRROR mode is enabled, both WAY0 and WAY1 of MIR register should be set to 1. Otherwise, it is a programming error.
15:9	RV	0h	*Reserved1*
8	RW	0	**SCRBALGO: Srub Algorithm for x8 uncorrectable error detection** 0: Normal mode 1: Enhanced mode
7	RW	0	**SCRBEN: Patrol Scrub Enable** 1: Enables patrol scrubbing. 0: Disables patrol scrubbing The scrub engine will start the scrub operations from the beginning to the end of the memory each time the SCRBEN register bit is set. Note that SCRBEN should be disabled during MIR updates.
6	RW	0	**DEMSEN: Demand Scrub Enable** Enables demand scrubbing.
5	RW	0	**ERRDETEN: Error Detection Enable** '1' = Northbound CRC/ECC checking enabled. '0' = Northbound CRC/ECC checking disabled FBD "Alert" detection is disabled, status packets are ignored, northbound error logging and data poisoning are disabled when Northbound CRC/ECC checking is disabled.
4	RW1C	0	**SCRBDONE: Scrub Complete** The scrub unit will set this bit to '1' when it has completed scrubbing the entire memory. Software should poll this bit after setting the Scrub Enable (SCRBEN) bit to determine when the operation has completed. If the Scrub enable bit is cleared midway during the scrub cycle, then the SCRBDONE bit will not be set and the MCH will stop the scrub cycle immediately.
3:0	RV	0h	*Reserved*

4.8.18.2 GBLACT: Global Activation Throttle Register

This register contains the limit for Global Activation throttle control.

Device: 16 Function: 1 Offset: 60h			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0h	**GBLACTLM: Global Activation Throttle Limit** This field controls the activation of Open Loop Thermal Throttling (OLTT) based on the number of activates sampled per DIMM pair on each branch. If the number of activates in a global throttling window[a] exceeds the number indicated by the GBLACTLM field, then OLTT is started. The MCH will set the THRSTS[1:0].GBLTHRT bit for the respective branch and use the THRTMID register setting for throttling. The granularity of this field is 65536 activations. Refer to Table 4-41. If Software sets this value greater than 168, the MCH will cap the GBLACTLM field to 168.

Notes:

a. if (MC.GTW_MODE==1), then the global throttling window is 4*1344 cycles (debug, validation). Else if (MC.GTW_MODE==0), then the window is set to 16384*1344 cycles (normal).

Table 4-41. Global Activation Throttling as a Function of Global Activation Throttling Limit (GBLACTM) and Global Throttling Window Mode (GTW_MODE) Register Fields

GBLACT.GBLACTM Range (0..168)	Number of Activations	
	MC.GTW_MODE=0 (16384*1344 window)	MC.GTW_MODE=1 (4*1344 window)
0	No Throttling (unlimited activations)	No Throttling (unlimited activations)
1	65,536	16
2	131,072	32
16	1,048,576	256
32	2,097,152	512
64	4,194,304	1,024
96	6,291,456	1,536
100	6,553,600	1,600
128	8,388,608	2,048
150	9,830,400	2,400
168	11,010,048 (100% BW)	2,688 (100% BW)

4.8.18.3 THRTSTS[1:0]: Throttling Status Register

This register records the global activation throttle status and internal thermal throttling value for FBD branches 1 and 0.

Device: 16 Function: 1 Offset: 6Ah, 68h			
Bit	**Attr**	**Default**	**Description**
15:9	RV	0h	*Reserved*

Device: 16 Function: 1 Offset: 6Ah, 68h			
Bit	**Attr**	**Default**	**Description**
8	RO	0h	**GBLTHRT: Global Activation Throttle** This field is set by the MCH to indicate the start of Open Loop Thermal Throttling (OLTT) based on the number of activates sampled in the global throttling window. If the number of activates in the global throttling window (16384*1344 cycles) exceeds the number indicated by the GBLACT.GBLACTLM field, then THRTSTS.GBLTHRT bit is set to enable open loop thermal throttling (OLTT). OLTT will remain active until 16 (or 2) global throttling[a] windows in a row have gone by without any DIMM exceeding the GBLACT.GBLACTLM register at which point this register field will be reset.
7:0	RO	0h	**THRMTHRT: Thermal Throttle Value** This field holds the current activation throttling value 0: No throttling (unlimited activation) 1: 4 activations per activation window 2: 8 activations per activation window 168: 672 activations per activation window This field will be set by the MCH and the value of this field will vary between THRTLOW and THRTHI registers based on the throttling.

Notes:

a. If MC.GTW_MODE=1, then the debug mode is enabled and the MCH will use 2 windows for global activation logic to be valid.

4.8.18.4 THRTLOW: Thermal Throttle Low Register

Device: 16 Function: 1 Offset: 64h			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0h	**THRTLOWLM: Thermal Throttle Low Limit** A base throttling level that is applied when the temperature is in the low range (below Tlow) and THRTCTRL.THRMHUNT is set (S-CLTT) and the THRTSTS.GBLTHRT* bit is not set by the MCH. This base throttling is also enabled If THRTCTRL.THRMHUNT = 0 (OLTT) The maximum value this field can be initialized by software is 168 (decimal). This corresponds to 672 activations per thermal throttling window and gives 100% BW. The granularity of this field is 4 activations. 0: No throttling (unlimited activation) 1: 4 activations per activation window 2: 8 activations per activation window 168: 672 activations per activation window If Software sets this value greater than 168, the chipset will cap the THRTLOWLM field to 168.

4.8.18.5 THRTMID: Thermal Throttle Mid Register

Device: 16 Function: 1 Offset: 65h			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0h	**THRTMIDLM: Thermal Throttle Middle Limit** A mid level throttling level that is applied when the temperature is in the middle range (above Tlow but below Tmid) and THRTCTRL.THRMHUNT is set (S-CLTT) or the THRTSTS.GBLTHRT* bit is set by the MCH (OLTT). This is the highest level throttling for OLTT. The maximum value this field can be initialized by software is 168 (decimal). This corresponds to 672 activations per activation window and gives 100% BW. The granularity of this field is 4 activations. 0: No throttling (unlimited activation) 1: 4 activations per activation window 2: 8 activations per activation window 168: 672 activations per activation window If Software sets this value greater than 168, the MCH will cap the THRTMIDLM field to 168. This field should be less than or equal to the THRTLOW.THRTLOWLM.

4.8.18.6 THRTHI: Thermal Throttle High Register

Device: 16 Function: 1 Offset: 66h			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0h	**THRTHILM: Thermal Throttle High Limit** The highest level of throttling (i.e. minimum number of activations) for S-CLTT. This throttling will be enabled if THRTCTRL.THRMHUNT is set and is applied whenever the temperature is above Tmid. The maximum value this field can be initialized by software is 168 (decimal). This corresponds to 672 activations per activation window and gives 100% BW. The granularity of this field is 4 activations. 0: No throttling (unlimited activation) 1: 4 activations per activation window 2: 8 activations per activation window 168: 672 activations per activation window If Software sets this value greater than 168, the chipset will cap the THRTHILM field to 168. This field should be less than or equal to the THRTMID.THRTMIDLM.

4.8.18.7 THRTCTRL: Thermal Throttling Control Register

Device: 16 Function: 1 Offset: 67h			
Bit	**Attr**	**Default**	**Description**
7:2	RV	0h	*Reserved*
1	RW	0	**THRMODE: Thermal Throttle Mode. This bit should be set when in S-CLTT.** 0: Reserved. 1: Set this when using S-CLTT.
0	RW	0	**THRMHUNT: S-CLTT Enable** 0: OLTT mode. THRTSTS.THRMTHRT register is set based on number of activations in a Global throttling window 1: S-CLTT mode. THRTSTS.THRMTHRT register is set based on temperature returned from AMB in the SYNC status packet

4.8.18.8 MCA: Memory Control Settings A

Additional miscellaneous control not reflected in other registers.

Device: 16 Function: 1 Offset: 58h			
Bit	**Attr**	**Default**	**Description**
31:28	RW	0h	**TO: Starvation Timeout** A value of zero represents eight cycles. Each increment adds eight cycles. Maximum is 128 cycles.
27:22	RW	32h	**RQHTH: Deferred Read Queue High Threshold** Deferred read queue sources are disabled when the depth of the deferred read queue meets or exceeds this level. A value of 000000b corresponds to a queue depth of zero. A value of 100000b corresponds to a queue depth of 32.
21:16	RW	2Ch	**RQLTH: Deferred Read Queue Low Threshold** Deferred read queue sources are enabled when the depth of the deferred read queue meets or falls below this level. A value of 000000b corresponds to a queue depth of zero. A value of 100000b corresponds to a queue depth of 32. A default value of 44d (2Ch) is chosen to set the low threshold for the Read request queue. This provides headroom for concurrent FSB and Inbound requests and mitigates possible starvation of the FSB.
15	RV	00	*Reserved*
14	RW	0	**SCHDIMM: Single Channel DIMM Operation** 0: The MC assumes that the MCH is operating normally, i.e. MC is not operating with only one FBD channel as in single channel mode. 1: In this mode, the MCH MC will operate such that only 1 channel (i.e. branch 0, channel 0) is active and there can be one or more DIMMS present in Channel 0.
13:8	RW	15h	**WQHTH: Write Queue High Threshold** Write queue sources are disabled when the depth of the write queue meets or exceeds this level. Writes are preferred over reads when the depth of the write queue meets or exceeds this level. A value of 000000b corresponds to a queue depth of zero. A value of 100000b corresponds to a queue depth of 32.
7:6	RV	00	*Reserved*
5:0	RW	0Dh	**WQLTH: Write Queue Low Threshold** Write queue sources are enabled when the depth of the write queue meets or falls below this level. Reads are preferred over writes when the depth of the write queue meets or falls below this level. A value of 000000b corresponds to a queue depth of zero. A value of 100000b corresponds to a queue depth of 32. A default value of 13d (0Dh) is chosen to set the low threshold for the write request queue.

4.8.18.9 MCCTRL: Memory Controller Control Register

Timing control and other operations for MC.

Device: 16
Function: 1
Offset: 5Ch

Bit	Attr	Default	Description
31:6	RV	0h	*Reserved*
5	RW	0	**IEIDRXTEN: Ignore electrical idle detection for NB calibration pattern detection and enable receiver training logic during calibration** 0: FBD calibration pattern is only detected on northbound receivers that are not in electrical idle, and receiver training logic is disabled during calibration. (default) 1: FBD calibration pattern detection ignores electrical idle detection and depends only on Rx data values. Northbound training logic is enabled during calibration
4	RW	0	**EDRCDIS**: **Disable Edge-off during initialization while enabling DRC for receivers in EI during TS0** This register field provide selective control for FBD I/O reset AFE issues relating to DRC. 0: Assert edge_off to all lanes when the lanes detect EI on their respective receivers (and/or soft_core_reset is asserted). This would disable DRC when the lane is in EI. 1: Disable edge_off assertion to the lanes while each lane is receiving EI. This means that DRC would be enabled while the link is still in EI
3:0	RWO	6h	**EDLD: Early Defer Reply launch delay** The MC will send an early defer reply signal to the FSB cluster "x" clocks before the data arrival from memory. The value "x" defined by this register field is programmed by BIOS to a recommended value of 13 decimal (0xD). Software should ensure that this parameter does not violate the memory round trip and FBD timing and should not be changed dynamically; otherwise data may be corrupted. A value of 0110b (decimal 6) is used as default for the early defer reply launch signal. Maximum value is 15 clocks. This field is of type "Read Write once" and it is recommended that BIOS lock this register if it desires that no other process should modify the field. (for stability)

4.8.18.10 MS: Memory Status

Miscellaneous status not reflected in other registers.

Device: 16
Function: 1
Offset: 44h

Bit	Attr	Default	Description
31:24	RV	00h	***Reserved***
23:17	RO	00h	**IRQD**: In Order Read Queue Depth Number of in order entries in the read queue.
16:10	RO	00h	**DRQD: Deferred Read Queue Depth** Number of deferred entries in the read queue.
9:7	RV	0h	*Reserved*
6:1	RO	00h	**WQD: Write Queue Depth** Number of entries in the write post queue. When this is '0', then the WP bit in this register is also '0'.
0	RO	0	**WP: Writes Posted** '0' = All memory writes have been flushed. The WQD field in this register is "0". '1' = Memory writes are posted

4.8.18.11 DRTA: DRAM Timing Register A

This register defines timing parameters that work with all DDR II SDRAM's in the memory subsystem. The parameters for these devices can be obtained by serial presence detect. This register must be set to provide timings that satisfy the specifications of all SDRAM's detected. E.g., if SDRAM's present have different TRC's, the maximum should be used to program this register. Consult the JEDEC DDR and DDR II SDRAM specifications [7] [8] for the technology of the devices in use.

Device: 16
Function: 1
Offset: 48h

Bit	Attr	Default	Description
31	RV	0	*Reserved*
30:28	RW	0h	**TAL: Additive Latency for Posted CAS** This parameter is the posted-CAS "t_{AL}" DDR II timing parameter. It must match the value written to the EMRS register in the DRAM.
27:22	RW	0h	**TWRC: Activate command to activate command delay following a DDR write** This parameter is the minimum delay from an activate command followed by a write with page-close to another activate command on the same bank. This parameter prevents bank activation protocol violations in the DRAM's. This parameter is defined in core cycles. This parameter is set to greater than or equal to the largest TWRC of any DIMM on the memory sub-system. This parameter is defined as follows: (((CL – 1) + BL/2 +((((t_{RCD} + t_{WR} + t_{RP}) / t_{FBD}) rounded up to the nearest integer)) rounded up to the nearest integer), where t_{RCD} is the DDR ras-to-cas delay, CL is the cas-to-first-read-data latency, BL is the burst length, t_{WR} is the write recovery time, t_{RP} is the precharge time, t_{FBD} is the cycle time of an FBD frame.
21:16	RW	0h	**TRC: Activate command to activate command delay (same bank)** This parameter is the minimum delay from an activate command to another activate or refresh command to the same bank. This parameter ensures that the page of the bank that was opened by the first activate command is closed before the next activate command is issued. This parameter is defined in core cycles. This parameter is set to greater than or equal to the largest t_{RC} of any DIMM on the memory sub-system. This parameter is defined as follows: (((t_{RC} / t_{FBD}) rounded up to the nearest integer) rounded up to the nearest integer) where t_{FBD} is the cycle time of an FBD frame.

Device: 16 Function: 1 Offset: 48h			
Bit	**Attr**	**Default**	**Description**
15:8	RW	00h	**TRFC: Refresh command to activate command delay** This parameter is the minimum delay from a refresh command to another activate or refresh command. This parameter ensures that the banks that were opened by the refresh command are closed before the next activate command is issued. This parameter is defined in core cycles. This parameter is set to greater than or equal to the largest TRFC of any DIMM on the memory sub-system. This parameter is defined as follows: (A * ((t_{RFC} / t_{FBD}) rounded up to the nearest integer) rounded up to the nearest integer) where t_{FBD} is the cycle time of an FBD frame.
7:4	RW	0h	**TRRD: Activate command to activate command delay (different banks)** This parameter is the minimum delay from an activate command to another activate or refresh command to a different bank on the same rank. This parameter ensures that the electrical disturbance to the SDRAM die caused by the first activate has attenuated sufficiently before the next activate is applied. This parameter is defined in core cycles. This parameter is set to greater than or equal to the largest TRRD of any DIMM on the memory sub-system. This parameter is defined as follows: (((t_{RRD} / t_{FBD}) rounded up to the nearest integer) rounded up to the nearest integer) where t_{FBD} is the cycle time of an FBD frame.
3:0	RW	0h	**TREF: Refresh command to Refresh command delay** This parameter specifies the average delay from a refresh command to another refresh command to the same rank over a period of nine refresh intervals (nine TREF's). This parameter ensures that a sufficient number of refreshes per time interval are issued to each rank. DRTA[3:0] selects the number of cycles for the refresh interval (usually 7.8us) divided by 16. For example, the refresh interval for DDR533 and setting 0x6 is 130x16x3.75nS = 7.8uS. 0xF - Reserved 0xE - 2000 0xD - 1500 0xC - 1000 0xB - 500 0xA - 250 0x9 - 195 0x8 - 185 0x7 - 162 0x6 - 130 0x5 - 154 0x4 - 124 0x3 - 97 0x2 - 81 0x1 - 65 0x0 - Refresh disabled

4.8.18.12 DRTB: DDR Timing Register B

This register defines timing parameters that work with all DDR ports in the memory subsystem. This register must be set to provide timings that satisfy the specifications of all detected DDR ports. E.g., if DDR ports have different TR2W's, the maximum should be used to program this register.

Device: 16 Function: 1 Offset: 4Ch			
Bit	**Attr**	**Default**	**Description**
31:19	RV	000h	*Reserved*

Device: 16 Function: 1 Offset: 4Ch			
Bit	**Attr**	**Default**	**Description**
18:16	RW	0h	**TW2RDR: Write command to read command delay, different rank** This parameter is the minimum delay from a write command to a read command on different ranks of the same DIMM. This parameter prevents data strobe protocol violations on the DIMM's DDR data bus. This parameter is defined in core cycles. The formula for this value is ((BL/2 + t_{FRR} - 1) rounded up to the nearest integer). BL is the burst length, t_{FRR} is the turnaround time from write to read on the DIMM in FBD frames.
15:12	RW	0h	**TR2W: Read command to write command delay** This parameter is the minimum delay from a read command to a write command on the same DIMM. This parameter prevents data strobe protocol violations on the DIMM's DDR data bus. This parameter is defined in core cycles. The formula for this value is ((BL/2 + t_{FRR} + 1) rounded up to the nearest integer). BL is the burst length, t_{FRR} is the turnaround time from read to write on the DIMM in FBD frames.
11:8	RW	0h	**TW2R: Write command to read command delay, same rank** This parameter is the minimum delay from a write command to a read command on the same rank. This parameter prevents data strobe protocol violations on the DIMM's DDR data bus. This parameter is defined in core cycles. The formula for this value is ((CL - 1 + BL/2 + max((t_{WTR} / t_{FBD}) rounded up to the nearest integer), t_{FRR})) rounded up to the nearest integer). CL is the cas-to-first-read-data latency, BL is the burst length, t_{WTR} is the internal write to read command delay, t_{FBD} is the cycle time of an FBD frame, t_{FRR} is the turnaround time from write to read on the DIMM in FBD frames.
7:4	RW	0h	**TR2R: Read command to read command delay** This parameter is the minimum delay from a read command to another read command on a different rank of the same DIMM. This parameter prevents data strobe protocol violations on the DIMM's DDR data bus. This parameter is defined in core cycles. The formula for this value is ((BL/2 + t_{FRR}) rounded up to the nearest integer). t_{FRR} is the turnaround time required to read from different ranks on the DIMM in FBD frames, BL is the burst length.
3:0	RW	0h	**TW2W: Write command to write command delay** This parameter is the minimum delay from a write command to another write command on the same DIMM. This parameter prevents data strobe protocol violations on the DIMMs DDR data bus. This parameter is defined in core cycles. The formula for this value is (BL/2 rounded up to the nearest integer). BL is the burst length.

4.8.18.13 ERRPER: Error Period

Non-zero CERRCNT counts are decremented when the error period counter reaches this threshold. The error period counter is cleared on reset or when it reaches this threshold. The error period counter increments every 524,288 cycles. Table 4-42 indicates the timing characteristics of this register:

Table 4-42. Timing Characteristics of ERRPER

Core Frequency	Per Increment	Maximum Period (increment period x 2^32)
333MHz	1.572864ms	78 days, 04 hours, 29 minutes, 59.44105574 seconds
266 MHz	1.966080 ms	97 days, 13 hours, 37 minutes, 29.30131968 seconds

Device: 16 Function: 1 Offset: 50h			
Bit	**Attr**	**Default**	**Description**
31:0	RW	0h	**THRESH: CERRCNT decrement threshold.** A value of 0 prevents decrementing CERRCNT.

Table 4-43. Leaky Bucket Configuration as a Function of DIMM Density and Error Scaling[a]

Max. DIMM Size per slot in the Motherboard	Expected Error Rate (EER) (# of errors per week)	Expected error period (EEP) (# of hrs between successive Errors = 1/EER)	Leaky Bucket Drip rate Scaling (# of hours = 2xEEP)	Intel® 7300 Chipset Setting based on 2xEEP (ERRPER.THRESH) (Decimal)	
				266 MHz	333 MHz
8 GB	8	21	42	76,904,296	96,130,371
4 GB	4	42	84	153,808,593	192,260,742
2 GB	2	84	168	307,617,187	384,521,484
1 GB	1	168	336	615,234,375	769,042,968
512 MB	0.5	336	672	1,230,468,750	1,538,085,937
256 MB	0.25	672	1344	2,460,937,500	3,076,171,875

Notes:

a. If DIMMs of different sizes (densities) are mixed, then the MCH will be programmed to the maximum permissible value. e.g a 4GB DIMM and a 1GB DIMM in the system would imply that BIOS sets ERRPER.THRESH to 0xFFFF_FFFF

4.8.19 Memory Ratio Registers

4.8.19.1 DDRFRQ: DDR Frequency Ratio

This register specifies the CORE:DDR frequency ratio.

Device: 16 Function: 1 Offset: 56h			
Bit	**Attr**	**Default**	**Description**
7:6	RV	00	*Reserved*
5:4	ROST	00h	**NOW: Present CORE:DDR Frequency Ratio** '00' = **1:1**. e.g. if BUSCLK=266 MHz, then DDR=533 MHz. '01' = Reserved '10' = **4:5**. e.g. if BUSCLK=266 MHz, then DDR=667 MHz. '11' = Reserved This field will only specify the relationship between the CORE-domain and FBD-domain clocks. This field does not indicate the frequency of the FBD SCiD link... that is entirely determined by the frequency of the FBDCLK reference clocks. To achieve successful FBD channel initialization, the frequency of the FBDCLK reference clock must match the frequency of the FBD-domain clock. E.g. if the BUSCLK=266 MHz and the NOW field specifies a ratio of 1:1, then FBD channel initialization will succeed with an FBDCLK frequency of 266 MHz.
3	RV	0	*Reserved*

Device: 16 Function: 1 Offset: 56h			
Bit	**Attr**	**Default**	**Description**
2	RWST	1	**COREFREQ: CORE Frequency** This frequency ratio tells the SPD master what divider ratio to employ in order to operate at 100 KHz. '1' = Core is operating at 333MHz. SPD Divider ratio = 3334 '0' = Core is operating at 266 MHz. SPD Divider ratio = 2667
1:0	RWST	00	**NEXT: Future CORE:DDR Frequency Ratio** This frequency ratio will take effect and transfer to the "NOW" field after the next the MCH hard reset. '00' = **1:1**. e.g. if BUSCLK=266 MHz, then DDR=533 MHz. '01' = Reserved '10' = **4:5**. e.g. if BUSCLK=266 MHz, then DDR=667 MHz. '11' = Reserved This field will only set the relationship between the CORE-domain and FBD-domain clocks. This field will not set the frequency of the FBD SCiD link... that is entirely determined by the frequency of the FBDCLK reference clocks. To achieve successful FBD channel initialization, the frequency of the FBDCLK reference clock must match the frequency of the FBD-domain clock. E.g. if the BUSCLK=266 MHz and the NEXT field specifies a ratio of 1:1, then after the next the MCH hard reset, FBD channel initialization will succeed with an FBDCLK frequency of 266 MHz.

4.8.20 Memory Map Registers

4.8.20.1 TOLM: Top Of Low Memory

This register defines the low MMIO gap below 4 GB. See Section 4.8.20.2, "MIR[2:0]: Memory Interleave Range" on page 230.

Whereas the MIR.LIMIT's are adjustable, TOLM establishes the maximum address below 4 GB that should be treated as a memory access. TOLM is defined in a 256 MB boundary.

This register must not be modified while servicing memory requests.

Device: 16 Function: 1 Offset: 6Ch			
Bit	**Attr**	**Default**	**Description**
15:12	RW	1h	**TOLM: Top Of Low Memory** This register defines the maximum DRAM memory address that lies below 4 GB. It does not denote the actual low MMIO gap but the upperbound of the system low memory. Addresses **equal to** or greater than the TOLM, and less than 4 G, are decoded as low MMIO, MMCFG (if map within this range by HECBASE), chipset, interrupt/SMM and firmware as described in the address mapping chapter. All accesses less than the TOLM are treated as DRAM accesses (except for the VGA region when enabled and PAM gaps). Configuration software should set this field either to maximize the amount of memory in the system (same as the top MIR.LIMIT), or to minimize the allocated space for the lower PCI memory (low MMIO) plus 32 MB (chipset/interrupt/SMM and firmware) at a 256MB boundary. This field must be set to at least 1h, for a minimum of 256 MB of DRAM. The smallest gap between TOLM and 4 GB (for low MMIO, MMCFG, chipset, interrupt/SMM and firmware) is 256 MB because the largest value of TOLM is 0Fh. TOLM cannot be set higher than the total amount of physical memory. This field corresponds to A[31:28]. Setting of "1111" corresponds to 3.75 GB DRAM, and so on down to "0001" corresponds to 0.25GB DRAM. "0000" setting is illegal and a programming error.
11:0	RV	000h	*Reserved*

4.8.20.2 MIR[2:0]: Memory Interleave Range

These registers define each memory branch's interleave participation in processor-physical (A) space.

The MC uses all 40 physical address bits for DRAM memory addressing. However, when the most significant bit is set in the LIMIT field of the MIR[2:0] registers defined below, the rest of the bits in the LIMIT field are ignored, and the LIMIT field is interpreted as 512 GB.

The MIR addresses A[39:28] in multiples of 256MB boundaries. The MMIO gap is defined as 4 GB - TOLM or mathematically (10H - TOLM.TOLM)*256MB

Each MIR register defines a range. If the processor-physical address falls in the range defined by an MIR, the "way" fields in that MIR define branch participation in the interleave. The way-sensitive address bit is A[6]. For a MIR to be effective, WAY0 and WAY1 fields can not be set to 00b. Matching addresses participate in the corresponding ways.

In mirror mode, the WAY0 and WAY1 fields in a participating MIR must be set to 11b. All addresses participate in both ways.

Compensation for MMIO gap size is performed by adjusting the limit of each range upward if it is above TOLM as shown in Table 4-44.

MIR updates can only occur in the RESET, READY, FAULT, DISABLED states.

Table 4-44. Interleaving of an Address is Governed by MIR[i] if

Limit with respect to TOLM	Match MIR[i]
if MIR[i].LIMIT[11:0] <= TOLM	then MIR[i].LIMIT[11:0] > A[39:28] >= MIR[i-1][1].LIMIT[11:0]
if MIR[i].LIMIT[11:0] > TOLM > MIR[i-1].LIMIT[11:0]	then MIR[i].LIMIT[11:0] + (10H - TOLM) > A[39:28] >= MIR[i-1][a].LIMIT[11:0]
if MIR[i].LIMIT[11:0] > MIR[i-1].LIMIT[11:0] >= TOLM	then MIR[i].LIMIT[11:0] + (10H - TOLM) > A[39:28] >= MIR[i-1][1].LIMIT[11:0] + (10H - TOLM)

Notes:

a. for MIR[0], MIR[i-1] is defined to be 0

Device: 16
Function: 1
Offset: 88h, 84h, 80h

Bit	Attr	Default	Description
15:4	RW	000h	**LIMIT** This field defines the highest address in the range A[39:28] prior to modification by the TOLM register. Note: The maximum value is 800h (512GB), and the minimum value is 1h (256 MB).
3:2	RV	00	*Reserved*
1	RW	0	**WAY1** Branch 1 participates in this MIR range if this bit is set AND (the way-sensitive address bit is 1b OR **WAY0** of this MIR is 0b OR MC.MIRROR is set).
0	RW	0	**WAY0** Branch 0 participates in this MIR range if this bit is set AND (the way-sensitive address bit is 0b OR **WAY1** of this MIR is 0b OR MC.MIRROR is set).

4.8.20.3 AMIR[2:0]: Adjusted Memory Interleave Range

For the convenience of software which is trying to determine the physical location to which a processor bus address is sent, 16 scratch bits are associated with each MIR.

Device: 16
Function: 1
Offset: 94h, 90h, 8Ch

Bit	Attr	Default	Description
15:0	RW	0000h	**ADJLIMIT: Adjusted MIR Limit**

4.8.21 FBD Error Registers

4.8.21.1 FERR_FAT_FBD

The first fatal error for an FBD branch is flagged in these registers. More than one flag can be set if different errors occur in the same cycle. For prioritization within the same error number, lower-numbered branches have higher priority than higher-numbered branches, and lower-numbered channels have higher priority than higher-numbered channels. For the FBDChan_Indx, Higher-order error bits within the register have higher priority than lower-order bits. The FBDChan_Indx field is not an error.

Device: 16
Function: 1
Offset: 98h

Bit	Attr	Default	Description
31:30	RV	00	*Reserved1*
29:28	RW1CS	00	**FBDChan_Indx: Logs channel for which the highest-order error occurred valid only when M1Err - M3Err is non-zero.** **Logs branch for which the highest-order error occurred valid only when M23Err is nonzero. Only bit 29 is relevant for the branch number.** **Note:** For M23Err set in the FERR_FAT_FBD, FBDChan_indx indicates the FBD *branch* that caused the error. Possible values are: • FBDChan_indx = 0 for branch 0 M23Err error • FBDChan_indx = 2 for branch 1 M23Err error
27:23	RV	0h	*Reserved*
22	RW1CS	0	**M23Err: Non-Redundant Fast Reset Timeout**
21:3	RV	0h	*Reserved*
2	RW1CS	0	**M3Err: >Tmid Thermal event with intelligent throttling disabled**
1	RW1CS	0	**M2Err: Memory or FBD configuration CRC read error**
0	RW1CS	0	**M1Err: Memory Write error on non-redundant retry or FBD configuration Write error on retry** **Note:** Due to the MCH's two-pass replay scheme, it is possible to log an M1Err on a memory write during the first pass, and have the error corrected in the second pass. In this case, the write error would not be fatal. There is no mechanism, however, to determine the pass on which the M1Err's should still be treated as fatal.

4.8.21.2 NERR_FAT_FBD: FBD Next Errors

If an error is already flagged in FERR_FAT_FBD, subsequent and lower-priority fatal errors are logged in NERR_FAT_FBD.

Device: 16 Function: 1 Offset: 9Ch			
Bit	**Attr**	**Default**	**Description**
31:23	RV	0h	*Reserved*
22	RW1CS	0	**M23Err: Non-Redundant Fast Reset Timeout**
21:3	RV	0h	*Reserved*
2	RW1CS	0	**M3Err: >Tmid Thermal event with intelligent throttling disabled**
1	RW1CS	0	**M2Err: Memory or FBD configuration CRC read error**
0	RW1CS	0	**M1Err: Memory Write error on non-redundant retry or FBD configuration Write error on retry** **Note:** Due to the MCH's two-pass replay scheme, it is possible to log an M1Err on a memory write during the first pass, and have the error corrected in the second pass. In this case, the write error would not be fatal. There is no mechanism, however, to determine the pass on which the M1Err's should still be treated as fatal.

4.8.21.3 FERR_NF_FBD: FBD First Errors

The first non-fatal error for an FBD branch is flagged in these registers. More than one flag can be set if different errors occur in the same cycle. For prioritization within the same error number, lower-numbered branches have higher priority than higher-numbered branches, and lower-numbered channels have higher priority than higher-numbered channels. For the FBDChan_Indx, Higher-order error bits within the register have higher priority than lower-order bits. The FBDChan_Indx field is not an error.

Device: 16 Function: 1 Offset: A0h			
Bit	**Attr**	**Default**	**Description**
31:30	RV	00	*Reserved*
29:28	RWCST	00	**FBDChan_Indx: Logs channel in which the highest-order error occurred** Valid only when M4Err - M22Err or M25Err - M28Err is non-zero. The least-significant-bit of this field has no significance for M4Err through M12Err and M17Err through M20Err. The least-significant-bit of this field only bears significance for M13Err through M15Err and M21Err and higher. For any ECC error flag set in FERR_NF_FBD (M4Err through M12Err and M17Err through M20Err) FBDChan_indx indicates the FBD *branch* that caused the error. Possible values are: • FBDChan_indx = 0 for branch 0 ECC error • FBDChan_indx = 2 for branch 1 ECC error
27:26	RV	0h	*Reserved*
25	RV	0	**Reserved**
24	RWCST	0	**M28Err: DIMM-Spare Copy Completed**
23	RWCST	0	**M27Err: DIMM-Spare Copy Initiated**
22	RWCST	0	**M26Err: Redundant Fast Reset Timeout**
21	RWCST	0	**M25Err: Memory Write error on redundant retry**
20	RWCST	0	*Reserved*
19	RV	0	*Reserved*
18	RWCST	0	**M22Err: SPD protocol Error**
17	RWCST	0	**M21Err: FBD Northbound parity error on FBD Sync Status**

Device: 16 Function: 1 Offset: A0h			
Bit	**Attr**	**Default**	**Description**
16	RWCST	0	**M20Err: Correctable Patrol Data ECC**
15	RWCST	0	**M19Err: Correctable Resilver- or Spare-Copy Data ECC**
14	RWCST	0	**M18Err: Correctable Mirrored Demand Data ECC**
13	RWCST	0	**M17Err: Correctable Non-Mirrored Demand Data ECC**
12	RWCST	0	*Reserved*
11	RWCST	0	**M15Err: Memory or FBD configuration CRC read error**
10	RWCST	0	**M14Err: FBD Configuration Write error on first attempt**
9	RWCST	0	**M13Err: Memory Write error on first attempt**
8	RWCST	0	**M12Err: Non-Aliased Uncorrectable Patrol Data ECC**
7	RWCST	0	**M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
6	RWCST	0	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC**
5	RWCST	0	**M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC**
4	RWCST	0	**M8Err: Aliased Uncorrectable Patrol Data ECC**
3	RWCST	0	**M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
2	RWCST	0	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC**
1	RWCST	0	**M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC**
0	RWCST	0	**M4Err: Uncorrectable Data ECC on Replay**

4.8.21.4 NERR_NF_FBD: FBD Next Errors

If an error is already flagged in FERR_NF_FBD, subsequent and lower-priority non-fatal errors are logged in NERR_NF_FBD. There are cases where alerts on the FBD due to south bound CRC errors leads to spurious NERR logs. The possible spurious NERR logs vary with the error logged in the FBD_NF_FERR, and are listed in Table 4-45

Table 4-45. Possible spurious FBD_NF_NERR logs due to South Bound CRC Errors

Error logged in FBD_NF_NERR	Potential spurious FBD_NF_NERR log	Comment
M13Err	M15Err	
M14Err	M15Err	
M15Err	M15Err	Patrol scrub disabled
M15Err	M13Err, M14Err, M15Err	Patrol scrub enabled

.

Device: 16 Function: 1 Offset: A4h			
Bit	**Attr**	**Default**	**Description**
31:26	RV	00h	**Reserved**
25	RV	0	**Reserved**
24	RWCST	0	**M28Err: DIMM-Spare Copy Completed**

Device: 16 Function: 1 Offset: A4h			
Bit	**Attr**	**Default**	**Description**
23	RWCST	0	**M27Err: DIMM-Spare Copy Initiated**
22	RWCST	0	**M26Err: Redundant Fast Reset Timeout**
21	RWCST	0	**M25Err: Memory Write error on redundant retry**
20	RWCST	0	*Reserved*
19	RV	0	*Reserved*
18	RWCST	0	**M22Err: SPD protocol Error**
17	RWCST	0	**M21Err: FBD Northbound parity error on FBD Sync Status**
16	RWCST	0	**M20Err: Correctable Patrol Data ECC**
15	RWCST	0	**M19Err: Correctable Resilver- or Spare-Copy Data ECC**
14	RWCST	0	**M18Err: Correctable Mirrored Demand Data ECC**
13	RWCST	0	**M17Err: Correctable Non-Mirrored Demand Data ECC**
12	RWCST	0	*Reserved*
11	RWCST	0	**M15Err: Memory or FBD configuration CRC read error**
10	RWCST	0	**M14Err: FBD Configuration Write error on first attempt**
9	RWCST	0	**M13Err: Memory Write error on first attempt**
8	RWCST	0	**M12Err: Non-Aliased Uncorrectable Patrol Data ECC**
7	RWCST	0	**M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
6	RWCST	0	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC**
5	RWCST	0	**M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC**
4	RWCST	0	**M8Err: Aliased Uncorrectable Patrol Data ECC**
3	RWCST	0	**M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
2	RWCST	0	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC**
1	RWCST	0	**M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC**
0	RWCST	0	**M4Err: Uncorrectable Data ECC on Replay**

4.8.21.5 EMASK_FBD: FBD Error Mask Register

A '0' in any field enables that error.

Device: 16 Function: 1 Offset: A8h			
Bit	**Attr**	**Default**	**Description**
31:29	RV	00h	**Reserved**
28	RV	0	**Reserved**
27	RWST	1	**M28Err: DIMM-Spare Copy Completed**
26	RWST	1	**M27Err: DIMM-Spare Copy Initiated**
25	RWST	1	**M26Err: Redundant Fast Reset Timeout**
24	RWST	1	**M25Err: Memory Write error on redundant retry**
23	RWST	1	*Reserved*
22	RWST	1	**M23Err: Non-Redundant Fast Reset Timeout**
21	RWST	1	**M22Err: SPD protocol Error**

Device: 16 Function: 1 Offset: A8h			
Bit	**Attr**	**Default**	**Description**
20	RWST	1	**M21Err: FBD Northbound parity error on FBD Sync Status**
19	RWST	1	**M20Err: Correctable Patrol Data ECC**
18	RWST	1	**M19Err: Correctable Resilver- or Spare-Copy Data ECC**
17	RWST	1	**M18Err: Correctable Mirrored Demand Data ECC**
16	RWST	1	**M17Err: Correctable Non-Mirrored Demand Data ECC**
15	RWST	1	*Reserved*
14	RWST	1	**M15Err: Memory or FBD configuration CRC read error**
13	RWST	1	**M14Err: FBD Configuration Write error on first attempt**
12	RWST	1	**M13Err: Memory Write error on first attempt**
11	RWST	1	**M12Err: Non-Aliased Uncorrectable Patrol Data ECC**
10	RWST	1	**M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
9	RWST	1	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC**
8	RWST	1	**M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC**
7	RWST	1	**M8Err: Aliased Uncorrectable Patrol Data ECC**
6	RWST	1	**M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
5	RWST	1	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC**
4	RWST	1	**M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC**
3	RWST	1	**M4Err: Uncorrectable Data ECC on Replay**
2	RWST	1	**M3Err: >Tmid Thermal event with intelligent throttling disabled**
1	RWST	1	**M2Err: Memory or FBD configuration CRC read error**
0	RWST	1	**M1Err: Memory Write error on non-redundant retry or FBD configuration Write error on retry**

4.8.21.6 ERR0_FBD: FBD Error 0 Mask Register

A '0' in any field enables that error. This register enables the signaling of Err[0] when an error flag is set.

Device: 16 Function: 1 Offset: ACh			
Bit	**Attr**	**Default**	**Description**
31:29	RV	0h	**Reserved**
28	RV	0	**Reserved**
27	RWST	1	**M28Err: DIMM-Spare Copy Completed**
26	RWST	1	**M27Err: DIMM-Spare Copy Initiated**
25	RWST	1	**M26Err: Redundant Fast Reset Timeout**
24	RWST	1	**M25Err: Memory Write error on redundant retry**
23	RWST	1	*Reserved*
22	RWST	1	**M23Err: Non-Redundant Fast Reset Timeout**
21	RWST	1	**M22Err: SPD protocol Error**
20	RWST	1	**M21Err: FBD Northbound parity error on FBD Sync Status**
19	RWST	1	**M20Err: Correctable Patrol Data ECC**

Device: 16 Function: 1 Offset: ACh			
Bit	**Attr**	**Default**	**Description**
18	RWST	1	**M19Err: Correctable Resilver- or Spare-Copy Data ECC**
17	RWST	1	**M18Err: Correctable Mirrored Demand Data ECC**
16	RWST	1	**M17Err: Correctable Non-Mirrored Demand Data ECC**
15	RWST	1	*Reserved*
14	RWST	1	**M15Err: Memory or FBD configuration CRC read error**
13	RWST	1	**M14Err: FBD Configuration Write error on first attempt**
12	RWST	1	**M13Err: Memory Write error on first attempt**
11	RWST	1	**M12Err: Non-Aliased Uncorrectable Patrol Data ECC**
10	RWST	1	**M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
9	RWST	1	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC**
8	RWST	1	**M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC**
7	RWST	1	**M8Err: Aliased Uncorrectable Patrol Data ECC**
6	RWST	1	**M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
5	RWST	1	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC**
4	RWST	1	**M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC**
3	RWST	1	**M4Err: Uncorrectable Data ECC on Replay**
2	RWST	1	**M3Err: >Tmid Thermal event with intelligent throttling disabled**
1	RWST	1	**M2Err: Memory or FBD configuration CRC read error**
0	RWST	1	**M1Err: Memory Write error on non-redundant retry or FBD configuration Write error on retry**

4.8.21.7 ERR1_FBD: FBD Error 1 Mask Register

A '0' in any field enables that error. This register enables the signaling of Err[1] when an error flag is set.

Device: 16 Function: 1 Offset: B0h			
Bit	**Attr**	**Default**	**Description**
31:29	RV	00h	**Reserved**
28	RV	0	**Reserved**
27	RWST	1	**M28Err: DIMM-Spare Copy Completed**
26	RWST	1	**M27Err: DIMM-Spare Copy Initiated**
25	RWST	1	**M26Err: Redundant Fast Reset Timeout**
24	RWST	1	**M25Err: Memory Write error on redundant retry**
23	RWST	1	*Reserved*
22	RWST	1	**M23Err: Non-Redundant Fast Reset Timeout**
21	RWST	1	**M22Err: SPD protocol Error**
20	RWST	1	**M21Err: FBD Northbound parity error on FBD Sync Status**
19	RWST	1	**M20Err: Correctable Patrol Data ECC**
18	RWST	1	**M19Err: Correctable Resilver- or Spare-Copy Data ECC**
17	RWST	1	**M18Err: Correctable Mirrored Demand Data ECC**

Device: 16 Function: 1 Offset: B0h			
Bit	**Attr**	**Default**	**Description**
16	RWST	1	**M17Err: Correctable Non-Mirrored Demand Data ECC**
15	RWST	1	*Reserved*
14	RWST	1	**M15Err: Memory or FBD configuration CRC read error**
13	RWST	1	**M14Err: FBD Configuration Write error on first attempt**
12	RWST	1	**M13Err: Memory Write error on first attempt**
11	RWST	1	**M12Err: Non-Aliased Uncorrectable Patrol Data ECC**
10	RWST	1	**M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
9	RWST	1	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC**
8	RWST	1	**M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC**
7	RWST	1	**M8Err: Aliased Uncorrectable Patrol Data ECC**
6	RWST	1	**M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
5	RWST	1	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC**
4	RWST	1	**M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC**
3	RWST	1	**M4Err: Uncorrectable Data ECC on Replay**
2	RWST	1	**M3Err: >Tmid Thermal event with intelligent throttling disabled**
1	RWST	1	**M2Err: Memory or FBD configuration CRC read error**
0	RWST	1	**M1Err: Memory Write error on non-redundant retry or FBD configuration Write error on retry**

4.8.21.8 ERR2_FBD: FBD Error 2 Mask Register

A '0' in any field enables that error. This register enables the signaling of Err[2] when an error flag is set.

Device: 16 Function: 1 Offset: B4h			
Bit	**Attr**	**Default**	**Description**
31:29	RV	00h	**Reserved**
28	RV	0	**Reserved**
27	RWST	1	**M28Err: DIMM-Spare Copy Completed**
26	RWST	1	**M27Err: DIMM-Spare Copy Initiated**
25	RWST	1	**M26Err: Redundant Fast Reset Timeout**
24	RWST	1	**M25Err: Memory Write error on redundant retry**
23	RWST	1	*Reserved*
22	RWST	1	**M23Err: Non-Redundant Fast Reset Timeout**
21	RWST	1	**M22Err: SPD protocol Error**
20	RWST	1	**M21Err: FBD Northbound parity error on FBD Sync Status**
19	RWST	1	**M20Err: Correctable Patrol Data ECC**
18	RWST	1	**M19Err: Correctable Resilver- or Spare-Copy Data ECC**
17	RWST	1	**M18Err: Correctable Mirrored Demand Data ECC**
16	RWST	1	**M17Err: Correctable Non-Mirrored Demand Data ECC**
15	RWST	1	*Reserved*

Device: 16 Function: 1 Offset: B4h			
Bit	**Attr**	**Default**	**Description**
14	RWST	1	**M15Err: Memory or FBD configuration CRC read error**
13	RWST	1	**M14Err: FBD Configuration Write error on first attempt**
12	RWST	1	**M13Err: Memory Write error on first attempt**
11	RWST	1	**M12Err: Non-Aliased Uncorrectable Patrol Data ECC**
10	RWST	1	**M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
9	RWST	1	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC**
8	RWST	1	**M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC**
7	RWST	1	**M8Err: Aliased Uncorrectable Patrol Data ECC**
6	RWST	1	**M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
5	RWST	1	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC**
4	RWST	1	**M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC**
3	RWST	1	**M4Err: Uncorrectable Data ECC on Replay**
2	RWST	1	**M3Err: >Tmid Thermal event with intelligent throttling disabled**
1	RWST	1	**M2Err: Memory or FBD configuration CRC read error**
0	RWST	1	**M1Err: Memory Write error on non-redundant retry or FBD configuration Write error on retry**

4.8.21.9 MCERR_FBD: FBD MCERR Mask Register

A ‘0’ in any field enables that error. This register enables the signaling of MCERR when an error flag is set.

Device: 16 Function: 1 Offset: B8h			
Bit	**Attr**	**Default**	**Description**
31:29	RV	0h	**Reserved**
28	RV	0	**Reserved**
27	RWST	1	**M28Err: DIMM-Spare Copy Completed**
26	RWST	1	**M27Err: DIMM-Spare Copy Initiated**
25	RWST	1	**M26Err: Redundant Fast Reset Timeout**
24	RWST	1	**M25Err: Memory Write error on redundant retry**
23	RWST	1	*Reserved*
22	RWST	1	**M23Err: Non-Redundant Fast Reset Timeout**
21	RWST	1	**M22Err: SPD protocol Error**
20	RWST	1	**M21Err: FBD Northbound parity error on FBD Sync Status**
19	RWST	1	**M20Err: Correctable Patrol Data ECC**
18	RWST	1	**M19Err: Correctable Resilver- or Spare-Copy Data ECC**
17	RWST	1	**M18Err: Correctable Mirrored Demand Data ECC**
16	RWST	1	**M17Err: Correctable Non-Mirrored Demand Data ECC**
15	RWST	1	*Reserved*
14	RWST	1	**M15Err: Memory or FBD configuration CRC read error**
13	RWST	1	**M14Err: FBD Configuration Write error on first attempt**

Device: 16 Function: 1 Offset: B8h			
Bit	**Attr**	**Default**	**Description**
12	RWST	1	**M13Err: Memory Write error on first attempt**
11	RWST	1	**M12Err: Non-Aliased Uncorrectable Patrol Data ECC**
10	RWST	1	**M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
9	RWST	1	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC**
8	RWST	1	**M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC**
7	RWST	1	**M8Err: Aliased Uncorrectable Patrol Data ECC**
6	RWST	1	**M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC**
5	RWST	1	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC**
4	RWST	1	**M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC**
3	RWST	1	**M4Err: Uncorrectable Data ECC on Replay**
2	RWST	1	**M3Err: >Tmid Thermal event with intelligent throttling disabled**
1	RWST	1	**M2Err: Memory or FBD configuration CRC read error**
0	RWST	1	**M1Err: Memory Write error on non-redundant retry or FBD configuration Write error on retry**

4.8.22 FBD Error Log Registers

4.8.22.1 NRECMEMA: Non-Recoverable Memory Error Log Register A

This register latches information on a fatal or non-recoverable uncorrectable memory error (M1Err, M2Err, or M4Err).

Device: 16 Function: 1 Offset: BEh			
Bit	**Attr**	**Default**	**Description**
15	RV	00	*Reserved*
14:12	ROST	0h	**BANK: Bank of the failed request**
11:8	ROST	0h	**RANK: Rank of the failed request**
7:0	ROST	00h	**REC_FBD_DM_BUF_ID: DM Buffer ID of the failed request**

4.8.22.2 NRECMEMB: Non-Recoverable Memory Error Log Register B

This register latches information on the first detected fatal or non-recoverable uncorrectable memory error (M1Err, M2Err, or M4Err).

Device: 16 Function: 1 Offset: C0h			
Bit	**Attr**	**Default**	**Description**
31	ROST	0	**RDWR** '0' = Read '1' = Write
30:29	RV	0h	**Reserved**

Device: 16 Function: 1 Offset: C0h			
Bit	**Attr**	**Default**	**Description**
28:16	ROST	000h	**CAS:** CAS address of the failed request The CAS address will map from 12:0 while bit 10 (autoprecharge) is hardwired to 0.
15:0	ROST	0h	**RAS:** RAS address of the failed request

4.8.22.3 NRECCFGLOG: Non-Recoverable DIMM Configuration Access Error Log Register

This register latches information on the first detected fatal DIMM configuration register access.

Device: 16 Function: 1 Offset: 74h			
Bit	**Attr**	**Default**	**Description**
31:29	RV	0h	*Reserved1*
27:24	ROST	0h	**BE: Byte Enables of the failed request**
23:16	ROST	00h	**REG: Register Address of the failed request**
15:12	RV	0h	*Reserved*
11	ROST	0	**RDWR** '0' = Read '1' = Write
10:8	ROST	0h	**FUNCTION: Function Number of the failed request**
7:0	ROST	00h	**CFG_FBD_DM_BUF_ID: DM Buffer ID of the failed request**

4.8.22.4 NRECFBDA: Non-Recoverable FBD Error Log Register A

The NRECFBD registers defined below (A through E) has data, ECC and CRC interlaced as defined in the FB-DIMM Architecture and Protocol Specification, Rev. 0.8. This register latches information on the first detected fatal northbound CRC error.

Device: 16 Function: 1 Offset: C4h			
Bit	**Attr**	**Default**	**Description**
31:0	ROST	0h	**BITS: Bits [31:0] of the packet**

4.8.22.5 NRECFBDB: Non-Recoverable FBD Error Log Register B

This register latches information on the first detected fatal northbound CRC error.

Device: 16 Function: 1 Offset: C8h			
Bit	**Attr**	**Default**	**Description**
31:0	ROST	0h	**BITS: Bits [63:32] of the packet**

4.8.22.6 NRECFBDC: Non-Recoverable FBD Error Log Register C

This register latches information on the first detected fatal northbound CRC error.

Device: 16 Function: 1 Offset: CCh			
Bit	**Attr**	**Default**	**Description**
31:0	ROST	0h	**BITS: Bits [95:64] of the packet**

4.8.22.7 NRECFBDD: Non-Recoverable FBD Error Log Register D

This register latches information on the first detected fatal northbound CRC error.

Device: 16 Function: 1 Offset: D0h			
Bit	**Attr**	**Default**	**Description**
31:0	ROST	0h	**BITS: Bits [127:96] of the packet**

4.8.22.8 NRECFBDE: Non-Recoverable FBD Error Log Register E

This register latches information on the first detected fatal northbound CRC error.

Device: 16 Function: 1 Offset: D4h			
Bit	**Attr**	**Default**	**Description**
31:28	RV	0	**Reserved**
27:0	ROST	0h	**BITS: Bits [155:128] of the packet**

4.8.22.9 NRECFBDF: Recoverable FBD Error Log Register F

This register latches information on the first detected non-fatal northbound CRC error.

Device: 16 Function: 1 Offset: D8h			
Bit	**Attr**	**Default**	**Description**
15:12	RV	0h	**Reserved**
11:0	ROST	0h	**BITS: Bits [167:156] of the packet**

4.8.22.10 REDMEMA: Recoverable Memory Data Error Log Register A

This register latches information on the first detected correctable ECC error.

Device: 16 Function: 1 Offset: DCh			
Bit	**Attr**	**Default**	**Description**
31:0	ROST	0h	**SYNDROME**

4.8.22.11 REDMEMB: Recoverable Memory Data Error Log Register B

This register latches information on the first detected ECC error. Note that the ECC_locator field in this register is valid only when the corresponding FERR_NF_FBD register bits are set.

Device: 16 Function: 1 Offset: 7Ch			
Bit	**Attr**	**Default**	**Description**
31:18	RV	0	*Reserved*
17:0	ROST	0h	**ECC _Locator:** identifies the adjacent symbol pair in error for correctable errors. Refer to Table 4-46

Table 4-46. ECC Locator Mapping Information

Symbols	Locator Bit
DS[1:0]	0
DS[3:2]	1
DS[5:4]	2
DS[7:6]	3
DS[9:8]	4
DS[11:10]	5
DS[13:12]	6
DS[15:14]	7
CS[1:0]	8
DS[17:16]	9
DS[19:18]	10
DS[21:20]	11
DS[23:22]	12
DS[25:24]	13
DS[27:26]	14
DS[29:28]	15
DS[31:30]	16
CS[3:2]	17

4.8.22.12 RECMEMA: Recoverable Memory Error Log Register A

This register latches information on the first detected non-fatal memory error.

Device: 16 Function: 1 Offset: E0h			
Bit	**Attr**	**Default**	**Description**
15	RV	0	*Reserved*
14:12	ROST	0h	**BANK: Bank of the failed request**
11:8	ROST	0h	**RANK: Rank of the failed request**
7:0	ROST	00h	**REC_FBD_DM_BUF_ID: DM Buffer ID of the failed request**

4.8.22.13 RECMEMB: Recoverable Memory Error Log Register B

This register latches information on the first detected non-fatal memory error.

Device: 16 Function: 1 Offset: E4h			
Bit	**Attr**	**Default**	**Description**
31	ROST	0	**RDWR** '0' = Read '1' = Write
30:29	RV	00	*Reserved*
28:16	ROST	000h	**CAS: CAS address of the failed request** The CAS address will map from 12:0 while bit 10 (autoprecharge) is hardwired to 0.
15:0	ROST	0h	**RAS: RAS address of the failed request**

4.8.22.14 RECCFGLOG: Recoverable DIMM Configuration Access Error Log Register

This register latches information on the first detected non-fatal DIMM configuration register access.

Device: 16 Function: 1 Offset: 78h			
Bit	**Attr**	**Default**	**Description**
31	ROST	0h	**RDWR**: A value of '1' indicates the failed request was a write. A value of '0' indicates the failed request was a read.
30:28	RV	0h	***Reserved***
27:24	ROST	0h	**BE:** Byte Enables of the failed request
23:16	ROST	00h	**REG:** Register Address of the failed request
15:12	RV	0h	***Reserved***
11	ROST	0	**RDWR** '0' = Read '1' = Write
10:8	ROST	0h	**FUNCTION:** Function Number of the failed request
7:0	ROST	00h	**CFG_FBD_CE_BUF_ID:** DM Buffer ID of the failed request

4.8.22.15 RECFBDA: Recoverable FBD Error Log Register A

This register latches information on the first northbound CRC error.The RECFBD registers defined below (A through F) have the following mapping:

Table 4-47. RECFBD Mapping Information

Bits	Description
167:144	**CRC**
143:128	**ECC**
127:0	**DATA**

Device: 16
Function: 1
Offset: E8h

Bit	Attr	Default	Description
31:0	ROST	0h	**BITS: Bits [31:0] of the packet**

4.8.22.16 RECFBDB: Recoverable FBD Error Log Register B

This register latches information on the first detected non-fatal northbound CRC error.

Device: 16
Function: 1
Offset: ECh

Bit	Attr	Default	Description
31:0	ROST	0h	**BITS: Bits [63:32] of the packet**

4.8.22.17 RECFBDC: Recoverable FBD Error Log Register C

This register latches information on the first detected non-fatal northbound CRC error.

Device: 16
Function: 1
Offset: F0h

Bit	Attr	Default	Description
31:0	ROST	0h	**BITS: Bits [95:64] of the packet**

4.8.22.18 RECFBDD: Recoverable FBD Error Log Register D

This register latches information on the first detected non-fatal northbound CRC error.

Device: 16 Function: 1 Offset: F4h			
Bit	**Attr**	**Default**	**Description**
31:0	ROST	0h	**BITS: Bits [127:96] of the packet**

4.8.22.19 RECFBDE: Recoverable FBD Error Log Register E

This register latches information on the first detected non-fatal northbound CRC error.

Device: 16 Function: 1 Offset: F8h			
Bit	**Attr**	**Default**	**Description**
31:28	RV	0h	**Reserved**
27:0	ROST	0h	**BITS: Bits [155:128] of the packet**

4.8.22.20 RECFBDF: Recoverable FBD Error Log Register F

This register latches information on the first detected non-fatal northbound CRC error.

Device: 16 Function: 1 Offset: FCh			
Bit	**Attr**	**Default**	**Description**
15:12	RV	0h	**Reserved**
11:0	ROST	0h	**BITS: Bits [167:156] of the packet**

4.8.23 FBD Branch Registers

There are two sets of the following registers, one set for each FBD branch. They each appear in function 0 of different devices.

4.8.23.1 FBDLVL[1:0][1:0]: FBD Packet Levelization

This register controls the FBD channel delays.

Device: 22, 21 Function: 0 Offset: 45h, 44h			
Bit	**Attr**	**Default**	**Description**
7:6	RV	00	*Reserved*
5:0	RO	0h	**TRRL: Read Round-Trip Latency** Measured in core cycles from issue of the FBD channel's southbound TS2 packet header to the arrival of it's northbound response header.

4.8.23.2 FBDHPC[1:0]: FBD State Control

This register controls the FBD channel for Initialization. It consists of a next State field.

The index in FBDHPC[index] associates the FBDHPC with branch[index]. FBDHPC[0] is associated with FBD branch 0, FBDHPC[1] is associated with FBD branch 1.

When software writes to FBDHPC[x].NEXTSTATE, the transition will take effect on one or both channels within the branch depending on whether the branch is operating is single- or dual-channel mode.

When Intel® 7300 Chipset hardware transitions FBDST.STATE with the following encodings: 1) disabled, 2) redundant, 3) recovery failed, 4) redundancy loss, and 5) reset, it will transition states of one or both channels within the same branch depending on whether the branch is operating in single- or dual-channel mode.

Device[a]: 22, 21
Function: 0
Offset: 4Fh

Bit	Attr	Default	Description
7:0	RW	00h	**NEXTSTATE: FBD Branch State Control field** This field is written by software to change the branch state. It returns the last value written when read. Some states can only be entered under hardware control and should not be written by software. 00h: Reset 10h: Init 20h: **Ready**[b] 30h: **Active**[b] 40h: **Redundant**[b] 50h: Disabled 60h: **Redundancy Loss**[b] - may not be written D0h: Fault

Notes:

a. The nomenclature is Device 22 (branch 1), 21 (branch 0)
b. Both sync and refresh packets are sent during this mode.

4.8.23.3 FBDST[1:0]: FBD Status

These registers are inspected by software to determine the current FBD branch state. This register contains Initialization state.

The indexing scheme is the same as in FBDHPC registers. The current FBD branch state field indicates state for one or both channels within the same branch depending on whether the branch is operating in single- or dual-channel mode.

Device[a]: 22, 21
Function: 0
Offset: 4Bh

Bit	Attr	Default	Description
7:0	RO	00h	**STATE: FBD Branch State** This field describes the current state of the FBD branch. It can be read by software to determine which FBD branch is being sequenced through recovery, and how far the FBD branch has progressed. 00h: Reset 10h: Init 20h: Ready 30h: Active 40h: Redundant 50h: Disabled 60h: Redundancy Loss - may not be written D0h: Fault

Notes:

a. The nomenclature is Device 22 (branch 1), 21 (branch 0)

4.8.23.4 FBDRST[1:0]: FBD Reset

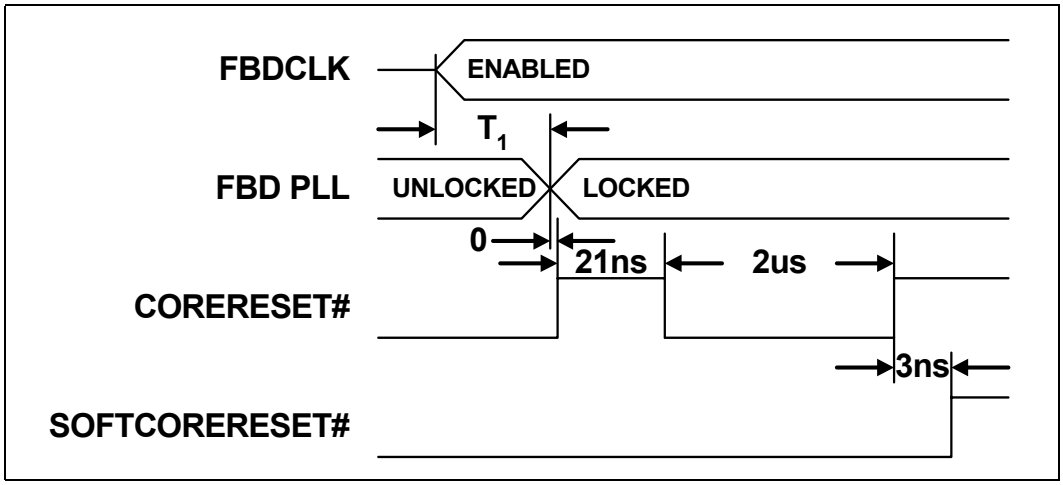
The FBD I/O blocks are reset separately from the rest of Intel® 7300 Chipset. These blocks, composed of FAST-clocked (GHz unit-interval clocked) logic, are supplied by a PLL whose FBDCLK is not available when PWRGOOD is asserted. After FBDCLK is enabled and the FBD PLL has acquired lock, CORERESET# is de-asserted for a minimum of 21 ns, then asserted for a minimum of 2 µs. After the 2 µs assertion, CORERESET# is de-asserted followed by a minimum delay of 3 ns at which time SOFTCORERESET# is de-asserted. If the platform removes FBDCLK on a hot-remove of the branch, CORERESET# and SOFTCORERESET# must be asserted prior to loss of FBDCLK.

A “disabled” (not enabled) FBDCLK is floated at the source, and pulled to ground through the termination near the receiver in the MCH.

All timing specifications in Figure 4-6 are minima.

After the sequence Figure 4-6 has been executed, the FBD branch is ready for initialization.

Figure 4-6. FBD Reset Timing



Device[a]: 22, 21
Function: 0
Offset: 53h

Bit	Attr	Default	Description
7:3	RV	00h	*Reserved*
2	RWST	0	**BRSELCMPRESET: Branch Select for Compensation Reset** 0: COMPreset is tied to CORERESET# from branch 0 1: COMPreset is tied to CORERESET# from branch 1 For Branch 1 to be selected for reset, this field has to be a '1' for both branch instances. Refer to the Table 4-48, "FBD Branch Reset Table" on page 248.
1	RWST	0	**SOFTCORERESET#: Soft Core Reset** See Timing diagram Figure 4-6. 0: Soft Core Reset Asserted 1: Soft Core Reset De-Asserted
0	RWST	0	**CORERESET#: Core Reset** See Timing diagram Figure 4-6. 0: Core Reset Asserted 1: Core Reset De-Asserted

Notes:
a. The nomenclature is Device 22 (branch 1), 21 (branch 0)

Table 4-48. FBD Branch Reset Table

FBDRST0.BRSELCMPRESET	FBDRST1.BRSELCMPRESET	Branch Selection for Reset
0	x	Branch 0 Reset
1	0	
1	1	Branch 1 Reset

Note: Signal soft_core_reset_b is controlled by the MCH hardware as well as FBDRST. The MCH will always assert soft_core_reset_b while the channel is in southbound electrical idle. FBDRST can force soft_core_reset_b assertion (0) at any time. It cannot cause soft_core_reset_b deassertion (1) when it is being asserted by the MCH hardware (SB electrical idle).

4.8.23.5 SPCPC[1:0]: Spare Copy Control

These controls set up sparing for each branch. Branch zero (device 21) takes precedence over branch one (device 22): if both spare-control-enabled branches' spare error thresholds trigger in the same cycle, sparing will only commence on branch zero. Sparing will not commence on a competing branch until it's in-progress competitor's spare control enable is cleared and it's CERRCNT criteria is still met.

Device: 22, 21
Function: 0
Offset: 40h

Bit	Attr	Default	Description
23:20	RV	0h	*Reserved2*
19:16	RW	0h	**FORCERANK: "From" Rank for Forced Spare Copy**
15:8	RW	0h	**SETH: Spare Error Threshold** A spare fail-over operation will commence when the SPAREN bit is set and a CERRCNT.RANK[i] count for one and only one rank hits this threshold.
7	RV	0h	*Reserved1*
6	RW	0	**FORCE: Initiate Spare Copy** '0'~>'1' transition while SPCPS.DSCIP = 0 initiates spare copy. SPAREN must be set to '1' for this bit to have effect.
5	RW	0	**SPAREN: Spare Control Enable** '1' enables sparing, '0' disables sparing. The SPRANK field defines other characteristics of the sparing operation. The MCH does not support sparing in mirrored mode: this bit should not be set if MC.MIRROR is set. If this bit is cleared before SPCPS.SFO is set, then if this bit is subsequently set while the spare trigger is still valid, then the spare copy operation will not resume from where it left off, but will instead restart from the beginning.
4	RV	0h	*Reserved*
3:0	RWL	0h	**SPRANK: Spare Rank** Target of the spare copy operation. This rank should not initially appear in a DMIR.RANK field. After the spare copy, The MCH will update the failed DMIR.RANK fields with this value. Enabled by SPAREN. Changes to this register will not be acknowledged by the hardware while SPCPS.DSCIP is set.

4.8.23.6 SPCPS[1:0]: Spare Copy Status

Device: 22, 21
Function: 0
Offset: 43h

Bit	Attr	Default	Description
7	RO	0	**LBTHR: Leaky Bucket Threshold Reached** '0' = Leaky-bucket threshold not reached '1' = Leaky-bucket count matches SPCPC.SETH. Cleared by reducing the offending count(s) in the CERRCNT registers.
6	RO	0	**DSCIP: DIMM Sparing Copy In Progress** '0' = DIMM sparing copy not in progress. '1' = DIMM sparing copy in progress. Set when SPCPC.SPAREN is set, and only one rank in CERRCNT is at threshold. This bit remains set until SFO is set. This bit is cleared when SFO is set. Error M27 is set when this bit transitions from '0' to '1'.
5	RO	0	**SFO: Spare Fail-Over** '0' = Spare has not been substituted for failing DIMM rank. '1' = Spare has been substituted for failing DIMM rank. Generates error M28. Cleared when SPCPC.SPAREN is cleared.
4	RV	0h	*Reserved*

Device: 22, 21 Function: 0 Offset: 43h			
Bit	**Attr**	**Default**	**Description**
3:0	RO	000	**FR: Failed Rank** Rank that was spared. Updated with the CERRCNT rank that has reached threshold when DSCIP is set.

4.8.23.7 MTR[1:0][7:0]: Memory Technology Registers

These registers define the organization of the DIMM's. There is one MTR for each pair of slots comprising either one, two, or four ranks. The parameters for these devices can be obtained by serial presence detect.

- MTR[0][7:0] defines slot-pairs [7:0] on branch[0]. MTR[1][7:0] defines slot-pairs [7:0] on branch[1].
- MTR[7:0] in Table 4-41 is MTR[7:0] for Device 21 which is MTR[0][7:0] for this Section 4.8.23.7.
- MTR[7:0] in Table 4-41 is MTR[7:0] for Device 22 which is MTR[1][7:0] for this Section 4.8.23.7.

This register must not be modified while servicing memory requests.The following three settings are mutually exclusive:

- 65,536 rows (NUMROW = "11")
- 4,096 columns (NUMCOL = "10")

Note the MTR offset arrangement in Table 4-27, "Device 21, 22, Function 0: FBD DIMM Map, Control, RAS". The offsets for MTR[1:0][0] to MTR[1:0][7] do not map linearly from 80h to 8Eh.

Device: 22, 21 Function: 0 Offset: 8Eh, 8Ch, 8Ah, 88h, 86h,84h, 82h, 80h			
Bit	**Attr**	**Default**	**Description**
15:9	RV	00h	*Reserved*
8	RW	0	**PRESENT: DIMMs are present** This bit is set if both DIMMs are present and their technologies are compatible.
7	RW	0	**ETHROTTLE: Technology - Electrical Throttle** Defines the electrical throttling level for these DIMMs: '0' = Electrical Throttling is disabled '1' = Electrical Throttling is enabled using the throttling level defined by the MC.ETHROT configuration field.
6	RW	0	**WIDTH: Technology - Width** Defines the data width of the SDRAMs used on these DIMMs '0' = x4 (4 bits wide) '1' = x8 (8 bits wide)
5	RW	0	**NUMBANK: Technology - Number of Banks** Defines the number of (real, not shadow) banks on these DIMMs '0' = four-banked '1' = eight-banked
4	RW	0	**NUMRANK: Technology - Number of Ranks** Defines the number of ranks on these DIMMs. '0' = single ranked '1' = double ranked

Device: 22, 21 Function: 0 Offset: 8Eh, 8Ch, 8Ah, 88h, 86h,84h, 82h, 80h			
Bit	**Attr**	**Default**	**Description**
3:2	RW	00	**NUMROW: Technology - Number of Rows** Defines the number of rows within these DIMMs. "00"= 8,192, 13 rows "01"= 16,384, 14 rows "10"= 32,768, 15 rows "11"= 65,536, 16 rows
1:0	RW	00	**NUMCOL: Technology - Number of Columns** Defines the number of columns within these DIMMs "00"= 1,024, 10 columns "01"= 2,048, 11 columns "10"= 4,096, 12 columns "11"= Reserved

4.8.23.8 DMIR[1:0][6:0]: DIMM Interleave Range

These registers define rank participation in various DIMM interleaves.

Each register defines a range. If the Memory (M) address falls in the range defined by an adjacent pair of DMIR.LIMIT's, the rank fields in the upper DMIR define the number and interleave position of ranks' way participation. Matching addresses participate in the corresponding ways. The combination of two equal ranks with three unequal ranks is illegal.

When a DMIR is programmed for a 2-way interleave, RANK0/RANK2 should be the same rank number and RANK1/RANK3 should be another rank number.

This register must not be modified while servicing memory requests.

Device: 22, 21 Function: 0 Offset: A8h, A4h, A0h, 9Ch, 98h, 94h, 90h			
Bit	**Attr**	**Default**	**Description**
31:27	RV	000h	*Reserved*
26:16	RW	00h	**LIMIT** This field defines the highest address in the range. Memory requests participate in this DMIR range if LIMIT[i] > M[37:28] >= LIMIT[i-1]. For i = 0, LIMIT[i-1]=0.
15:12	RW	0h	**RANK3** Defines which rank participates in WAY3.
11:8	RW	0h	**RANK2** Defines which rank participates in WAY2.
7:4	RW	0h	**RANK1** Defines which rank participates in WAY1.
3:0	RW	0h	**RANK0** Defines which rank participates in WAY0.

4.8.23.9 FBDICMD[1:0][1:0]: FBD Initialization Command

These registers define channel behavior during the "Init", "Reset" FBD initialization states. The "AMBID" field for the even-numbered channel also defines branch behavior during fast reset. This register cannot be manipulated until its corresponding FBDRST

register's reset sequence has been executed. For channels 0 and 1, the reset sequence on FBDRST[0] must be completed. For channels 2 and 3, the reset sequence on FBDRST[1] must be completed.

Device: 22, 21
Function: 0
Offset: 47h, 46h

Bit	Attr	Default	Description
7	RW	0	**EN: Enable** '0' = Drive electrical idle on the channel. '1' = Drive INITPAT on the channel. This field is not used during fast reset.
6:4	RW	00	**INITPAT: Initialization pattern** "000"=TS0: Training Sequence 0 to last AMB (not valid in "Reset") "001"=TS1: Training Sequence 1 to last AMB (not valid in "Reset") "010"=TS2: Training Sequence 2 to last AMB (not valid in "Reset") "011"=TS3: Training Sequence 3 to last AMB (not valid in "Reset") "100"=*reserved* "101"=TS2: Training Sequence 2 not to last AMB with NB Merge disabled (not valid in "Reset") "110"=TS2: Training Sequence 2 not to last AMB with NB Merge enabled (not valid in "Reset") "111"=All Ones (valid only in "Reset") This pattern is superseded by the "EN" bit. This field is not used during fast reset. The note '(not valid in "Reset")' indicates that is not valid when FBDST.STATE="Reset" or "Recovery Reset" and EN='1'. The note '(valid only in "Reset")' indicates that this is valid only when FBDST.STATE="Reset" or "Recovery Reset".
3:0	RW	0h	**AMBID: Advanced Memory Buffer IDentifier** Driven during the training sequences. This field is also used during fast reset to identify the last (southernmost) DIMM. ***Note:*** For HVM loopback to work the AMBID field should be set to 0xF. ***Note:*** For master-to-master IBIST, the only requirement is that the AMBID should be the same in both The MCH and in the AMB that is being setup as the master. A value of 0xF is used for both components (since AMB's do not merge AMBID in to TS1 header unless they are the last AMB).

4.8.23.10 FBDISTS[1:0][1:0]: FBD Initialization Status

The contents of this register are valid only during the "Init" and "Reset" FBD Initialization states. The fourteen bits [13:0] correspond to the northbound bitlanes.

Device: 22, 21
Function: 0
Offset: 5Ah, 58h

Bit	Attr	Default	Description
15:14	RV	0	*Reserved*
13:0	RO	0000h	**PATDET: Pattern Detection** "1" = Pattern recognized. "0" = Pattern not recognized. Bit-Lane Status is evaluated at the end of each instance of the pattern specified by the FBDICMD.EN and FBDICMD.INITPAT fields. Bit-Lane status is evaluated on each change to the FBDICMD.EN and FBDICMD.INITPAT.Only bits [2:0] are valid during electrical idle, and only after the FBDRST reset sequence has been executed. A recognizable training sequence must contain the FBDICMD.AMBID. TS1 detection is qualified by test patterns specified in section 4.3 of rev. 0.75 of FBD DFx specification [21], which defines the "SB/NB_Mapping" (1 bit), the "Test Parameters" (24 bits), and the "Electrical Stress Pattern".

4.8.23.11 AMBPRESENT[1:0][1:0]: FBD AMB Slot Present Register

This register controls the configuration transaction routing to the AMB slot per FBD channel. This includes accesses through both the memory mapped region (based on the AMBASE register) and the AMBSELECT register (for SMBus/JTAG access only, access via device 23, function 0.). Software needs to program this register after SPD discovery process. The MCH will check this register before it sends actual FBD AMB configuration transaction.

Device: 22, 21
Function: 0
Offset: 66h, 64h

Bit	Attr	Default	Description
15:0	RW	0000h	**AMBSP: Slot [bit_position] present in the FBD channel** 1: Indicates AMB slot addressed by DS[3:0] in decimal = [bit_position] is present; configuration transaction will be routed to FBD channel. Bit 15 controls DS[3:0] = 1111b, bit 14 controls DS[3:0] = 1110b,..., bit 0 controls DS[3:0] = 0000b. 0: AMB slot addressed by DS[3:0] in decimal = [bit_position] is not populated; no configuration transaction will be sent to FBD channel.

4.8.24 FBD RAS Registers

There are four sets of the following registers, one set for each FBD channel. They each appear in function 0 of different devices.

4.8.24.1 CERRCNTA[1:0][1:0]: Correctable Error Count A

These four registers (CERRCNTA through CERRCNTD) implement the "leaky-bucket" counters for correctable errors for each rank. Each field "limits" at a value of "255" ("1111 1111"). Non-zero counts are decremented when the ERRPER threshold is reached by the error period counter. Counts are frozen at the threshold defined by SPCPC.SETH and set the SPCPS.LBTHR bit. Writing a value of "1111 1111" clears and thaws the count. Changing SPCPC.SETH has no effect upon a frozen count.

For a CERRCNTx[b:a][d:c], the "b:a" range spans branches "1:0" for the two devices, and the "d:c" spans channels "upper:lower" for the two offsets.

Note: This register counts the number of correctable errors based on codeword granularity. Since each 32B of a cacheline plus 4B of ECC constitute a codeword, the correctable errors are incremented if the ECC check indicates an correctable error. Thus, there can be up to 2 correctable errors per cacheline if both portions were deemed correctable.

Note: Aliased uncorrectable errors are not counted as correctable errors.

This register "works" whether or not sparing is enabled. This register exercises no influence on mirrored branch operation.

Device: 22, 21
Function: 0
Offset: F0h, E0h

Bit	Attr	Default	Description
31:24	RWCST	0h	**RANK3: Error Count for Rank 3**
23:16	RWCST	0h	**RANK2: Error Count for Rank 2**
15:8	RWCST	0h	**RANK1: Error Count for Rank 1**
7:0	RWCST	0h	**RANK0: Error Count for Rank 0**

4.8.24.2 CERRCNTB[1:0][1:0]: Correctable Error Count B

Device: 22, 21
Function: 0
Offset: F4h, E4h

Bit	Attr	Default	Description
31:24	RWCST	0h	**RANK7: Error Count for Rank 7**
23:16	RWCST	0h	**RANK6: Error Count for Rank 6**
15:8	RWCST	0h	**RANK5: Error Count for Rank 5**
7:0	RWCST	0h	**RANK4: Error Count for Rank 4**

4.8.24.3 CERRCNTC[1:0][1:0]: Correctable Error Count C

Device: 22, 21
Function: 0
Offset: F8h, E8h

Bit	Attr	Default	Description
31:24	RWCST	0h	**RANK11: Error Count for Rank 11**
23:16	RWCST	0h	**RANK10: Error Count for Rank 10**
15:8	RWCST	0h	**RANK9: Error Count for Rank 9**
7:0	RWCST	0h	**RANK8: Error Count for Rank 8**

4.8.24.4 CERRCNTD[1:0][1:0]: Correctable Error Count D

Device: 22, 21
Function: 0
Offset: FCh, ECh

Bit	Attr	Default	Description
31:24	RWCST	0h	**RANK15: Error Count for Rank 15**
23:16	RWCST	0h	**RANK14: Error Count for Rank 14**
15:8	RWCST	0h	**RANK13: Error Count for Rank 13**
7:0	RWCST	0h	**RANK12: Error Count for Rank 12**

4.8.24.5 BADRAMA[1:0]: Bad DRAM Marker A

This register implements "failed-device" markers for the enhanced demand scrub algorithm. Hardware "marks" bad devices. The "mark" is a number between 1 and 18 inclusive. A value of "0_0000" indicates an "un-marked" rank: all RAM's are presumed "good". Only ranks containing x8 DRAM are "marked".

Device: 22, 21
Function: 0
Offset: C0h

Bit	Attr	Default	Description
31:30	RV	0h	Reserved
29:25	RWCST	00h	**RANK5: Bad device in Rank 5**
24:20	RWCST	00h	**RANK4: Bad device in Rank 4**
19:15	RWCST	00h	**RANK3: Bad device in Rank 3**
14:10	RWCST	00h	**RANK2: Bad device in Rank 2**
9:5	RWCST	00h	**RANK1: Bad device in Rank 1**
4:0	RWCST	00h	**RANK0: Bad device in Rank 0**

4.8.24.6 BADRAMB[1:0]: Bad DRAM Marker B

Device: 22, 21
Function: 0
Offset: C4h

Bit	Attr	Default	Description
31:30	RV	0h	Reserved
29:25	RWCST	00h	**RANK11: Bad device in Rank 11**
24:20	RWCST	00h	**RANK10: Bad device in Rank 10**
19:15	RWCST	00h	**RANK9: Bad device in Rank 9**
14:10	RWCST	00h	**RANK8: Bad device in Rank 8**
9:5	RWCST	00h	**RANK7: Bad device in Rank 7**
4:0	RWCST	00h	**RANK6: Bad device in Rank 6**

4.8.24.7 BADRAMC[1:0]: Bad DRAM Marker C

Device: 22, 21
Function: 0
Offset: C8h

Bit	Attr	Default	Description
31:20	RV	0h	Reserved
19:15	RWCST	00h	**RANK15: Bad device in Rank 15**
14:10	RWCST	00h	**RANK14: Bad device in Rank 14**
9:5	RWCST	00h	**RANK13: Bad device in Rank 13**
4:0	RWCST	00h	**RANK12: Bad device in Rank 12**

4.8.24.8 BADCNTA[1:0]: Bad DRAM Counter A

This register implements “failing-device” counters for the aliased uncorrectable error identification algorithm. “Count” double-adjacent symbol errors within x8 devices. “Drip” each counter after “MC.BADRAMTH” patrol scrub cycles through all of memory. Values of “MC.BADRAMTH” and “0” cannot be “dripped”. A value of “MC.BADRAMTH” cannot be incremented. “Mark” the BADRAM(A/B) register when a count reaches “MC.BADRAMTH”.

Device: 22, 21
Function: 0
Offset: CCh

Bit	Attr	Default	Description
31:28	RWCST	0000	**RANK7: Adjacent x8 symbol error count in Rank 7**
27:24	RWCST	0000	**RANK6: Adjacent x8 symbol error count in Rank 6**
23:20	RWCST	0000	**RANK5: Adjacent x8 symbol error count in Rank 5**
19:16	RWCST	0000	**RANK4: Adjacent x8 symbol error count in Rank 4**
15:12	RWCST	0000	**RANK3: Adjacent x8 symbol error count in Rank 3**
11:8	RWCST	0000	**RANK2: Adjacent x8 symbol error count in Rank 2**
7:4	RWCST	0000	**RANK1: Adjacent x8 symbol error count in Rank 1**
3:0	RWCST	0000	**RANK0: Adjacent x8 symbol error count in Rank 0**

4.8.24.9 BADCNTB[1:0]: Bad DRAM Counter B

Device: 22, 21
Function: 0
Offset: D0h

Bit	Attr	Default	Description
31:28	RWCST	0000	**RANK15: Adjacent x8 symbol error count in Rank 15**
27:24	RWCST	0000	**RANK14: Adjacent x8 symbol error count in Rank 14**
23:20	RWCST	0000	**RANK13: Adjacent x8 symbol error count in Rank 13**
19:16	RWCST	0000	**RANK12: Adjacent x8 symbol error count in Rank 12**
15:12	RWCST	0000	**RANK11: Adjacent x8 symbol error count in Rank 11**
11:8	RWCST	0000	**RANK10: Adjacent x8 symbol error count in Rank 10**
7:4	RWCST	0000	**RANK9: Adjacent x8 symbol error count in Rank 9**
3:0	RWCST	0000	**RANK8: Adjacent x8 symbol error count in Rank 8**

4.8.24.10 FBDSBTXCFG[1:0][1:0]: FBD Southbound Transmit Configuration Register

This register controls the FBD Southbound I/O Transmit configuration during normal operation. This value is programmed by BIOS on per channel basis.

Device: 22, 21
Function: 0
Offset: D5h, D4h

Bit	Attr	Default	Description
7:4	RV	0h	*Reserved.*
3:2	RWST	01	**SBTXDRVCUR: South Bound Tx drive Current** 00: 120% current 01: 100% current 10: 80% current 11: 60% current
1:0	RWST	10	**SBTXDEEMP: South Bound Tx De-emphasis** **With De-emphasis, the Tx differential p-p swing (eye height) is maintained at nominal during data transitions, but drops down to the de-emphasized value when there is no transition between the previous bit and current bit** 00: No De-emphasis 01: -3.5dB 10: -6dB 11: -9.5 dB

4.8.24.11 FBDPSCNTRL[1:0]: FBD Power Save Mode Register

This register controls inputs to the FBD IO cells related to power saving features.

Device: 22, 21
Function: 0
Offset: D6h

Bit	Attr	Default	Description
7:4	RV	0h	*Reserved.*
3	RWST	0	**DISDRCDET:** Setting this bit to '1' disabled the DRCLOCK detector circuit in the FBD IO.
2	RWST	0	**SAUEDGECLKDIS:** Setting this bit to '1' shuts off the edge logic in the FBD IO when it's not in use.
1	RWST	0	**TXSTRNCLKOFF:** Setting this bit to '1' shuts off the TX training logic in the FBD IO when it's not in use.
0	RWST	0	**EDGEIDACPSEN:** Setting this bit to '1' shuts off edge IDAC in the FBD IO when edge_off is asserted.

4.8.24.12 NBTRL[3:0]: Northbound Training Low Status

Result from TS1 with lower SB bit-lanes mapped to NB bit-lanes.

Device: 22, 21
Function: 0
Offset: 1D0, 2D0h

Bit	Attr	Default	Description
15:14	RV	0h	*Reserved.*
13:0	ROST	0	**NBTR: Northbound Training Results** For each bit-position, a: '0' = Bit-lane passed IBIST during TS1 '1' = Bit-lane failed IBIST during TS1.

4.8.24.13 NBTRH[3:0]: Northbound Training High Status

Result from TS1 with higher SB bit-lanes mapped to NB bit-lanes.

Device: 22, 21
Function: 0
Offset: 1D2h, 2D2h

Bit	Attr	Default	Description
15:14	RV	0h	*Reserved.*
13:0	ROST	0	**NBTR: Northbound Training Results** For each bit-position, a: '0' = Bit-lane passed IBIST during TS1 '1' = Bit-lane failed IBIST during TS1.

4.8.25 FB-DIMM IBIST Registers

4.8.25.1 FBD[3:0]IBPORTCTL: FBD IBIST Port Control Register

This register contains bits to control the operation of the IBIST DFT feature.

Device: 22,21
Function: 0
Offset: 280h, 180h

Bit	Attr	Default	Description
31:26	RV	0h	*Reserved*
25	RWST	0	**RXINVSWPMD: Rx Inversion Sweep Mode** 0: Match Sweep according to the SB-to-NB_Mapping field in the TS1 training sequence. The default setting forces the RX inversion pointers to follow the unique northbound inversion across the port width. It is based on a Modulo 5 of SBNBMAP bit setting. If e lanes Example; If SBNBMAP = 0 then Lanes [4:0] are used as the reference for checking Lanes[13:10], [9:5], and [4:0]. If SBNBMAP = 1 then Lanes [9:5] are used as the reference for checking Lanes[13:10], [9:5], and [4:0]. For the MCH lane [13] does not exist but it does participate in rotate-left-shift operations. 1: Enable full inversion sweep across the entire port. When enabled the RX inversion pointers become a single entity. Lanes [13:10] rotate left-shift completely across the width of the port. Even though Lane[13] is a DFT lane it will be "shifted through" to make the logic design easier. 0->1->2->3->4->5->6->7->8->9->10->11->12->13->0.
24	RW	1	**RXAUTOINVSWPEN: Auto-inversion sweep enable** This bit enable the inversion shift register to continuously rotate the pattern in the FIBRXSHFT register. This register enables the inversion pattern to the lane at the bit position indicated by a logic 1. 0: Disable Auto-inversion 1: Enable Auto-inversion
23	RW	0	**SBNBMAP: Southbound to northbound mapping for loopback testing** This bit indicates which set of lanes are replicated onto the northbound lanes. 0: Lower SB lanes 1: Upper SB lanes
22	RW	0	**CMMSTR: Compliance Measurement Mode** This bit forces the component into link reset then transmits the default IBIST pattern set. on all Tx lanes. The standard initialization sequence is followed with the training state machine progressing from EI to TS0 then TS1. 0: Disable CMM 1: Enable CMM. This feature requires the IBIST start bit to be set before the mode is enabled.
21:12	RWST	000h	**ERRCNT: Error Counter [9:0]** Total number of errors encountered in this port. Errors are accumulated per lane. If several errors occurred in one phit time then a binary encoded value of the number of errors is added to the error count.
11:8	ROST	00h	**ERRLNNUM: Error Lane Number [3:0]** This points to the first lane that encountered an error. If more than one lane reports an error in a cycle, the most significant lane number that reported the error will be logged.

Device: 22,21 Function: 0 Offset: 280h, 180h			
Bit	**Attr**	**Default**	**Description**
7:6	RWCST	0	**ERRSTAT: Port Error Status [1:0]** When IBIST is started, status goes to 01 until first start delimiter is received and then goes to 00 until the end or to10/11 as appropriate. 00: No error. 01: Did not receive first start delimiter. 10: Transmission error (first error). 11: Reserved.
5	RW	1	**AUTOINVSWPEN: Auto-inversion sweep enable** This bit enable the inversion shift register to continuously rotate the pattern in the FIBTXSHFT register. 0: Disable Auto-inversion 1: Enable Auto-inversion
4	RW	0	**STOPONERR: Stop IBIST on Error** 0: Do not stop on error, only update error counter 1: Stop on error
3	RW	0	**LOOPCON: Loop forever** 0: No looping 1: Loop forever
2	RWCST	0	**IBDONE: IBIST done flag** 0: Not done 1: Done
1	RWST	1	**MSTRMD: Master Mode Enable** When this bit is set the next TS1 training set that has the loopback bit set will cause the transmitter to operate as a master. Even though the IBIST is in the loopback state it is not in loopback. 0: Disable Master mode. This component will not enter into master when a TS1 training set with loopback bit set. 1: Enable Master Mode on the next TS1 training with loopback bit set
0	RWST	0	**IBSTR: IBIST Start** When set, it enables receiver logic to look for start delimiters during TS1 training set. If the MSTRMD bit is set, the start bit enables the transmit state machine to start transmitting patterns during the TS1 training set. The receiver is enable in both cases. For master-slave mode, the pattern will be looped back as defined in the FBD spec. In master-master mode, the IBIST controller will originate patterns and also check the incoming pattern for errors. 0: Stop IBIST transmitter 1: Start IBIST transmitter

4.8.25.2 FBD[3:0]IBTXPGCTL: FBD IBIST Pattern Generator Control Register

This register contains bits to control the operation of the IBIST pattern generator.

Device: 22,21 Function: 0 Offset: 284h, 184h			
Bit	**Attr**	**Default**	**Description**
31:26	RWST	04h	**OVRLOPCNT: Overall Loop Count[5:0]** 0h: Send no IBIST data in payload 1h-3Fh: The number of times to loop through all the patterns

Device: 22,21 Function: 0 Offset: 284h, 184h			
Bit	**Attr**	**Default**	**Description**
25:21	RWST	0h	**CNSTGENCNT: Constant Generator Loop Counter[4:0]** 00h: Disable constant generator output 01h: 1Fh The number of times the Modulo-N counter should be repeated before going to the next pattern type. Each buffer transfer is composed of two frames (loop counts of 24-bits each).
20	RWST	0	**CNSTGENSET: Constant Generator Setting** 0: Generate 0 1: Generate 1
19:13	RWST	19h	**MODLOPCNT: Modulo-N Loop Counter [7:0]** Each count represents 24-bits of the pattern specified by the MODPERIOD bit field. 00h: Disable Pattern Output 01h: 7Fh The number of times the Pattern Buffer should loop before going to the next
12:10	RWST	1h	**MODPERIOD: Period of the Modulo-N counter** 000: Reserved 001: L/2 0101_0101_0101_0101_0101_0101 010: L/4 0011_0011_0011_0011_0011_0011 011: L/6 0001_1100_0111_0001_1100_0111 100: L/8 0000_1111_0000_1111_0000_1111 101: Reserved 110: L/12 0000_0000_0000_1111_1111_1111 111: Reserved
9:3	RWST	19h	**PATTLOPCNT: Pattern Buffer Loop Counter[6:0]** 00h: Disable Pattern Output 01h-7Fh: The number of times the Pattern Buffer should be repeated before going to the next pattern type. Each buffer transfer is composed of two frames (loop counts of 24-bits each).
2:0	RWST	000	**PTGENORD: Pattern Generation Order** 000: Pattern Store + Modulo N Cntr + Constant Generator 001: Pattern Store + Constant Generator + Modulo N Cntr 010: Modulo N Cntr + Pattern Store + Constant Generator 011: Modulo N Cntr + Constant Generator + Pattern Store 100: Constant Generator + Pattern Store + Modulo N Cntr 101: Constant Generator + Modulo N Cntr + Pattern Store 110: Reserved 111: Reserved

4.8.25.3 FBD[3:0]IBPATBUF: FBD IBIST Pattern Buffer Register

This register contains the pattern bits used in IBIST operations.

Device: 22,21 Function: 0 Offset: 288h, 188h			
Bit	**Attr**	**Default**	**Description**
31:24	RV	0	**Reserved**
23:0	RWST	02CCFDh	**IBPATBUF: IBIST Pattern Buffer** Pattern buffer storing the default and the user programmable pattern. Default: 0000_0010_1100_1100_1111_1101

4.8.25.4 FBD[3:0]IBTXMSK: IBIST Transmitter Mask

This register determines which lanes are enabled for IBIST operations. These bits also control the power saving features of each lane. If a particular lane is masked off, the power to that lane is reduced as much as possible.

Device: 22,21
Function: 0
Offset: 28Ch, 18Ch

Bit	Attr	Default	Description
31:14	RV	0h	**Reserved**
13:10	RWST	0h	**txmaskhvm: Transmit Mask extra DFT pins for HVM symmetry** Selects which lanes to enable for testing. A lane that is *not* selected remains in electrical idle.
9:0	RWST	3FFh	**txmask: Transmit Mask** Selects which lanes to enable for testing. A lane that is *not* selected remains in electrical idle.

4.8.25.5 FBD[3:0]IBRXMSK: IBIST Receiver Mask

This register determines which lanes are enabled for IBIST operations. These bits also control the power saving features of each lane. If a particular lane is masked off, the power to that lane is reduced as much as possible.

Device: 22,21
Function: 0
Offset: 290h, 190h

Bit	Attr	Default	Description
31:14	RV	0h	**Reserved**
13:0	RWST	3FFFh	**rxmask: Receive Mask** Selects which lanes to enable for testing. An Rx lane that is not selected is not included in Rx channel training does not contribute to the accumulation of error counts.

4.8.25.6 FBD[3:0]IBTXSHFT: IBIST Transmit Shift Inversion Register

This register indicates which channel is currently inverting the pattern to create cross talk conditions on the port.

Device: 22,21
Function: 0
Offset: 294h, 194h

Bit	Attr	Default	Description
31:14	RV	0h	**Reserved**
13:10	RWST	0h	**txinvshfthvm: Transmit Inversion shift register extra DFT pins for HVM symmetry** The pattern loaded in this register indicates which lanes are used for inversion. A logic 1 enables the lane connected to a particular bit position to invert the pattern that is being transmitted. Because this is a shift register the initial value will be left-shifted at the end of the loop count during IBIST operations.
9:0	RWST	001h	**txinvshft: Transmitter Inversion Shift Register** The pattern loaded in this register indicates which lanes are used for inversion. A logic 1 enables the lane connected to a particular bit position to invert the pattern that is being transmitted. Because this is a shift register the initial value will be left-shifted at the end of the loop count during IBIST operations.

4.8.25.7 FBD[3:0]IBRXSHFT: IBIST Receive Shift Inversion Register

This register indicates which channel is currently inverting the pattern to create cross talk conditions on the port.

Device: 22,21
Function: 0
Offset: 298h, 198h

Bit	Attr	Default	Description
31:14	RV	0h	**Reserved**
13	RWST	0	**rxinvshfthi: Receiver Inversion Shift Register for DFT** The pattern loaded in this bit field indicates which lanes are used for inversion. A logic 1 enables the lane connected to a particular bit position to invert the pattern that is being transmitted. This bit location will experience rotate-left-shift operation with bits[12:0].
12:0	RWST	0001h	**rxinvshft: Receiver Inversion Shift Register** The pattern loaded in this register indicates which lanes are used for inversion. A logic 1 enables the lane connected to a particular bit position to invert the pattern that is being transmitted. This register acts as a rotate-left shift register regardless of the setting of **RXINVSWPMD** bit. The Modulo-5 value is used to compare each sub-section of the northbound lanes for error checking.

4.8.25.8 FBD[3:0]RXLNERR: IBIST Receive Lane Error Register

This register enables IBIST operations for individual lanes.

Device: 22,21
Function: 0
Offset: 29Ch, 19Ch

Bit	Attr	Default	Description
31:14	RV	0h	**Reserved**
13	ROST	0	**rxerrstat: Receive error lane status for DFT.** This register records the error from lane 13 of this port.
12:0	ROST	0	**rxerrstat: Receive error lane status.** This register records the errors from all lanes of this port.

4.8.25.9 FBD[3:0]IBRXPGCTL: FBD IBIST Rx Pattern Generator Control Register

This register contains bits to control the operation of the Rx pattern generator.

Device: 22,21
Function: 0
Offset: 2A0h, 1A0h

Bit	Attr	Default	Description
31:26	RWST	04h	**OVRLOPCNT: Overall Loop Count[5:0]** 0h: Send no IBIST data in payload 1h-3Fh: The number of times to loop through all the patterns
25:21	RWST	0h	**CNSTGENCNT: Constant Generator Loop Counter[4:0]** 00h: Disable constant generator output 01h: 1Fh The number of times the Modulo-N counter should be repeated before going to the next pattern type. Each buffer transfer is composed of two frames (loop counts of 24-bits each).
20	RWST	0	**CNSTGENSET: Constant Generator Setting** 0: Generate 0 1: Generate 1
19:13	RWST	19h	**MODLOPCNT: Modulo-N Loop Counter [7:0]** Each count represents 24-bits of the pattern specified by the MODPERIOD bit field. 00h: Disable Pattern Output 01h: 7Fh The number of times the Pattern Buffer should loop before going to the next
12:10	RWST	1h	**MODPERIOD: Period of the Modulo-N counter** 001: L/2 0101_0101_0101_0101_0101_0101 010: L/4 0011_0011_0011_0011_0011_0011 011: L/6 0001_1100_0111_0001_1100_0111 100: L/8 0000_1111_0000_1111_0000_1111 110: L/12 0000_0000_0000_1111_1111_1111
9:3	RWST	19h	**PATTLOPCNT: Pattern Buffer Loop Counter[6:0]** 00h: Disable Pattern Output 01h-7Fh: The number of times the Pattern Buffer should be repeated before going to the next pattern type. Each buffer transfer is composed of two frames (loop counts of 24-bits each).
2:0	RWST	000	**PTGENORD: Pattern Generation Order** 000: Pattern Store + Modulo N Cntr + Constant Generator 001: Pattern Store + Constant Generator + Modulo N Cntr 010: Modulo N Cntr + Pattern Store + Constant Generator 011: Modulo N Cntr + Constant Generator + Pattern Store 100: Constant Generator + Pattern Store + Modulo N Cntr 101: Constant Generator + Modulo N Cntr + Pattern Store 110: Reserved 111: Reserved

4.8.25.10 FBD[3:0]IBPATBUF2: FBD IBIST Pattern Buffer 2 Register

This register contains the pattern bits used in IBIST operations.

Device: 22,21
Function: 0
Offset: 2A4h, 1A4h

Bit	Attr	Default	Description
31:24	RV	0	**Reserved**
23:0	RWST	02CCFDh	**IBPATBUF: IBIST Pattern Buffer** Pattern buffer storing the default and the user programmable pattern. Default: 0000_0010_1100_1100_1111_1101

4.8.25.11 FBD[3:0]IBTXPAT2EN: IBIST TX Pattern Buffer 2 Enable

This register enables which channels are inverted when IBIST operations are activated.

Device: 22,21
Function: 0
Offset: 2A8h, 1A8h

Bit	Attr	Default	Description
31:15	RV	0h	**Reserved**
14	RWST	0h	**lfsrmode:** If this bit is zero, pattern buffer 2 will be used as a static value. If this bit is one, pattern buffer 2 will be lsfr.
13:10	RWST	0h	**txpatt2hvmen: receiver Pattern Buffer 2 Enable for the HVM lanes** Selects which channels to enable the second pattern buffer.
9:0	RWST	000h	**txpatt2en: receiver Pattern Buffer 2 Enable** Selects which channels to enable the second pattern buffer.

4.8.25.12 FBD[3:0]IBRXPAT2EN: IBIST RX Pattern Buffer 2 Enable

This register enables inversion pattern testing on individual lanes.

Device: 22,21
Function: 0
Offset: 2ACh, 1ACh

Bit	Attr	Default	Description
31:15	RV	0h	**Reserved**
14	RWST	0h	**lfsrmode:** If this bit is zero, pattern buffer 2 will be used as a static value. If this bit is one, pattern buffer 2 will be lsfr.
13:0	RWST	0000h	**rxpatt2en: Receiver Pattern Buffer 2 Enable** Selects which channels to enable the second pattern buffer.

4.8.26 Serial Presence Detect Registers

There are two sets of the following registers, one set for each FBD branch. They each appear in function 0 of different devices.

4.8.26.1 SPD[1:0][1:0]: Serial Presence Detect Status Register

This register provides the interface to the SPD bus (SCL and SDA signals) that is used to access the Serial Presence Detect EEPROM that defines the technology, configuration, and speed of the DIMMs controlled by the MCH.

Device: 22, 21
Function: 0
Offset: 76h, 74h

Bit	Attr	Default	Description
15	RO	0	**RDO: Read Data Valid.** This bit is set by the MCH when the Data field of this register receives read data from the SPD EEPROM after completion of an SPDR command. It is cleared by the MCH when a subsequent SPDR command is issued.
14	RO	0	**WOD: Write Operation Done.** This bit is set by the MCH when a SPDW command has been completed on the SPD bus. It is cleared by the MCH when a subsequent SPDW command is issued.
13	RO	0	**SBE: SPD Bus Error.** This bit is set by the MCH if it initiates an SPD bus transaction that does not complete successfully. It is cleared by the MCH when an SPDR or SPDW command is issued.
12	RO	0	**BUSY: Busy state.** This bit is set by the MCH while an SPD command is executing.
11:8	RV	0h	*Reserved.*
7:0	RO	00h	**DATA: Data.** Holds data read from SPDR commands.

4.8.26.2 SPDCMD[1:0][1:0]: Serial Presence Detect Command Register

A write to this register initiates a DIMM EEPROM access through the SPD bus.

Device: 22, 21
Function: 0
Offset: 7Ch, 78h

Bit	Attr	Default	Description
31:28	RWST	1010	**DTI: Device Type Identifier.** This field specifies the device type identifier. Only devices with this device-type will respond to commands. "1010" specifies EEPROM's. "0110" specifies a write-protect operation for an EEPROM. Other identifiers can be specified to target non-EEPROM devices on the SPD bus.
27	RWST	1	**CKOVRD: Clock Override.** '0' = Clock signal is driven low, overriding writing a '1' to CMD. '1' = Clock signal is released high, allowing normal operation of CMD. Toggling this bit can be used to "budge" the port out of a "stuck" state.
26:24	RWST	000	**SA: Slave Address.** This field identifies the DIMM EEPROM to be accessed through the SPD register.
23:16	RWST	00h	**BA: Byte Address.** This field identifies the byte address to be accessed through the SPD register.
15:8	RWST	00h	**DATA: Data.** Holds data to be written by SPDW commands.
7:1	RV	0h	*Reserved*
0	RWST	0	**CMD: Command.** Writing a '0' to this bit initiates an SPDR command. Writing a '1' to this bit initiates an SPDW command.

4.8.27 DMA Engine Configuration Registers

4.8.27.1 PCICMD: PCI Command Register

Device: 8
Function: 0
Offset: 04h

Bit	Attr	Default	Description
15:11	RV	0h	*Reserved*
10	RW	0	**INTxDisable: Interrupt Disable** This bit controls the ability of the Intel® QuickData Technology device to assert a legacy PCI interrupt during DMA completions or DMA errors. 1: Legacy Interrupt mode is disabled 0: Legacy Interrupt mode is enabled
9	RO	0	**FB2B: Fast Back-to-Back Enable** This bit does not apply to the Intel® QuickData Technology Device and hardwired to 0.
8	RW	0	**SERRE: SERR Message Enable** This bit indicates whether the Intel® QuickData Technology device is allowed to signal a SERR condition. This field handles the reporting of fatal and non-fatal errors by enabling the error pins ERR[2:0]. 1: The Intel® QuickData Technology device is enabled to send fatal/non-fatal errors. 0: The Intel® QuickData Technology device is disabled from generating fatal/non-fatal errors.
7	RV	0	*Reserved*
6	RW	0	**PERRRSP: Parity Error Response** Controls the response when a parity error is detected in the DMA engine 1: The device can report Parity errors 0: Parity errors can be ignored by the device.
5:4	RV	00	*Reserved*
3	RO	0	**SPCEN: Special Cycle Enable** This bit does not apply to the Intel® QuickData Technology Device.
2	RW	0	**BME: Bus Master Enable** Controls the ability for the Intel® QuickData Technology device to initiate transactions to memory including MMIO 1: Enables the Intel® QuickData Technology device to successfully complete memory read/write requests. 0: Disables upstream memory writes/reads If this bit is not set and the DMA engine is programmed by software to process descriptors, the Chipset will flag *DMA3 error in the FERR/NERR and CHANERR registers when it attempts to issue descriptor access to memory. It is expected that software should not change this BME bit dynamically until the DMA engine is quiesced (reset/hotplug/Power Management etc.)
1	RW	0	**MAEN: Memory Access Enable** Controls the ability for the Intel® QuickData Technology Device to respond to memory mapped I/O transactions initiated in the MCH in its range. 1: Allow MMIO accesses in the Intel® QuickData Technology 0: Disable MMIO accesses in Intel® QuickData Technology This only applies to access IQD_BAR space in Device 8, fn 1 where the MMIO space resides (Requests from both fast/slow paths will be master-aborted)
0	RO	0	**IOAEN: I/O Access Enable** Controls the ability for the Intel® QuickData Technology Device to respond to legacy I/O transactions. The Intel® QuickData TechnologyDevice does not support/allow legacy I/O cycles.

The PCI Command register follows a subset of the *PCI Local Bus Specification*, Revision 2.3 specification. This register provides the basic control of the ability of the Intel® QuickData Technology device to initiate and respond to transactions sent to it and maintains compatibility with PCI configuration space.

4.8.27.2 PCISTS: PCI Status Register

Device: 8
Function: 0
Offset: 06h

Bit	Attr	Default	Description
15	RWC	0	**DPE: Detected Parity Error** This bit is set when the Intel® QuickData Technology device receives an uncorrectable data error or Address/Control parity errors regardless of the Parity Error Response bit (PCICMD.PERRRSP). This applies only to parity errors that target the Intel® QuickData Technology device (inbound/outbound direction). The detected parity error maps to B1, F6, M2 and M4 (uncorrectable data error from FSB, Memory or internal sources). The DMA engine also records the data parity error in bit[6] (**Cdata_par_err) of** the CHANERR register.
14	RWC	0	**SSE: Signalled System Error** 1: The Intel® QuickData Technology device reported internal FATAL/NON FATAL errors (DMA0-15) through the ERR[2:0] pins with SERRE bit enabled. Software clears this bit by writing a '1' to it. 0: No internal Intel® QuickData Technology device port errors are signaled.
13	RO	0	**RMA: Received Master Abort Status** This field is hardwired to 0 as there is no Master Abort for the DMA operations
12	RWC	0	**RTA: Received Target Abort Status** This field is hardwired to 0 as there is no Target Abort for the DMA operations
11	RWC	0	**STA: Signalled Target Abort Status**: This field is hardwired to 0
10:9	RO	00	**DEVSELT: DEVSEL# Timing**: This bit does not apply to the Intel® QuickData Technology Device.
8	RWC	0	**MDIERR: Master Data Integrity Error** This bit is set by the Intel® QuickData Technology device if the Parity Error Response bit (PCICMD.PERRRSP) is set and it receives error B1, F2, F6, M2 and M4 (uncorrectable data error or Address/Control parity errors or an internal failure). If the PERRRSP bit in the Section 4.8.27.1, "PCICMD: PCI Command Register" on page 267 is cleared, this bit is never set.
7	RO	0	**FB2B: Fast Back-to-Back Capable** Not applicable to Intel® QuickData Technology. Hardwired to 0.
6	RV	0	*Reserved*
5	RO	0	**66MHZCAP: 66MHz capable.** Not applicable to Intel® QuickData Technology. Hardwired to 0.
4	RO	1	**CAPL: Capability List Implemented**: This bit indicated that the Intel® QuickData Technology device implements a PCI Capability list. See CAPPTR at offset 34h
3	RO	0	**INTxST: INTx State** This bit is set by the hardware when the Intel® QuickData Technology device issues a legacy INTx (pending) and is reset when the Intx is deasserted. The intx status bit should be deasserted when all the relevant status bits/events viz DMA errors/completions that require legacy interrupts are cleared by software.
2:0	RV	000	*Reserved*

The PCI Status register follows a subset of the *PCI Local Bus Specification*, Revision 2.3 specification. This register maintains compatibility with PCI configuration space. Since this register is part of the standard PCI header, there is a PCISTS register per PCI function.

4.8.27.3 CCR: Class Code Register

Device: 8 Function: 0 Offset: 09h			
Bit	**Attr**	**Default**	**Description**
23:16	RWO	08h	**Base Class Code**: A 08H code indicates that the Intel® QuickData Technology device is a peripheral device[a]. A 06H code is used to indicate a Host bridge device. Default: 08h
15:8	RWO	80h	**Sub-Class Code**: An 80H code indicates that the Intel® QuickData Technology device is a non-specific peripheral device. A 00H code is used to indicate a Host bridge device. Default: 80h
7:0	RWO	0h	**Register-Level Programming Interface**: This field identifies a default value for non-specific programming requirements.

Notes:

a. A peripheral device in this case denotes an integrated device in the root complex.

The bits in this register are writable once by BIOS in order to allow the device to be programmable either as an OS-visible device [088000h](implementing a driver) or a chipset host bridge device [060000h] (relying on BIOS code and/or pure hardware control for programming the Intel® QuickData Technology registers). The default value of the CCR is set to 088000h (corresponding to an integrated device in the root port).

4.8.27.4 IQD_BAR: Intel® QuickData Technology Base Address Register

Device: 8 Function: 0 Offset: 10h			
Bit	**Attr**	**Default**	**Description**
63:40	RW	0h	**IQD_BASE_Win_Upper: Upper DMABase Window**: The upper bits of the 64-bit addressable space are left as RW only for compliance reasons. These are not expected to be programmed to be anything other than all 0s.
39:10	RW	003F9C00h	**IQD_BASE_WIN: DMABase Window** This marks the 1KB memory-mapped registers used for the chipset DMA and can be placed in any MMIO region (low/high) within the physical limits of the system. For instance the MCH uses only 40-bit addressable space. Hence bits 39:10 are assumed to be valid and also contains the default value of the IQD_BAR in the FE70_0000h to FE70_03FFh range.
9:4	RV	0h	*Reserved*
3	RO	0	**Pref: Prefetchable** The DMA registers are not prefetchable.
2:1	RO	10	**Type: Type** The DMA registers is 64-bit address space and can be placed anywhere within the addressable region of the BF chipset (up to 40-bits).
0	RO	0	**Mem_space: Memory Space**: This Base Address Register indicates memory space.

This Intel® QuickData Technology base address register marks the memory-mapped registers used for the DMA functionality.

4.8.27.5 CAPPTR: Capability Pointer Register

Device: 8 Function: 0 Offset: 34h			
Bit	**Attr**	**Default**	**Description**
7:0	RO	50h	**CAPPTR: Capability Pointer** This register field points to the first capability viz. PM structure in the Intel® QuickData Technology device.

4.8.27.6 INTL: Interrupt Line Register

The Interrupt Line register is used to communicate interrupt line routing information between initialization code and the device driver.

Device: 8 Function: 0 Offset: 3Ch			
Bit	**Attr**	**Default**	**Description**
7:0	RW	00h	**INTL: Interrupt Line** BIOS writes the interrupt routing information to this register to indicate which input of the interrupt controller this PCI Express Port is connected to. Not used in the MCH since the PCI Express port does not have interrupt lines.

4.8.27.7 INTP: 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 ESI port using the Assert_Intx commands as appropriate.

Device: 8 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 ***Note:*** Since the Intel® QuickData Technology is a single function device, only INTA will be generated and BIOS should lock this "001" encoding.

4.8.27.8 Power Management Capability Structure

The DMA engine integrated device within the MCH incorporates power management capability with D0 (working) and a pseudo D3hot/cold states (sleep) that can be controlled independently through software. From a software perspective, the D3 states convey information to the power controller that the device is in the sleep mode though the physical entity inside the chipset may be fully powered. During transition[1] from D0 to D3, it will ensure that all pending DMA Channels are completed in full.

1. The Intel® QuickData Technology must enter D3 before completing the power management handshake with the MCH.

4.8.27.8.1 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.

Device: 8
Function: 0
Offset: 50h

Bit	Attr	Default	Description
31:27	RO	11001	**PMES: PME Support** Identifies power states which assert PMEOUT. Bits 31, 30 and 27 must be set to '1' for PCI-PCI bridge structures representing ports on root complexes. The definition of these bits is taken from the PCI Bus Power Management Interface Specification Revision 1.1. XXXX1b - PMEOUT can be asserted from D0 XXX1Xb - PMEOUT can be asserted from D1 (Not supported by the MCH) XX1XXb - PMEOUT can be asserted from D2 (Not supported by the MCH) X1XXXb - PMEOUT can be asserted from D3 hot 1XXXXb - PMEOUT can be asserted from D3 cold
26	RO	0	**D2S: D2 Support** The MCH does not support power management state D2.
25	RO	0	**D1S: D1 Support** The MCH does not support power management state D1**.**
24:22	RO	0h	**AUXCUR: AUX Current**
21	RO	0	**DSI: Device Specific Initialization**
20	RV	0	*Reserved.*
19	RO	0	**PMECLK: PME Clock** This field is hardwired to 0h.
18:16	RO	010	**VER: Version** This field is set to 2h as version number from the PCI Express 1.0 specification.
15:8	RO	58h	**NXTCAPPTR: Next Capability Pointer** This field is set to offset 58h for the next capability structure (MSI) in the PCI 2.3 compatible space.
7:0	RO	01h	**CAPID: Capability ID** Provides the PM capability ID assigned by PCI-SIG.

4.8.27.8.2 PMCSR: Power Management Control and Status Register

This register provides status and control information for PM events in the PCI Express port of the Intel® QuickData Technology Device.

Device: 8
Function: 0
Offset: 54h

Bit	Attr	Default	Description
31:24	RO	0h	**Data: Data** Data read out based on data select (DSEL). Refer to section 3.2.6 of PCI PM specification for details. This is not implemented in the Intel® QuickData Technology Power Management capability for the MCH and is hardwired to 0h.
23	RO	0h	**BPCCEN: Bus Power/Clock Control Enable** This field is hardwired to 0h.
22	RO	0h	**B2B3S: B2/B3 Support** This field is hardwired to 0h.
21:16	RV	0h	*Reserved.*

Device: 8 Function: 0 Offset: 54h			
Bit	**Attr**	**Default**	**Description**
15	RWCST	0h	**PMESTS: PME Status** This PME Status is a sticky bit. When set, the device generates a PME internally independent of the PMEEN bit defined below. Software clears this bit by writing a '1'. As an integrated device within the root complex, the MCH will never set this bit, because it never generates a PME internally independent of the PMEEN bit.
14:13	RO	0h	**DSCL: Data Scale** This 2-bit field indicates the scaling factor to be used while interpreting the "data_scale" field.
12:9	RO	0h	**DSEL: Data Select** This 4-bit field is used to select which data is to reported through the "data" and the "Data Scale" fields.
8	RWST	0h	**PMEEN: PME Enable** This field is a sticky bit and when set enables PMEs generated internally to appear at the ICH6 through the "Assert(Deassert)_PMEGPE"message. This has no effect on the MCH since it does not generate PME events internally.
7:2	RV	0h	*Reserved.*
1:0	RW	0h	**PS: 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 (reserved) 10: D2 (reserved) 11: D3_hot If Software sets this to D1 or D2, then the power state will default to D0. ***Note:*** If Software sets the PS field to D3hot when the DMA is underway, it will complete all the pending transfers/commands. i.e. walk through the entire descriptor chain, status writes/interrupts and then reach the halted state (quiesced). The MCH will drop any start, abort, suspend, resume and append commands when in D3_hot and the channel is halted or completed.

4.8.27.9 MSICAPID: Message Signalled Interrupt Capability ID Register

Device: 8 Function: 0 Offset: 58h			
Bit	**Attr**	**Default**	**Description**
7:0	RO	05h	**CAPID: MSI Capability ID** This code denotes the standard MSI capability assigned by PCI-SIG

4.8.27.10 MSINXPTR: Message Signalled Interrupt Next Pointer Register

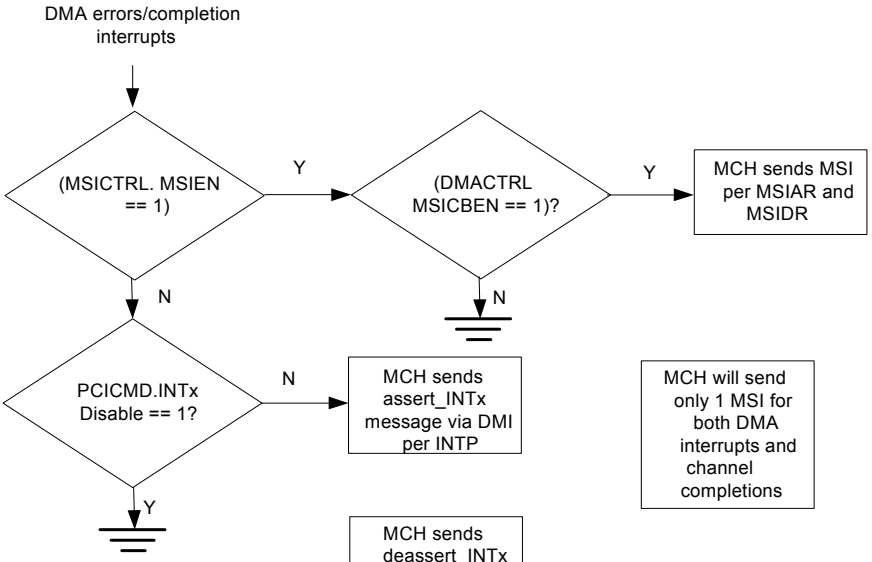
Device: 8 Function: 0 Offset: 59h			
Bit	**Attr**	**Default**	**Description**
7:0	RO	6Ch	**NXTPRT: MSI Next Pointer**: The Intel® QuickData Technology device is implemented as a PCI Express device and this points to the PCI Express capability structure.

4.8.27.11 MSICTRL: Message Signalled Interrupt Control Register

Device: 8
Function: 0
Offset: 5Ah

Bit	Attr	Default	Description
15:8	RO	0h	*Reserved*
7	RO	0	**AD64CAP: 64-bit Address Capable** All current Intel® architecture requires only 32b addresses and hence is hardwired to 0
6:4	RW	000	**MMEN: Multiple Message Enable** Software initializes this to indicate the number of allocate messages which is aligned to a power of two. When MSI is enabled, the software will allocate at least one message to the device. See Section 4.8.27.13, "MSIDR: Message Signalled Interrupt Data Register" on page 275 below for discussion on how the interrupts are handled. ***Note:*** If software writes a value greater than the limit specified by the MMCAP field in the MMEN field, it is considered as a programming error. The MCH will only use the LSB of the MMEN (as a power of 2) to decode the Table 4-49, "IV Vector Table for DMA Errors and Interrupts" on page 275 and signal one message for DMA errors/completions.
3:1	RO	0h	**MMCAP: Multiple Message Capable** The MCH Intel® QuickData Technology supports only one interrupt message (power of two) for handling • DMA errors • DMA completions
0	RW	0	**MSIEN: MSI Enable** This bit enables MSI as the interrupt mode of operation instead of the legacy interrupt mechanism. 0: Disables MSI from being generated. 1: Enables MSI messages to be generated for DMA related interrupts. An extract of the flowchart of the Intel® QuickData Technology error handling is given in Figure 4-7, "Intel® 7300 Chipset DMA Error/Channel Completion Interrupt Handling Flow" on page 274

Figure 4-7. Intel® 7300 Chipset DMA Error/Channel Completion Interrupt Handling Flow



MSIEN	MSICBEN	INTx Disable	Output
1	1	x	MSI
1	0	x	--
0	x	0	assert_intx
0	x	1	--

4.8.27.12 MSIAR: Message Signalled Interrupt Address Register

Device: 8
Function: 0
Offset: 5Ch

Bit	Attr	Default	Description
31:20	RO	FEEh	**AMSB: Address MSB** This field specifies the 12 most significant bits of the 32-bit MSI address.
19:12	RW	0h	**ADSTID: Address Destination ID** This field is initialized by software for routing the interrupts to the appropriate destination.
11:4	RW	0h	**AEXDSTID: Address Extended Destination ID** This field is not used by IA-32 processor.
3	RW	0	**ARDHINT: Address Redirection Hint** 0: directed 1: redirectable
2	RW	0	**ADM: Address Destination Mode** 0: physical 1: logical

Device: 8 Function: 0 Offset: 5Ch			
Bit	**Attr**	**Default**	**Description**
1:0	RV	00	*Reserved.* Not used since the memory write is D-word aligned

4.8.27.13 MSIDR: Message Signalled Interrupt Data Register

Device: 8 Function: 0 Offset: 60h			
Bit	**Attr**	**Default**	**Description**
31:16	RV	0000h	*Reserved.*
15	RW	0h	**TM: Trigger Mode** This field Specifies the type of trigger operation 0: Edge 1: level
14	RW	0h	**LVL: Level** if TM is 0h, then this field is a don't care. Edge triggered messages are always treated as assert messages. For level triggered interrupts, this bit reflects the state of the interrupt input if TM is 1h, then 0: Deassert Messages 1: Assert Messages
13:11	RW	0h	These bits are don't care in IOxAPIC interrupt message data field specification.
10:8	RW	0h	**DM: Delivery Mode** 000: Fixed 001: Lowest Priority 010: SMI/HMI 011: *Reserved* 100: NMI 101: INIT 110: *Reserved* 111: ExtINT
7:0	RW	0h	**IV: Interrupt Vector** The interrupt vector as programmed by BIOS/Software will be used by the MCH to provide context sensitive interrupt information for different events such as DMA Errors, DMA completions that require attention from the processor. See Table 4-49 for IV handling for DMA.

Table 4-49. IV Vector Table for DMA Errors and Interrupts

Number of Messages enabled by Software[a]	Events	IV[7:0]
1+	All (DMA completions/errors)	xxxxxxxx[b]

Notes:

a. If Software sets MSICTRL.MMEN such that the number of messages is greater than 1, the MCH Intel® QuickData Technology device will still generate only 1 message.

b. The term "xxxxxx" in the Interrupt vector denotes that software/BIOS initializes them and the MCH will not modify any of the "x" bits since it handles only 1 message vector that is common to all events

4.8.27.14 PEXCAPID: PCI Express Capability ID Register

Device: 8
Function: 0
Offset: 6Ch

Bit	Attr	Default	Description
7:0	RO	10h	**CAPID: PCI Express Capability ID** This code denotes the standard PCI Express capability.

4.8.27.15 PEXNPTR: PCI Express Next Pointer Register

Device: 8
Function: 0
Offset: 6Dh

Bit	Attr	Default	Description
7:0	RO	00h	**NXTPTR: PCI Express Next Pointer** The PCI Express capability structure is the last capability in the linked list and set to NULL.

4.8.27.16 PEXCAP: PCI Express Capabilities Register

Device: 8
Function: 0
Offset: 6Eh

Bit	Attr	Default	Description
15:14	RV	0h	*Reserved*
13:9	RO	0h	**IMN: Interrupt Message Number**: This field indicates the interrupt message number that is generated from the Intel® QuickData Technology device. When there are more than one MSI interrupt Number, this register field is required to contain the offset between the base Message Data and the MSI Message that is generated when the status bits in the slot status register or root port status registers are set.
8	RO	0	**Slot_Impl: Slot Implemented**: Intel® QuickData Technology is an integrated device and therefore a slot is never implemented.
7:4	RO	1001	**DPT: Device/Port Type**: Intel® QuickData Technology device represents a Root Complex Integrated Endpoint.
3:0	RO	0001	VERS: **Capability Version**: Intel® QuickData Technology supports revision 1 of the PCI Express specification.

4.8.27.17 PEXDEVCAP: Device Capabilities Register

Device: 8
Function: 0
Offset: 70h

Bit	Attr	Default	Description
31:28	RV	0h	*Reserved*
27:26	RO	00	**CSPLS: Captured Slot Power Limit Scale** This field applies only to upstream ports. Hardwired to 0h
25:18	RO	00h	**CSPLV: Captured Slot Power Limit Value** This field applies only to upstream ports. Hardwired to 0h
17:15	RV	0h	*Reserved*

Device: 8 Function: 0 Offset: 70h			
Bit	**Attr**	**Default**	**Description**
14	RO	0	**PIPD: Power Indicator Present** The Intel® QuickData Technology is an integrated device and therefore, an Power Indicator does not exist. Hardwired to 0h
13	RO	0	**AIPD: Attention Indicator Present** The Intel® QuickData Technology is an integrated device and therefore, an Attention Indicator does not exist. Hardwired to 0h
12	RO	0	**ABPD: Attention Button Present** The Intel® QuickData Technology is an integrated device and therefore, an Attention Button does not exist. Hardwired to 0h
11:9	RO	000	**EPL1AL: Endpoint L1 Acceptable Latency** The Intel® QuickData Technology device is not implemented on a physical PCI Express link and therefore, this value is irrelevant. Hardwired to 0h
8:6	RO	000	**EPL0AL: Endpoint L0s Acceptable Latency** The Intel® QuickData Technology device is not implemented on a physical PCI Express link and therefore, this value is irrelevant. Hardwired to 0h
5	RO	0	**ETFS: Extended Tag Field Supported** The Intel® QuickData Technology device does not support extended tags. Hardwired to 0h
4:3	RO	00	**PFS: Phantom Functions Supported** The Intel® QuickData Technology device does not support Phantom Functions. Hardwired to 0h
2:0	RO	000	**MPLSS: Max_Payload_Size Supported** This field indicates the maximum payload size that the Intel® QuickData Technology integrated device can support. **000: 128B max payload size** others- *Reserved*

4.8.27.18 PEXDEVCTRL: Device Control Register

Device: 8 Function: 0 Offset: 74h			
Bit	**Attr**	**Default**	**Description**
15	RV	0	*Reserved*
14:12	RO	000	**MRRS: Max_Read_Request_Size** Since the Intel® QuickData Technology device does not issue read requests on a PCI Express interface, this field is irrelevant. Hardwired to 0h
11	RW	1	**ENNOSNP: Enable No Snoop** 1: Setting this bit enables the Intel® QuickData Technology device to issue requests with the No Snoop attribute. 0: Clearing this bit behaves as a global disable when the corresponding capability is enabled for source/destination snoop control in the DMA's descriptor's Desc_Control field.
10	RO	0	**APPME: Auxiliary Power PM Enable** The Intel® QuickData Technology device does not implement auxiliary power so setting this bit has no effect. Hardwired to 0h
9	RO	0	**PFEN: Phantom Functions Enable** The Intel® QuickData Technology device does not implement phantom functions so setting this bit has no effect. Hardwired to 0h
8	RO	0	**ETFEN: Extended Tag Field Enable**: The Intel® QuickData Technology device does not implement extended tags so setting this bit has no effect.

Device: 8 Function: 0 Offset: 74h			
Bit	**Attr**	**Default**	**Description**
7:5	RW	000	**MPS: Max_Payload_Size**: The Intel® QuickData Technology device must not generate packets on any PCI Express interface which exceeds the length allowed with this field. 000: 128B max payload size 001: 256B max payload size 010: 512B max payload size 011: 1024B max payload size 100: 2048B max payload size 101: 4096B max payload size **Note** that this field has no impact internally to the MCH and the maximum payload size of the TLPs that appear on the PCI Express port is governed by the PEXDEVCTRL.MPS for that port defined in Section 4.8.13.4, "PEXDEVCTRL[7:0]: PCI Express Device Control Register" on page 150
4	RO	0	**ENRORD: Enable Relaxed Ordering** No relaxed ordering is supported by the MCH. Hardwired to 0h.
3	RO	0	**URREN: Unsupported Request Reporting Enable** For an integrated Intel® QuickData Technology device, this bit is irrelevant. Hardwired to 0h
2	RW	0	**FERE: Fatal Error Reporting Enable**: This bit controls the reporting of fatal errors internal to the Intel® QuickData Technology device 0: Fatal error reporting is disabled 1: Fatal error reporting is enabled
1	RW	0	**NFERE: Non-Fatal Error Reporting Enable** This bit controls the reporting of non fatal errors internal to the Intel® QuickData Technology device in the PCI Express port. 0: Non Fatal error reporting is disabled 1: Non Fatal error reporting is enabled This has no effect on the MCH Intel® QuickData Technology device as it does not report any non-fatal errors.
0	RW	0	**CERE: Correctable Error Reporting Enable** This bit controls the reporting of correctable errors internal to the Intel® QuickData Technology device in the PCI Express port. 0: Correctable error reporting is disabled 1: Correctable Fatal error reporting is enabled This has no effect on the MCH Intel® QuickData Technology device as it does not report any correctable errors.

4.8.27.19 PEXDEVSTS: PCI Express Device Status Register

Device: 8
Function: 0
Offset: 76h

Bit	Attr	Default	Description
15:6	RV	0h	*Reserved*
5	RO	0	**TP: Transactions Pending** This bit indicates that the Intel® QuickData Technology device has issued non-posted PCI Express transactions which have not yet completed. Note the MCH Intel® QuickData Technology device does not issue any NP transactions and hence this is hardwired to zero.
4	RO	0	**APD: AUX Power Detected** The Intel® QuickData Technology device does not support AUX power. Hardwired to 0h.
3	RO	0	**URD: Unsupported Request Detected** This does not apply to Intel® QuickData Technology in BF as there are no messages for the DMA engine. Hardwired to 0h
2	RWC	0	**FED: Fatal Error Detected** This bit gets set if a fatal uncorrectable error is detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register (See FERE in Section 4.8.27.18, "PEXDEVCTRL: Device Control Register" on page 277) 1: Fatal errors detected 0: No Fatal errors detected
1	RWC	0	**NFED: Non-Fatal Error Detected** This bit gets set if a non-fatal uncorrectable error is detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. (See NFERE in Section 4.8.27.18, "PEXDEVCTRL: Device Control Register" on page 277) 1: Non Fatal errors detected 0: No non-Fatal Errors detected
0	RWC	0	**CED: Correctable Error Detected** This bit gets set if a correctable error is detected. Errors are logged in this register regardless of whether error reporting is enabled or not in the Device Control register. (See CERE in Section 4.8.27.18, "PEXDEVCTRL: Device Control Register" on page 277) 1: correctable errors detected 0: No correctable errors detected

§

5 Ball Assignment

5.1 Intel® 7300 Chipset MCH Ball Assignment

.

Figure 5-1. Intel® 7300 Chipset Quadrant Map

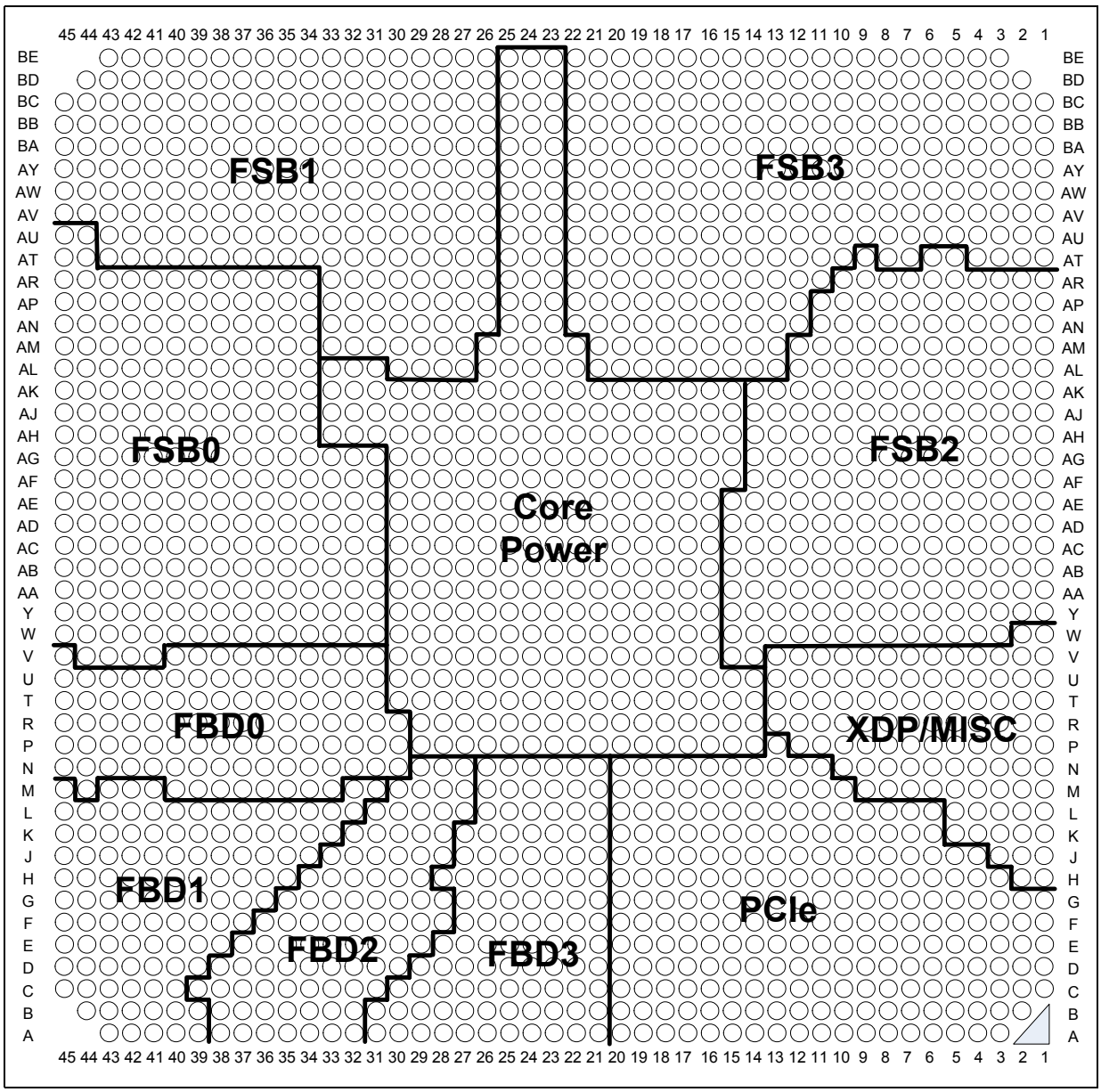


Figure 5-2. Intel® 7300 Chipset MCH Ball Assignemnt Left Side (Top View)

	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	
BE			VSS	VSS	VTT	VSS	FSB1A_N[19]	FSB1ODTCRES	VSS	FSB1RS_N[0]	FSB1RS_N[1]	VTT	FSB1DP_N[2]	FSB1DP_N[1]	VSS	BE
BD		VSS	VSS	VTT	VTT	FSB1A_N[24]	FSB1A_N[39]	VSS	FSB1SLWCRES	FSB1BNR_N	VSS	FSB1DEFER_N	FSB1HIT_N	VSS	FSB1D_N[42]	BD
BC	VSS	VSS	VTT	VSS	FSB1A_N[25]	FSB1A_N[28]	VSS	FSB1DRVCRES	FSB1DBSY_N	VTT	FSB1TRDY_N	FSB1HITM_N	VSS	FSB1DP_N[0]	FSB1D_N[43]	BC
BB	VSS	VTT	VTT	FSB1A_N[23]	FSB1A_N[29]	VSS	FSB1A_N[18]	FSB1A_N[17]	VSS	FSB1RESET_N	FSB1ADS_N	VSS	FSB1BPRI_N	FSB1DP_N[3]	VTT	BB
BA	VTT	VSS	FSB1A_N[21]	FSB1A_N[22]	VSS	FSB1ADSTB_N[1]	FSB1A_N[30]	VTT	FSB1VREF[0]	FSB1LOCK_N	VSS	FSB1BREQ1_N	FSB1RS_N[2]	VSS	FSB1D_N[45]	BA
AY	VTT	FSB1A_N[26]	FSB1A_N[34]	VSS	FSB1A_N[20]	FSB1REQ_N[3]	VSS	FSB1A_N[3]	RSVD	VSS	FSB1BREQ0_N	FSB1DRDY_N	VTT	FSB1VREF[4]	FSB1D_N[44]	AY
AW	FSB1A_N[27]	FSB1A_N[31]	VSS	FSB1A_N[35]	FSB1A_N[7]	VTT	FSB1REQ_N[1]	FSB1REQ_N[2]	VSS	FSB1BPM_N[4]	FSB1AP_N[0]	VSS	FSB1BINIT_N	FSB1D_N[47]	VSS	AW
AV	FSB1A_N[38]	VSS	FSB1A_N[33]	FSB1A_N[5]	VSS	FSB1A_N[6]	FSB1REQ_N[0]	VSS	FSB1REQ_N[4]	FSB1A_N[37]	VTT	FSB1AP_N[1]	FSB1D_N[62]	VSS	FSB1D_N[46]	AV
AU	VSS	FSB0D_N[4]	FSB1A_N[32]	VTT	FSB1A_N[9]	FSB1ADSTB_N[0]	VSS	FSB1A_N[8]	FSB1A_N[4]	VSS	FSB1RSP_N	FSB1MCERR_N	VSS	FSB1D_N[54]	FSB1D_N[53]	AU
AT	FSB0D_N[0]	FSB0D_N[7]	VSS	FSB1A_N[36]	FSB1A_N[11]	VSS	FSB1A_N[15]	FSB1A_N[13]	VTT	FSB1A_N[14]	FSB1BPM_N[5]	VSS	FSB1DBI_N[3]	FSB1D_N[55]	VSS	AT
AR	FSB0D_N[3]	VTT	FSB0D_N[8]	FSB0D_N[1]	VSS	FSB1A_N[16]	FSB1A_N[12]	VSS	FSB1A_N[10]	FSB0D_N[17]	VSS	FSB1D_N[61]	FSB1D_N[56]	VTT	FSB1D_N[51]	AR
AP	VSS	FSB0D_N[2]	FSB0D_N[6]	VSS	FSB0DBI_N[0]	FSB0D_N[20]	VTT	FSB0D_N[21]	FSB0D_N[19]	VSS	FSB0D_N[16]	FSB1D_N[57]	VSS	FSB1STBP_N[3]	FSB1STBN_N[3]	AP
AN	FSB0D_N[13]	FSB0D_N[10]	VSS	FSB0D_N[5]	FSB0D_N[9]	VSS	FSB0D_N[23]	FSB0D_N[22]	VSS	FSB0D_N[27]	FSB0D_N[18]	VTT	FSB1D_N[59]	FSB1D_N[58]	VSS	AN
AM	FSB0D_N[11]	VSS	FSB0STBN_N[0]	FSB0STBP_N[0]	VTT	FSB0DBI_N[1]	FSB0VREF[2]	VSS	FSB0D_N[29]	FSB0D_N[26]	VSS	FSB1D_N[60]	FSB1D_N[63]	VSS	FSB1D_N[52]	AM
AL	VSS	FSB0D_N[12]	FSB0D_N[14]	VSS	FSB0D_N[15]	FSB0D_N[28]	VSS	FSB0D_N[25]	FSB0D_N[30]	VTT	FSB0STBN_N[1]	FSB0D_N[31]	VSS	VCCSF	VCCSF	AL
AK	FSB0STBP_N[2]	FSB0STBN_N[2]	VTT	FSB0D_N[37]	FSB0D_N[32]	VSS	FSB0D_N[36]	FSB0D_N[24]	VSS	FSB0D_N[52]	FSB0STBP_N[1]	VSS	VCCSF	VCCSF	VSS	AK
AJ	FSB0D_N[38]	VSS	FSB0D_N[35]	FSB0D_N[39]	VSS	FSB0D_N[33]	FSB0D_N[34]	VTT	FSB0STBN_N[3]	FSB0STBP_N[3]	VSS	FSB0D_N[48]	VCCSF	VSS	VCCSF	AJ
AH	VTT	FSB0D_N[40]	FSB0D_N[43]	VSS	FSB0D_N[41]	FSB0DBI_N[2]	VSS	FSB0DP_N[0]	FSB0D_N[53]	VSS	FSB0D_N[55]	FSB0D_N[51]	VSS	VCCSF	VCCSF	AH
AG	FSB0D_N[42]	FSB0D_N[45]	VSS	FSB0D_N[44]	FSB0D_N[47]	VTT	FSB0D_N[46]	FSB0VREF[4]	VSS	FSB0D_N[54]	FSB0DBI_N[3]	VTT	FSB0D_N[49]	FSB0D_N[50]	VSS	AG
AF	FSB0HITM_N	VSS	FSB0RS_N[1]	FSB0DEFER_N	VSS	FSB0HIT_N	FSB0DP_N[2]	VSS	FSB0D_N[62]	FSB0D_N[56]	VSS	FSB0D_N[61]	FSB0D_N[63]	VSS	VSS	AF
AE	VSS	FSB0RS_N[0]	FSB0BNR_N	VTT	FSB0RS_N[2]	FSB0BPRI_N	VSS	FSB0DP_N[3]	FSB0DP_N[1]	VTT	FSB0D_N[58]	FSB0D_N[57]	VSS	FSB0D_N[60]	VSS	AE
AD	FSB0RESET_N	FSB0TRDY_N	VSS	FSB0DRDY_N	FSB0BINIT_N	VSS	FSB0MCERR_N	FSB0BPM_N[5]	VSS	FSB0BREQ1_N	FSB0D_N[59]	VSS	FSB0A_N[3]	VSS	VTT	AD
AC	FSB0DBSY_N	VTT	FSB0ADS_N	FSB0LOCK_N	VSS	FSB0BPM_N[4]	FSB0RSP_N	VTT	FSB0VREF[0]	FSB0BREQ0_N	VSS	FSB0REQ_N[3]	FSB0REQ_N[2]	VSS	VSS	AC
AB	VSS	FSB0AP_N[0]	FSB0AP_N[1]	VSS	FSB0A_N[24]	FSB0A_N[18]	VSS	FSB0A_N[17]	FSB0A_N[5]	VSS	FSB0A_N[6]	FSB0REQ_N[0]	VTT	FSB0A_N[37]	VSS	AB
AA	FSB0A_N[30]	FSB0A_N[23]	VSS	FSB0A_N[29]	FSB0A_N[25]	VTT	FSB0A_N[19]	FSB0A_N[39]	VSS	FSB0A_N[9]	FSB0ADSTB_N[0]	VSS	FSB0REQ_N[1]	FSB0REQ_N[4]	VSS	AA
Y	FSB0A_N[20]	VSS	FSB0A_N[21]	FSB0A_N[22]	VSS	FSB0A_N[28]	FSB0ADSTB_N[1]	VSS	FSB0A_N[36]	FSB0A_N[8]	VTT	FSB0A_N[7]	FSB0A_N[4]	VSS	VSS	Y
W	VSS	FSB0A_N[26]	FSB0A_N[35]	VTT	FSB0A_N[34]	FSB0A_N[27]	VSS	FSB0A_N[16]	FSB0A_N[15]	VSS	FSB0A_N[12]	FSB0A_N[11]	VSS	FSB0A_N[14]	VSS	W
V	FBD0SBON[9]	FSB0A_N[31]	VSS	FSB0A_N[33]	FSB0A_N[38]	VSS	FBD0SBON[8]	FSB0A_N[13]	VCCFBD	VSS	FSB0A_N[10]	VSS	FBD0NBIN[0]	FBD0NBIP[0]	VSS	V
U	FBD0SBOP[9]	VCCFBD	FBD0SBON[4]	FSB0A_N[32]	VSS	FBD0SBON[7]	FBD0SBOP[8]	VSS	VSS	VSS	VSS	FBD0NBIN[1]	FBD0NBIP[1]	VCCFBD	VSS	U
T	VSS	FBD0SBON[3]	FBD0SBOP[4]	VSS	FBD0SBON[6]	FBD0SBOP[7]	VCCFBD	VSS	VSS	VSS	VSS	VSS	VSS	VCCFBD	VSS	T
R	FBD0SBON[2]	FBD0SBOP[3]	VSS	FBD0SBON[5]	FBD0SBOP[6]	VSS	FBD0NBIN[11]	VSS	VSS	FBD0NBIN[6]	FBD0NBIP[6]	VCCFBD	FBD0NBIN[4]	FBD0NBIP[4]	VSS	R
P	FBD0SBOP[2]	VSS	FBD0SBON[1]	FBD0SBOP[5]	VCCFBD	FBD0NBIN[10]	FBD0NBIP[11]	VSS	FBD0NBIN[9]	FBD0NBIP[9]	VSS	FBD0NBIN[5]	FBD0NBIP[5]	VSS	FBD0NBIN[2]	P
N	VSS	FBD0SBON[0]	FBD0SBOP[1]	VSS	FBD1SBON[6]	FBD0NBIP[10]	VSS	FBD0NBIN[7]	FBD0NBIP[7]	VCCFBD	FBD0NBIN[13]	FBD0NBIP[13]	VSS	FBD0NBIN[3]	FBD0NBIP[3]	N
M	FBD1SBON[9]	FBD0SBOP[0]	VCCFBD	FBD1SBON[5]	FBD1SBOP[6]	VSS	FBD0NBIN[8]	FBD0NBIP[8]	VSS	FBD0NBIN[12]	FBD0NBIP[12]	VSS	FBD1NBIN[0]	FBD1NBIP[0]	VCCFBD	M
L	FBD1SBOP[9]	VSS	FBD1SBON[4]	FBD1SBOP[5]	VSS	FBD1SBOP[8]	FBD1SBON[8]	VCCFBD	VSS	VSS	VSS	FBD1NBIN[1]	FBD1NBIP[1]	VSS	FBD2NBIP[2]	L
K	VCCFBD	FBD1SBON[3]	FBD1SBOP[4]	VSS	FBD1SBOP[7]	FBD1SBON[7]	VSS	VSS	VSS	VSS	FBD1NBIN[2]	FBD1NBIP[2]	VCCFBD	VSS	VSS	K
J	FBD01CLKN	FBD1SBOP[3]	VSS	FBD1SBON[2]	FBD1SBOP[2]	VCCFBD	VSS	VSS	VSS	FBD1NBIN[3]	FBD1NBIP[3]	VSS	VSS	VSS	VSS	J
H	FBD01CLKP	VSS	FBD1SBON[1]	FBD1SBOP[1]	VSS	VSS	VSS	VSS	FBD1NBIN[4]	FBD1NBIP[4]	VCCFBD	FBD2SBON[8]	FBD2SBOP[8]	VSS	FBD2NBIP[12]	H
G	VSS	FBD1SBON[0]	FBD1SBOP[0]	VCCFBD	FBD1NBIN[9]	FBD1NBIP[9]	VSS	FBD1NBIN[5]	FBD1NBIP[5]	VSS	FBD2SBON[7]	FBD2SBOP[7]	VSS	VSS	FBD2NBIN[12]	G
F	FBD01VCCA	FBD01VSSA	VSS	FBD1NBIN[11]	FBD1NBIP[11]	VSS	FBD1NBIN[13]	FBD1NBIP[13]	VCCFBD	FBD2SBON[6]	FBD2SBOP[6]	VSS	FBD2NBIN[10]	FBD2NBIP[10]	VSS	F
E	VCCFBD	VCCFBD	FBD1NBIN[10]	FBD1NBIP[10]	VSS	FBD1NBIN[12]	FBD1NBIP[12]	VSS	FBD2SBON[5]	FBD2SBOP[5]	VSS	FBD2NBIN[11]	FBD2NBIP[11]	VCCFBD	FBD2NBIN[8]	E
D	VSS	VCCFBD	RSVD	VSS	FBD1NBIN[7]	FBD1NBIP[7]	VCCFBD	FBD2SBON[9]	FBD2SBOP[9]	VSS	FBD2SBON[1]	FBD2SBOP[1]	VSS	FBD2NBIN[9]	FBD2NBIP[9]	D
C	VSS	VSS	VCCFBD	FBD1NBIN[8]	FBD1NBIP[8]	VSS	FBD2SBON[4]	FBD2SBOP[4]	VSS	FBD2SBON[3]	FBD2SBOP[3]	VCCFBD	VSS	VSS	VSS	C
B		VSS	VSS	VCCFBD	VCCFBD	FBD1NBIN[6]	FBD1NBIP[6]	VSS	FBD2SBON[2]	FBD2SBOP[2]	VSS	VSS	VSS	VSS	FBD3SBON[5]	B
A			VSS	VSS	VCCFBD	FBD01RESIN	VSS	FBD01BGBIASEX	FBD01COMPBIA	VSS	FBD2SBON[0]	FBD2SBOP[0]	VSS	FBD23CLKN	FBD23CLKP	A
	45	44	43	42	41	40	39	38	37	36	35	34	33	32	31	

Figure 5-3. Intel® 7300 Chipset MCH Ball Assignment Center (Top View)

	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	
BE	FSB1D_N[40]	FSB1D_N[38]	VSS	FSB1D_N[9]	FSB1D_N[3]	VSS	FSBVCCA	COREVSSA	VSS	FSB3D_N[61]	FSB3DBI_N[3]	VSS	FSB3STBP_N[3]	FSB3D_N[25]	VTT	BE
BD	FSB1D_N[41]	VTT	FSB1D_N[10]	FSB1D_N[5]	VSS	FSB1D_N[0]	COREVCCA	VSS	CORECLKN	FSB3D_N[56]	VSS	FSB3D_N[55]	FSB3STBN_N[3]	VSS	FSB3D_N[26]	BD
BC	VSS	FSB1D_N[35]	FSB1D_N[39]	VSS	FSB1D_N[8]	FSB1D_N[7]	VSS	RSVD	CORECLKP	VSS	FSB3D_N[62]	FSB3D_N[53]	VTT	FSB3D_N[30]	FSB3D_N[28]	BC
BB	FSB1STBP_N[2]	FSB1DBI_N[2]	VSS	FSB1D_N[2]	FSB1D_N[6]	VSS	VSS	VSS	VSS	FSB3D_N[58]	FSB3D_N[54]	VSS	FSB3D_N[48]	FSB3D_N[24]	VSS	BB
BA	FSB1STBN_N[2]	VSS	FSB1D_N[11]	FSB1D_N[13]	VTT	FSB1D_N[1]	VSS	VSS	FSB3D_N[63]	FSB3D_N[57]	VTT	FSB3D_N[51]	FSB3D_N[52]	VSS	FSB3STBN_N[1]	BA
AY	VSS	FSB1D_N[37]	FSB1D_N[33]	VSS	FSB1STBN_N[0]	FSB1D_N[4]	VSS	VSS	FSB3D_N[60]	VSS	FSB3D_N[59]	FSB3D_N[49]	VSS	FSB3D_N[29]	FSB3STBP_N[1]	AY
AW	FSB1D_N[32]	FSB1D_N[34]	VTT	FSB1D_N[12]	FSB1STBP_N[0]	VSS	VSS	VSS	VSS	FSB3D_N[43]	FSB3DBI_N[2]	VSS	FSB3D_N[50]	FSB3DBI_N[1]	VSS	AW
AV	FSB1D_N[36]	VSS	FSB1D_N[15]	FSB1D_N[14]	VSS	FSB1DBI_N[0]	VCCSF	VSS	VCCSF	FSB3D_N[40]	VSS	FSB3D_N[35]	FSB3VREF[4]	VTT	FSB3D_N[31]	AV
AU	VTT	FSB1VREF[2]	FSB1D_N[27]	VSS	FSB1D_N[18]	FSB1D_N[17]	VSS	VCCSF	VSS	VSS	FSB3D_N[38]	FSB3D_N[39]	VSS	FSB3D_N[11]	FSB3D_N[13]	AU
AT	FSB1D_N[28]	FSB1D_N[26]	VSS	FSB1D_N[22]	FSB1D_N[21]	VSS	VCCSF	VCCSF	VCCSF	FSB3D_N[42]	FSB3D_N[41]	VTT	FSB3D_N[37]	FSB3D_N[10]	VSS	AT
AR	FSB1D_N[48]	VSS	FSB1D_N[29]	FSB1D_N[23]	VTT	FSB1D_N[19]	VCCSF	VCCSF	VCCSF	FSB3D_N[45]	VSS	FSB3STBP_N[2]	FSB3D_N[33]	VSS	FSB3D_N[5]	AR
AP	VSS	FSB1D_N[30]	FSB1D_N[25]	VSS	FSB1D_N[20]	FSB1D_N[16]	VCCSF	VCCSF	VCCSF	VTT	FSB3D_N[44]	FSB3STBN_N[2]	VSS	FSB3D_N[12]	FSB3D_N[14]	AP
AN	FSB1D_N[49]	FSB1D_N[24]	VTT	FSB1STBN_N[1]	FSB1DBI_N[1]	VSS	VCCSF	VCCSF	VCCSF	VSS	FSB3D_N[47]	VSS	FSB3D_N[32]	FSB3D_N[34]	VTT	AN
AM	FSB1D_N[50]	VSS	FSB1D_N[31]	FSB1STBP_N[1]	VSS	VCCSF	VCCSF	VCCSF	VCCSF	VCCSF	VSS	FSB3D_N[46]	FSB3D_N[36]	VSS	FSB3D_N[15]	AM
AL	VTT	VSS	VSS	VSS	VSS	VCCSF	VCCSF	VCCSF	VCCSF	VCCSF	VSS	VSS	VTT	VSS	VSS	AL
AK	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	AK
AJ	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	AJ
AH	VTT	VTT	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VTT	VTT	AH
AG	VTT	VTT	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VTT	VTT	AG
AF	VTT	VTT	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VTT	VTT	AF
AE	VTT	VTT	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VTT	VTT	AE
AD	VTT	VTT	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VTT	VTT	AD
AC	VTT	VTT	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VTT	VTT	AC
AB	VTT	VTT	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VTT	VTT	AB
AA	VTT	VTT	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VTT	VTT	AA
Y	VTT	VTT	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VTT	VTT	Y
W	VTT	VTT	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VTT	VTT	W
V	VTT	VTT	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	V
U	VSS	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCCSPL	VSS	VCC	U
T	VSS	VSS	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCCPE	VSS	VSS	VSS	T
R	VCCFBD	VCCFBD	VSS	VSS	VCCFBD	VSS	VCCFBD	VSS	VCCFBD	VSS	VSS	VSS	VCCPE	VSS	VSS	R
P	FBD0NBIP[2]	VCCFBD	VCCFBD	VSS	VSS	VSS	VCCFBD	VSS	VSS	VSS	VSS	VCCPE	VCCPE	VCCPE	VCCPE	P
N	VSS	FBD2NBIP[1]	FBD2NBIN[1]	VSS	FBD3NBIP[0]	FBD3NBIN[0]	VCCFBD	FBD3NBIP[2]	FBD3NBIN[2]	VSS	PEICOMPI	PERCOMPO	VSS	PE7RN[3]	VCCPE	N
M	FBD2NBIP[0]	FBD2NBIN[0]	VSS	FBD2NBIP[3]	VSS	VSS	FBD3NBIP[3]	FBD3NBIN[3]	VSS	FBD3NBIP[4]	VSS	VCCPE	VCCPE	PE7RP[3]	VSS	M
L	FBD2NBIN[2]	VSS	FBD2NBIP[4]	FBD2NBIN[3]	VCCFBD	FBD3NBIP[1]	FBD3NBIN[1]	VSS	FBD3NBIP[5]	FBD3NBIN[4]	VSS	PE4TP[0]	VCCPE	VSS	PE6TP[0]	L
K	VSS	FBD2NBIP[5]	FBD2NBIN[4]	VSS	VSS	VSS	VSS	FBD3NBIP[13]	FBD3NBIN[5]	VCCFBD	PE4RP[0]	PE4TN[0]	VSS	PE5RN[3]	PE6TN[0]	K
J	FBD2NBIP[13]	FBD2NBIN[5]	VCCFBD	VSS	VSS	VSS	FBD3NBIP[12]	FBD3NBIN[13]	VSS	VSS	PE4RN[0]	VSS	PE5TN[3]	PE5RP[3]	VCCPE	J
H	FBD2NBIN[13]	VSS	VSS	VSS	VSS	VSS	FBD3NBIN[12]	VCCFBD	FBD3NBIN[6]	FBD3NBIP[6]	VSS	PE4TP[1]	PE5TP[3]	VSS	PE6RP[0]	H
G	VCCFBD	FBD2NBIN[6]	FBD2NBIP[6]	VSS	FBD3NBIN[10]	FBD3NBIP[10]	VSS	FBD3NBIN[7]	FBD3NBIP[7]	VSS	PE4RP[1]	PE4TN[1]	VCCPE	PE5RN[2]	PE6RN[0]	G
F	FBD2NBIN[7]	FBD2NBIP[7]	VSS	FBD3SBON[8]	FBD3SBOP[8]	VCCFBD	FBD3NBIN[9]	FBD3NBIP[9]	VSS	VSS	PE4RN[1]	VSS	PE4TP[2]	PE5RP[2]	VSS	F
E	FBD2NBIP[8]	VSS	FBD3SBON[7]	FBD3SBOP[7]	VSS	FBD3NBIN[11]	FBD3NBIP[11]	VSS	FBD3NBIN[8]	FBD3NBIP[8]	VCCPE	PE4RP[2]	PE4TN[2]	VSS	PE6RP[1]	E
D	VSS	FBD3SBON[6]	FBD3SBOP[6]	VCCFBD	FBD3SBON[2]	FBD3SBOP[2]	VSS	VSS	VSS	VSS	PE4TP[3]	PE4RN[2]	VSS	PE5TP[2]	PE5TN[2]	D
C	FBD3SBON[9]	FBD3SBOP[9]	VSS	FBD3SBON[4]	FBD3SBOP[4]	VSS	VSS	VSS	VCCFBD	VSS	PE4TN[3]	VSS	PE5RP[0]	PE5RN[0]	VSS	C
B	FBD3SBOP[5]	VCCFBD	FBD3SBON[3]	FBD3SBOP[3]	VSS	FBD3SBON[0]	FBD3SBOP[0]	VSS	FBD23BGBIASEXT	FBD23ICOMPBIAS	VSS	PE5TP[0]	PE5TN[0]	VCCPE	PE5RP[1]	B
A	VSS	FBD23VSSA	FBD23VCCA	VSS	FBD3SBON[1]	FBD3SBOP[1]	VCCFBD	RSVD	FBD23RESIN	VSS	PE4RP[3]	PE4RN[3]	VSS	PE5TP[1]	PE5TN[1]	A
	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	

Figure 5-4. Intel® 7300 Chipset MCH Ball Assignemnt Right (Top View)

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	
BE	FSB3D_N[27]	FSB3DP_N[3]	VSS	FSB3MCERR_N	FSB3AP_N[0]	VSS	FSB3A_N[32]	FSB3A_N[33]	VSS	FSB3A_N[26]	VTT	VSS	VSS			BE
BD	FSB3D_N[23]	VSS	FSB3DP_N[1]	FSB3BPM_N[4]	VTT	FSB3LOCK_N	FSB3A_N[38]	VSS	FSB3A_N[27]	FSB3A_N[34]	VSS	VTT	VSS	VSS		BD
BC	VSS	FSB3D_N[20]	FSB3DP_N[0]	VSS	FSB3BINIT_N	FSB3DRDY_N	VSS	FSB3A_N[31]	FSB3A_N[35]	VTT	FSB3A_N[22]	FSB3A_N[30]	VTT	VSS	VSS	BC
BB	FSB3D_N[22]	FSB3D_N[21]	VTT	FSB3AP_N[1]	FSB3DBSY_N	VSS	FSB3SLWCRES	FSB3A_N[20]	VSS	FSB3A_N[21]	FSB3A_N[23]	VSS	FSB3A_N[29]	VTT	VSS	BB
BA	FSB3D_N[18]	VSS	FSB3DP_N[2]	FSB3ADS_N	VSS	FSB3TRDY_N	FSB3ODTCRES	VTT	FSB3ADSTB_N[1]	FSB3A_N[25]	VSS	FSB3A_N[28]	FSB3A_N[24]	VSS	VTT	BA
AY	VTT	FSB3D_N[19]	FSB3BPM_N[5]	VSS	FSB3RS_N[0]	FSB3DRVCRES	VSS	FSB3A_N[17]	FSB3A_N[39]	VSS	FSB3A_N[19]	FSB3A_N[18]	VTT	FSB3A_N[15]	FSB3A_N[11]	AY
AW	FSB3D_N[17]	FSB3D_N[16]	VSS	FSB3BNR_N	FSB3RS_N[1]	VTT	RSVD	FSB3A_N[10]	VSS	FSB3A_N[13]	FSB3A_N[12]	VSS	FSB3A_N[36]	FSB3A_N[8]	VSS	AW
AV	FSB3D_N[9]	VSS	FSB3RSP_N	FSB3RS_N[2]	VSS	FSB3VREF[0]	FSB3A_N[14]	VSS	FSB3REQ_N[2]	FSB3A_N[16]	VTT	FSB3A_N[9]	FSB3A_N[5]	VSS	FSB3ADSTB_N[0]	AV
AU	VSS	FSB3VREF[2]	FSB3HITM_N	VTT	FSB3DEFER_N	FSB3BPRI_N	VSS	FSB3REQ_N[4]	FSB3A_N[3]	VSS	FSB3A_N[4]	FSB3REQ_N[0]	VSS	FSB3A_N[6]	FSB3A_N[7]	AU
AT	FSB3D_N[2]	FSB3D_N[3]	VSS	FSB3RESET_N	FSB3HIT_N	VSS	FSB2D_N[47]	FSB3A_N[37]	VTT	FSB2D_N[60]	FSB2D_N[59]	VSS	FSB3REQ_N[1]	FSB3REQ_N[3]	VSS	AT
AR	FSB3D_N[6]	VTT	FSB3D_N[0]	FSB3BREQ0_N	VSS	FSB2D_N[46]	FSB2D_N[44]	VSS	FSB2D_N[45]	FSB2D_N[63]	VSS	FSB2D_N[58]	FSB2D_N[61]	VTT	FSB2D_N[62]	AR
AP	VSS	FSB3D_N[8]	FSB3BREQ1_N	VSS	FSB2D_N[41]	FSB2D_N[42]	VTT	FSB2D_N[40]	FSB2D_N[43]	VSS	FSB2D_N[57]	FSB2D_N[56]	VSS	FSB2DBI_N[3]	FSB2D_N[54]	AP
AN	FSB3STBN_N[0]	FSB3D_N[7]	VSS	VSS	FSB2D_N[38]	VSS	FSB2DBI_N[2]	FSB2VREF[4]	VSS	FSB2STBN_N[3]	FSB2STBP_N[3]	VTT	FSB2D_N[55]	FSB2D_N[53]	VSS	AN
AM	FSB3STBP_N[0]	VSS	FSB3D_N[4]	FSB2D_N[33]	VTT	FSB2STBP_N[2]	FSB2D_N[35]	VSS	FSB2D_N[50]	FSB2D_N[49]	VSS	FSB2D_N[52]	FSB2D_N[48]	VSS	FSB2D_N[51]	AM
AL	VSS	FSB3D_N[1]	FSB3DBI_N[0]	VSS	FSB2D_N[32]	FSB2STBN_N[2]	VSS	FSB2D_N[39]	FSB2D_N[31]	VTT	FSB2D_N[24]	FSB2D_N[30]	VSS	FSB2D_N[28]	FSB2D_N[25]	AL
AK	VCCSF	VCCSF	VTT	FSB2D_N[36]	FSB2D_N[34]	VSS	FSB2D_N[37]	FSB2D_N[16]	VSS	FSB2D_N[17]	FSB2D_N[19]	VSS	FSB2D_N[26]	FSB2D_N[29]	VTT	AK
AJ	VCCSF	VSS	FSB2D_N[15]	FSB2D_N[12]	VSS	FSB2STBP_N[0]	FSB2STBN_N[0]	VSS	FSB2STBP_N[1]	FSB2STBN_N[1]	VSS	FSB2DBI_N[1]	FSB2D_N[23]	VSS	FSB2D_N[27]	AJ
AH	VTT	VCCSF	FSB2D_N[14]	VSS	FSB2D_N[11]	FSB2D_N[10]	VSS	FSB2DP_N[2]	FSB2RESET_N	VSS	FSB2D_N[18]	FSB2D_N[21]	VTT	FSB2D_N[22]	FSB2D_N[20]	AH
AG	VCCSF	FSB2D_N[13]	VSS	FSB2D_N[5]	FSB2D_N[2]	VTT	FSB2D_N[9]	FSB2VREF[2]	VSS	FSB2DP_N[1]	FSB2BPM_N[5]	VSS	FSB2RSP_N	FSB2MCERR_N	VSS	AG
AF	VCCSF	VSS	FSB2D_N[1]	FSB2D_N[0]	VSS	FSB2D_N[3]	FSB2D_N[6]	VSS	FSB2DP_N[0]	FSB2BINIT_N	VTT	FSB2AP_N[0]	FSB2BPM_N[4]	VSS	FSB2AP_N[1]	AF
AE	VSS	FSB2DBI_N[0]	FSB2D_N[4]	VTT	FSB2D_N[7]	FSB2D_N[8]	VSS	FSB2DP_N[3]	FSB2HITM_N	VSS	FSB2BNR_N	FSB2DBSY_N	VSS	FSB2DRDY_N	FSB2LOCK_N	AE
AD	VSS	VSS	VSS	FSB2BREQ1_N	FSB2BREQ0_N	VSS	FSB2BPRI_N	FSB2DEFER_N	VTT	FSB2RS_N[2]	FSB2RS_N[0]	VSS	FSB2TRDY_N	FSB2ADS_N	VSS	AD
AC	VSS	VTT	FSB2A_N[13]	FSB2A_N[16]	VSS	FSB2A_N[11]	FSB2HIT_N	VSS	FSB2A_N[32]	FSB2RS_N[1]	VSS	FSB2A_N[38]	FSB2A_N[33]	VTT	FSB2A_N[31]	AC
AB	VSS	FSB2A_N[14]	FSB2A_N[12]	VSS	FSB2A_N[36]	FSB2VREF[0]	VTT	FSB2A_N[9]	FSB2A_N[27]	VSS	FSB2A_N[35]	FSB2A_N[34]	VSS	FSB2A_N[21]	FSB2A_N[20]	AB
AA	VSS	FSB2A_N[10]	VSS	FSB2A_N[15]	FSB2A_N[8]	VSS	FSB2A_N[5]	FSB2A_N[6]	VSS	FSB2ADSTB_N[1]	FSB2A_N[26]	VTT	FSB2A_N[22]	FSB2A_N[30]	VSS	AA
Y	VSS	VSS	FSB2A_N[3]	FSB2ADSTB_N[0]	VTT	FSB2A_N[7]	FSB2REQ_N[0]	VSS	FSB2A_N[39]	FSB2A_N[19]	VSS	FSB2A_N[25]	FSB2A_N[23]	VSS	FSB2A_N[29]	Y
W	VTT	FSB2A_N[37]	FSB2REQ_N[4]	VSS	FSB2REQ_N[1]	FSB2A_N[4]	VSS	FSB2REQ_N[3]	FSB2A_N[17]	VTT	FSB2A_N[28]	FSB2A_N[24]	VSS	XDPSLWCRES	XDPODTCRES	W
V	VSS	VSS	VSS	SPD0SMBDATA	FSB2REQ_N[2]	VSS	XDPD_N[3]	XDPD_N[5]	VSS	XDPD_N[4]	FSB2A_N[18]	VSS	XDPD_N[2]	XDPD_N[1]	VSS	V
U	VSS	VSS	VPPSMBCLK	SPD0SMBCLK	VSS	XDPD_N[7]	XDPD_N[8]	VSS	XDPD_N[10]	XDPD_N[9]	VSS	XDPDSTBP_N	XDPD_N[6]	VSS	XDPCOMCRES	U
T	VSS	VSS	VPPSMBDATA	VSS	SPD1SMBDATA	XDPD_N[13]	VSS	XDPRDY_N	XDPD_N[11]	VSS	XDPD_N[0]	XDPDSTBN_N	VSS	XDPD_N[15]	XDPD_N[14]	T
R	VCCPE	VSS	VSS	TESTHI_V3REF	SPD1SMBCLK	VSS	RESET_N	TESTHI_V3REF	VSS	PWRGOOD	XDPD_N[12]	VSS	TDO	TRST_N	VSS	R
P	VCCPE	VSS	PE3RN[3]	TESTHI_V3REF	VSS	SPD2SMBDATA	VSS	VSS	CFGSMBDATA	CFGSMBCLK	VSS	ERR_N[2]	TDI	VSS	V3REF	P
N	VSS	PE1RN[3]	PE3RP[3]	VCCPE	PE0TN[3]	SPD2SMBCLK	VSS	SPD3SMBCLK	VSS	VSS	ERR_N[1]	ERR_N[0]	VSS	TCK	TMS	N
M	PE7TN[3]	PE1RP[3]	VSS	PE3TN[3]	PE0TP[3]	VSS	VSS	SPD3SMBDATA	VCCPE	VSS	INT_N[6]	VSS	INT_N[4]	INT_N[5]	VSS	M
L	PE7TP[3]	VCCPE	PE1TN[3]	PE3TP[3]	VSS	PE0RN[3]	PE0RP[3]	VSS	PE0TN[2]	PE0TP[2]	VSS	INT_N[3]	INT_N[1]	VSS	INT_N[2]	L
K	VSS	PE1RN[2]	PE1TP[3]	VSS	PE3RN[2]	PE3RP[2]	VCCPE	PE0RN[2]	PE0RP[2]	VSS	FSBSLWCTRL	INT_N[0]	VSS	RSVD	RSVD	K
J	PE7RN[2]	PE1RP[2]	VSS	PE2TP[0]	PE2TN[0]	VSS	PE3TN[2]	PE3TP[2]	VSS	PE0TN[1]	PE0TP[1]	VCCPE	PSEL[1]	PSEL[2]	VSS	J
H	PE7RP[2]	VSS	PE1TP[2]	PE1TN[2]	VCCPE	PE2TP[1]	PE2TN[1]	VSS	PE3RN[1]	PE3RP[1]	VSS	PE0RN[1]	PE0RP[1]	VSS	PSEL[0]	H
G	VSS	PE7TP[2]	PE7TN[2]	VSS	PE2RP[0]	PE2RN[0]	VSS	PE2TP[2]	PE2TN[2]	VCCPE	PE0TN[0]	PE0TP[0]	VSS	PE0RP[0]	PE0RN[0]	G
F	PE6TP[1]	PE6TN[1]	VCCPE	PE1RN[1]	PE1RP[1]	VSS	PE2RP[1]	PE2RN[1]	VSS	PE3TN[1]	PE3TP[1]	VSS	PE3RN[0]	PE3RP[0]	VCCPE	F
E	PE6RN[1]	VSS	PE7RP[1]	PE7RN[1]	VSS	PE1TP[0]	PE1TN[0]	VCCPE	PE2RP[2]	PE2RN[2]	VSS	PE3TN[0]	PE3TP[0]	VSS	VCCPE	E
D	VCCPE	PE6TP[2]	PE6TN[2]	VSS	PECLKP	PECLKN	VSS	PE1TP[1]	PE1TN[1]	VSS	PE2RP[3]	PE2RN[3]	VCCPE	VCCPE	VSS	D
C	PE6RP[2]	PE6RN[2]	VSS	PE7TP[0]	PE7TN[0]	VCCPE	PE1RP[0]	PE1RN[0]	VSS	PE2TP[3]	PE2TN[3]	VSS	VCCPE	VSS	VSS	C
B	PE5RN[1]	VSS	PE6RP[3]	PE6RN[3]	VSS	PE7TP[1]	PE7TN[1]	VSS	PEVSSBG	PEVCCBG	VCCPE	VCCPE	VSS	VSS		B
A	VSS	PE6TP[3]	PE6TN[3]	VCCPE	PE7RP[0]	PE7RN[0]	VSS	PEVSSA	PEVCCA	VSS	VCCPE	VSS	VSS			A
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 1 of 23)

Ball No.	Signal Name	Buffer Type	Direction
A3	VSS	Power	
A4	VSS	Power	
A5	VCCPE	Power	
A6	VSS	Power	
A7	PEVCCA	Power	
A8	PEVSSA	Power	
A9	VSS	Power	
A10	PE7RN[0]		
A11	PE7RP[0]		
A12	VCCPE	Power	
A13	PE6TN[3]		
A14	PE6TP[3]		
A15	VSS	Power	
A16	PE5TN[1]		
A17	PE5TP[1]		
A18	VSS	Power	
A19	PE4RN[3]		
A20	PE4RP[3]		
A21	VSS	Power	
A22	FBD23RESIN		
A23	RSVD		
A24	VCCFBD	Power	
A25	FBD3SBOP[1]		
A26	FBD3SBON[1]		
A27	VSS	Power	
A28	FBD23VCCA	Power	
A29	FBD23VSSA	Power	
A30	VSS	Power	
A31	FBD23CLKP		
A32	FBD23CLKN		
A33	VSS	Power	
A34	FBD2SBOP[0]		
A35	FBD2SBON[0]		
A36	VSS	Power	
A37	FBD01ICOMPBIAS		
A38	FBD01BGBIASEXT		
A39	VSS	Power	
A40	FBD01RESIN		
A41	VCCFBD	Power	
A42	VSS	Power	
A43	VSS	Power	
B2	VSS	Power	
B3	VSS	Power	
B4	VCCPE	Power	
B5	VCCPE	Power	
B6	PEVCCBG		
B7	PEVSSBG		
B8	VSS	Power	
B9	PE7TN[1]		
B10	PE7TP[1]		
B11	VSS	Power	
B12	PE6RN[3]		
B13	PE6RP[3]		
B14	VSS	Power	
B15	PE5RN[1]		
B16	PE5RP[1]		
B17	VCCPE	Power	
B18	PE5TN[0]		
B19	PE5TP[0]		
B20	VSS	Power	
B21	FBD23ICOMPBIAS		
B22	FBD23BGBIASEXT		
B23	VSS	Power	
B24	FBD3SBOP[0]		
B25	FBD3SBON[0]		
B26	VSS	Power	
B27	FBD3SBOP[3]		
B28	FBD3SBON[3]		
B29	VCCFBD	Power	
B30	FBD3SBOP[5]		
B31	FBD3SBON[5]		
B32	VSS	Power	
B33	VSS	Power	
B34	VSS	Power	
B35	VSS	Power	
B36	FBD2SBOP[2]		
B37	FBD2SBON[2]		
B38	VSS	Power	
B39	FBD1NBIP[6]		
B40	FBD1NBIN[6]		
B41	VCCFBD	Power	
B42	VCCFBD	Power	
B43	VSS	Power	
B44	VSS	Power	
C1	VSS	Power	
C2	VSS	Power	
C3	VCCPE	Power	
C4	VSS	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 2 of 23)

Ball No.	Signal Name	Buffer Type	Direction
C5	PE2TN[3]		
C6	PE2TP[3]		
C7	VSS	Power	
C8	PE1RN[0]		
C9	PE1RP[0]		
C10	VCCPE	Power	
C11	PE7TN[0]		
C12	PE7TP[0]		
C13	VSS	Power	
C14	PE6RN[2]		
C15	PE6RP[2]		
C16	VSS	Power	
C17	PE5RN[0]		
C18	PE5RP[0]		
C19	VSS	Power	
C20	PE4TN[3]		
C21	VSS	Power	
C22	VCCFBD		
C23	VSS	Power	
C24	VSS	Power	
C25	VSS	Power	
C26	FBD3SBOP[4]		
C27	FBD3SBON[4]		
C28	VSS	Power	
C29	FBD3SBOP[9]		
C30	FBD3SBON[9]		
C31	VSS	Power	
C32	VSS	Power	
C33	VSS	Power	
C34	VCCFBD	Power	
C35	FBD2SBOP[3]		
C36	FBD2SBON[3]		
C37	VSS	Power	
C38	FBD2SBOP[4]		
C39	FBD2SBON[4]		
C40	VSS	Power	
C41	FBD1NBIP[8]		
C42	FBD1NBIN[8]		
C43	VCCFBD	Power	
C44	VSS	Power	
C45	VSS	Power	
D1	VSS	Power	
D2	VCCPE	Power	
D3	VCCPE	Power	
D4	PE2RN[3]		
D5	PE2RP[3]		
D6	VSS	Power	
D7	PE1TN[1]		
D8	PE1TP[1]		
D9	VSS	Power	
D10	PECLKN		
D11	PECLKP		
D12	VSS	Power	
D13	PE6TN[2]		
D14	PE6TP[2]		
D15	VCCPE	Power	
D16	PE5TN[2]		
D17	PE5TP[2]		
D18	VSS	Power	
D19	PE4RN[2]		
D20	PE4TP[3]		
D21	VSS	Power	
D22	VSS	Power	
D23	VSS	Power	
D24	VSS	Power	
D25	FBD3SBOP[2]		
D26	FBD3SBON[2]		
D27	VCCFBD	Power	
D28	FBD3SBOP[6]		
D29	FBD3SBON[6]		
D30	VSS	Power	
D31	FBD2NBIP[9]		
D32	FBD2NBIN[9]		
D33	VSS	Power	
D34	FBD2SBOP[1]		
D35	FBD2SBON[1]		
D36	VSS	Power	
D37	FBD2SBOP[9]		
D38	FBD2SBON[9]		
D39	VCCFBD	Power	
D40	FBD1NBIP[7]		
D41	FBD1NBIN[7]		
D42	VSS	Power	
D43	RSVD		
D44	VCCFBD	Power	
D45	VSS	Power	
E1	VCCPE	Power	
E2	VSS	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 3 of 23)

Ball No.	Signal Name	Buffer Type	Direction
E3	PE3TP[0]		
E4	PE3TN[0]		
E5	VSS	Power	
E6	PE2RN[2]		
E7	PE2RP[2]		
E8	VCCPE	Power	
E9	PE1TN[0]		
E10	PE1TP[0]		
E11	VSS	Power	
E12	PE7RN[1]		
E13	PE7RP[1]		
E14	VSS	Power	
E15	PE6RN[1]		
E16	PE6RP[1]		
E17	VSS	Power	
E18	PE4TN[2]		
E19	PE4RP[2]		
E20	VCCPE	Power	
E21	FBD3NBIP[8]		
E22	FBD3NBIN[8]		
E23	VSS	Power	
E24	FBD3NBIP[11]		
E25	FBD3NBIN[11]		
E26	VSS	Power	
E27	FBD3SBOP[7]		
E28	FBD3SBON[7]		
E29	VSS	Power	
E30	FBD2NBIP[8]		
E31	FBD2NBIN[8]		
E32	VCCFBD	Power	
E33	FBD2NBIP[11]		
E34	FBD2NBIN[11]		
E35	VSS	Power	
E36	FBD2SBOP[5]		
E37	FBD2SBON[5]		
E38	VSS	Power	
E39	FBD1NBIP[12]		
E40	FBD1NBIN[12]		
E41	VSS	Power	
E42	FBD1NBIP[10]		
E43	FBD1NBIN[10]		
E44	VCCFBD	Power	
E45	VCCFBD	Power	
F1	VCCPE	Power	
F2	PE3RP[0]		
F3	PE3RN[0]		
F4	VSS	Power	
F5	PE3TP[1]		
F6	PE3TN[1]		
F7	VSS	Power	
F8	PE2RN[1]		
F9	PE2RP[1]		
F10	VSS	Power	
F11	PE1RP[1]		
F12	PE1RN[1]		
F13	VCCPE	Power	
F14	PE6TN[1]		
F15	PE6TP[1]		
F16	VSS	Power	
F17	PE5RP[2]		
F18	PE4TP[2]		
F19	VSS	Power	
F20	PE4RN[1]		
F21	VSS	Power	
F22	VSS	Power	
F23	FBD3NBIP[9]		
F24	FBD3NBIN[9]		
F25	VCCFBD	Power	
F26	FBD3SBOP[8]		
F27	FBD3SBON[8]		
F28	VSS	Power	
F29	FBD2NBIP[7]		
F30	FBD2NBIN[7]		
F31	VSS	Power	
F32	FBD2NBIP[10]		
F33	FBD2NBIN[10]		
F34	VSS	Power	
F35	FBD2SBOP[6]		
F36	FBD2SBON[6]		
F37	VCCFBD	Power	
F38	FBD1NBIP[13]		
F39	FBD1NBIN[13]		
F40	VSS	Power	
F41	FBD1NBIP[11]		
F42	FBD1NBIN[11]		
F43	VSS	Power	
F44	FBD01VSSA	Power	
F45	FBD01VCCA	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 4 of 23)

Ball No.	Signal Name	Buffer Type	Direction
G1	PE0RN[0]		
G2	PE0RP[0]		
G3	VSS	Power	
G4	PE0TP[0]		
G5	PE0TN[0]		
G6	VCCPE	Power	
G7	PE2TN[2]		
G8	PE2TP[2]		
G9	VSS	Power	
G10	PE2RN[0]		
G11	PE2RP[0]		
G12	VSS	Power	
G13	PE7TN[2]		
G14	PE7TP[2]		
G15	VSS	Power	
G16	PE6RN[0]		
G17	PE5RN[2]		
G18	VCCPE	Power	
G19	PE4TN[1]		
G20	PE4RP[1]		
G21	VSS	Power	
G22	FBD3NBIP[7]		
G23	FBD3NBIN[7]		
G24	VSS	Power	
G25	FBD3NBIP[10]		
G26	FBD3NBIN[10]		
G27	VSS	Power	
G28	FBD2NBIP[6]		
G29	FBD2NBIN[6]		
G30	VCCFBD	Power	
G31	FBD2NBIN[12]		
G32	VSS	Power	
G33	VSS	Power	
G34	FBD2SBOP[7]		
G35	FBD2SBON[7]		
G36	VSS	Power	
G37	FBD1NBIP[5]		
G38	FBD1NBIN[5]		
G39	VSS	Power	
G40	FBD1NBIP[9]		
G41	FBD1NBIN[9]		
G42	VCCFBD	Power	
G43	FBD1SBOP[0]		
G44	FBD1SBON[0]		
G45	VSS	Power	
H1	PSEL[0]		
H2	VSS	Power	
H3	PE0RP[1]		
H4	PE0RN[1]		
H5	VSS	Power	
H6	PE3RP[1]		
H7	PE3RN[1]		
H8	VSS	Power	
H9	PE2TN[1]		
H10	PE2TP[1]		
H11	VCCPE	Power	
H12	PE1TN[2]		
H13	PE1TP[2]		
H14	VSS	Power	
H15	PE7RP[2]		
H16	PE6RP[0]		
H17	VSS	Power	
H18	PE5TP[3]		
H19	PE4TP[1]		
H20	VSS	Power	
H21	FBD3NBIP[6]		
H22	FBD3NBIN[6]		
H23	VCCFBD	Power	
H24	FBD3NBIN[12]		
H25	VSS	Power	
H26	VSS	Power	
H27	VSS	Power	
H28	VSS	Power	
H29	VSS	Power	
H30	FBD2NBIN[13]		
H31	FBD2NBIP[12]		
H32	VSS	Power	
H33	FBD2SBOP[8]		
H34	FBD2SBON[8]		
H35	VCCFBD	Power	
H36	FBD1NBIP[4]		
H37	FBD1NBIN[4]		
H38	VSS	Power	
H39	VSS	Power	
H40	VSS	Power	
H41	VSS	Power	
H42	FBD1SBOP[1]		
H43	FBD1SBON[1]		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 5 of 23)

Ball No.	Signal Name	Buffer Type	Direction
H44	VSS	Power	
H45	FBD01CLKP		
J1	VSS	Power	
J2	PSEL[2]		
J3	PSEL[1]		
J4	VCCPE	Power	
J5	PE0TP[1]		
J6	PE0TN[1]		
J7	VSS	Power	
J8	PE3TP[2]		
J9	PE3TN[2]		
J10	VSS	Power	
J11	PE2TN[0]		
J12	PE2TP[0]		
J13	VSS	Power	
J14	PE1RP[2]		
J15	PE7RN[2]		
J16	VCCPE	Power	
J17	PE5RP[3]		
J18	PE5TN[3]		
J19	VSS	Power	
J20	PE4RN[0]		
J21	VSS	Power	
J22	VSS	Power	
J23	FBD3NBIN[13]		
J24	FBD3NBIP[12]		
J25	VSS	Power	
J26	VSS	Power	
J27	VSS	Power	
J28	VCCFBD	Power	
J29	FBD2NBIN[5]		
J30	FBD2NBIP[13]		
J31	VSS	Power	
J32	VSS	Power	
J33	VSS	Power	
J34	VSS	Power	
J35	FBD1NBIP[3]		
J36	FBD1NBIN[3]		
J37	VSS	Power	
J38	VSS	Power	
J39	VSS	Power	
J40	VCCFBD	Power	
J41	FBD1SBOP[2]		
J42	FBD1SBON[2]		
J43	VSS	Power	
J44	FBD1SBOP[3]		
J45	FBD01CLKN		
K1	RSVD		
K2	RSVD		
K3	VSS	Power	
K4	INT_N[0]		
K5	FSBSLWCTRL		
K6	VSS	Power	
K7	PE0RP[2]		
K8	PE0RN[2]		
K9	VCCPE	Power	
K10	PE3RP[2]		
K11	PE3RN[2]		
K12	VSS	Power	
K13	PE1TP[3]		
K14	PE1RN[2]		
K15	VSS	Power	
K16	PE6TN[0]		
K17	PE5RN[3]		
K18	VSS	Power	
K19	PE4TN[0]		
K20	PE4RP[0]		
K21	VCCFBD	Power	
K22	FBD3NBIN[5]		
K23	FBD3NBIP[13]		
K24	VSS	Power	
K25	VSS	Power	
K26	VSS	Power	
K27	VSS	Power	
K28	FBD2NBIN[4]		
K29	FBD2NBIP[5]		
K30	VSS	Power	
K31	VSS	Power	
K32	VSS	Power	
K33	VCCFBD	Power	
K34	FBD1NBIP[2]		
K35	FBD1NBIN[2]		
K36	VSS	Power	
K37	VSS	Power	
K38	VSS	Power	
K39	VSS	Power	
K40	FBD1SBON[7]		
K41	FBD1SBOP[7]		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 6 of 23)

Ball No.	Signal Name	Buffer Type	Direction
K42	VSS	Power	
K43	FBD1SBOP[4]		
K44	FBD1SBON[3]		
K45	VCCFBD	Power	
L1	INT_N[2]		
L2	VSS	Power	
L3	INT_N[1]		
L4	INT_N[3]		
L5	VSS	Power	
L6	PE0TP[2]		
L7	PE0TN[2]		
L8	VSS	Power	
L9	PE0RP[3]		
L10	PE0RN[3]		
L11	VSS	Power	
L12	PE3TP[3]		
L13	PE1TN[3]		
L14	VCCPE	Power	
L15	PE7TP[3]		
L16	PE6TP[0]		
L17	VSS	Power	
L18	VCCPE	Power	
L19	PE4TP[0]		
L20	VSS	Power	
L21	FBD3NBIN[4]		
L22	FBD3NBIP[5]		
L23	VSS	Power	
L24	FBD3NBIN[1]		
L25	FBD3NBIP[1]		
L26	VCCFBD	Power	
L27	FBD2NBIN[3]		
L28	FBD2NBIP[4]		
L29	VSS	Power	
L30	FBD2NBIN[2]		
L31	FBD2NBIP[2]		
L32	VSS	Power	
L33	FBD1NBIP[1]		
L34	FBD1NBIN[1]		
L35	VSS	Power	
L36	VSS	Power	
L37	VSS	Power	
L38	VCCFBD	Power	
L39	FBD1SBON[8]		
L40	FBD1SBOP[8]		

Ball No.	Signal Name	Buffer Type	Direction
L41	VSS	Power	
L42	FBD1SBOP[5]		
L43	FBD1SBON[4]		
L44	VSS	Power	
L45	FBD1SBOP[9]		
M1	VSS	Power	
M2	INT_N[5]		
M3	INT_N[4]		
M4	VSS	Power	
M5	INT_N[6]		
M6	VSS	Power	
M7	VCCPE	Power	
M8	SPD3SMBDATA		
M9	VSS	Power	
M10	VSS	Power	
M11	PE0TP[3]		
M12	PE3TN[3]		
M13	VSS	Power	
M14	PE1RP[3]		
M15	PE7TN[3]		
M16	VSS	Power	
M17	PE7RP[3]		
M18	VCCPE	Power	
M19	VCCPE	Power	
M20	VSS	Power	
M21	FBD3NBIP[4]		
M22	VSS	Power	
M23	FBD3NBIN[3]		
M24	FBD3NBIP[3]		
M25	VSS	Power	
M26	VSS	Power	
M27	FBD2NBIP[3]		
M28	VSS	Power	
M29	FBD2NBIN[0]		
M30	FBD2NBIP[0]		
M31	VCCFBD	Power	
M32	FBD1NBIP[0]		
M33	FBD1NBIN[0]		
M34	VSS	Power	
M35	FBD0NBIP[12]		
M36	FBD0NBIN[12]		
M37	VSS	Power	
M38	FBD0NBIP[8]		
M39	FBD0NBIN[8]		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 7 of 23)

Ball No.	Signal Name	Buffer Type	Direction
M40	VSS	Power	
M41	FBD1SBOP[6]		
M42	FBD1SBON[5]		
M43	VCCFBD	Power	
M44	FBD0SBOP[0]		
M45	FBD1SBON[9]		
N1	TMS		
N2	TCK		
N3	VSS	Power	
N4	ERR_N[0]		
N5	ERR_N[1]		
N6	VSS	Power	
N7	VSS	Power	
N8	SPD3SMBCLK		
N9	VSS	Power	
N10	SPD2SMBCLK		
N11	PE0TN[3]		
N12	VCCPE	Power	
N13	PE3RP[3]		
N14	PE1RN[3]		
N15	VSS		
N16	VCCPE	Power	
N17	PE7RN[3]		
N18	VSS	Power	
N19	PERCOMPO		
N20	PEICOMPI		
N21	VSS	Power	
N22	FBD3NBIN[2]		
N23	FBD3NBIP[2]		
N24	VCCFBD	Power	
N25	FBD3NBIN[0]		
N26	FBD3NBIP[0]		
N27	VSS	Power	
N28	FBD2NBIN[1]		
N29	FBD2NBIP[1]		
N30	VSS	Power	
N31	FBD0NBIP[3]		
N32	FBD0NBIN[3]		
N33	VSS	Power	
N34	FBD0NBIP[13]		
N35	FBD0NBIN[13]		
N36	VCCFBD	Power	
N37	FBD0NBIP[7]		
N38	FBD0NBIN[7]		
N39	VSS	Power	
N40	FBD0NBIP[10]		
N41	FBD1SBON[6]		
N42	VSS	Power	
N43	FBD0SBOP[1]		
N44	FBD0SBON[0]		
N45	VSS	Power	
P1	V3REF	Power	
P2	VSS	Power	
P3	TDI		
P4	ERR_N[2]		
P5	VSS	Power	
P6	CFGSMBCLK		
P7	CFGSMBDATA		
P8	VSS	Power	
P9	VSS	Power	
P10	SPD2SMBDATA		
P11	VSS	Power	
P12	TESTHI_V3REF		
P13	PE3RN[3]		
P14	VSS	Power	
P15	VCCPE	Power	
P16	VCCPE	Power	
P17	VCCPE	Power	
P18	VCCPE	Power	
P19	VCCPE	Power	
P20	VSS	Power	
P21	VSS	Power	
P22	VSS	Power	
P23	VSS	Power	
P24	VCCFBD	Power	
P25	VSS	Power	
P26	VSS	Power	
P27	VSS	Power	
P28	VCCFBD	Power	
P29	VCCFBD	Power	
P30	FBD0NBIP[2]		
P31	FBD0NBIN[2]		
P32	VSS	Power	
P33	FBD0NBIP[5]		
P34	FBD0NBIN[5]		
P35	VSS	Power	
P36	FBD0NBIP[9]		
P37	FBD0NBIN[9]		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 8 of 23)

Ball No.	Signal Name	Buffer Type	Direction
P38	VSS	Power	
P39	FBD0NBIP[11]		
P40	FBD0NBIN[10]		
P41	VCCFBD	Power	
P42	FBD0SBOP[5]		
P43	FBD0SBON[1]		
P44	VSS	Power	
P45	FBD0SBOP[2]		
R1	VSS	Power	
R2	TRST_N		
R3	TDO		
R4	VSS	Power	
R5	XDPD_N[12]		
R6	PWRGOOD		
R7	VSS	Power	
R8	TESTHI_V3REF		
R9	RESET_N		
R10	VSS	Power	
R11	SPD1SMBCLK		
R12	TESTHI_V3REF		
R13	VSS	Power	
R14	VSS	Power	
R15	VCCPE	Power	
R16	VSS	Power	
R17	VSS	Power	
R18	VCCPE	Power	
R19	VSS	Power	
R20	VSS	Power	
R21	VSS	Power	
R22	VCCFBD	Power	
R23	VSS	Power	
R24	VCCFBD	Power	
R25	VSS	Power	
R26	VCCFBD	Power	
R27	VSS	Power	
R28	VSS	Power	
R29	VCCFBD	Power	
R30	VCCFBD	Power	
R31	VSS	Power	
R32	FBD0NBIP[4]		
R33	FBD0NBIN[4]		
R34	VCCFBD	Power	
R35	FBD0NBIP[6]		
R36	FBD0NBIN[6]		

Ball No.	Signal Name	Buffer Type	Direction
R37	VSS	Power	
R38	VSS	Power	
R39	FBD0NBIN[11]		
R40	VSS	Power	
R41	FBD0SBOP[6]		
R42	FBD0SBON[5]		
R43	VSS	Power	
R44	FBD0SBOP[3]		
R45	FBD0SBON[2]		
T1	XDPD_N[14]		
T2	XDPD_N[15]		
T3	VSS	Power	
T4	XDPDSTBN_N		
T5	XDPD_N[0]		
T6	VSS	Power	
T7	XDPD_N[11]		
T8	XDPRDY_N		
T9	VSS	Power	
T10	XDPD_N[13]		
T11	SPD1SMBDATA		
T12	VSS	Power	
T13	VPPSMBDATA		
T14	VSS	Power	
T15	VSS	Power	
T16	VSS	Power	
T17	VSS	Power	
T18	VSS	Power	
T19	VCCPE	Power	
T20	VSS	Power	
T21	VCC	Power	
T22	VSS	Power	
T23	VCC	Power	
T24	VSS	Power	
T25	VCC	Power	
T26	VSS	Power	
T27	VCC	Power	
T28	VSS	Power	
T29	VSS	Power	
T30	VSS	Power	
T31	VSS	Power	
T32	VCCFBD	Power	
T33	VSS	Power	
T34	VSS	Power	
T35	VSS	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 9 of 23)

Ball No.	Signal Name	Buffer Type	Direction
T36	VSS	Power	
T37	VSS	Power	
T38	VSS	Power	
T39	VCCFBD	Power	
T40	FBD0SBOP[7]		
T41	FBD0SBON[6]		
T42	VSS	Power	
T43	FBD0SBOP[4]		
T44	FBD0SBON[3]		
T45	VSS	Power	
U1	XDPCOMCRES		
U2	VSS	Power	
U3	XDPD_N[6]		
U4	XDPDSTBP_N		
U5	VSS	Power	
U6	XDPD_N[9]		
U7	XDPD_N[10]		
U8	VSS	Power	
U9	XDPD_N[8]		
U10	XDPD_N[7]		
U11	VSS	Power	
U12	SPD0SMBCLK		
U13	VPPSMBCLK		
U14	VSS	Power	
U15	VSS	Power	
U16	VCC	Power	
U17	VSS	Power	
U18	VCCSPL	Power	
U19	VSS	Power	
U20	VCC	Power	
U21	VSS	Power	
U22	VCC	Power	
U23	VSS	Power	
U24	VCC	Power	
U25	VSS	Power	
U26	VCC	Power	
U27	VSS	Power	
U28	VCC	Power	
U29	VSS	Power	
U30	VSS	Power	
U31	VSS	Power	
U32	VCCFBD	Power	
U33	FBD0NBIP[1]		
U34	FBD0NBIN[1]		

Ball No.	Signal Name	Buffer Type	Direction
U35	VSS	Power	
U36	VSS	Power	
U37	VSS	Power	
U38	VSS	Power	
U39	FBD0SBOP[8]		
U40	FBD0SBON[7]		
U41	VSS	Power	
U42	FSB0A_N[32]		
U43	FBD0SBON[4]		
U44	VCCFBD	Power	
U45	FBD0SBOP[9]		
V1	VSS	Power	
V2	XDPD_N[1]		
V3	XDPD_N[2]		
V4	VSS	Power	
V5	FSB2A_N[18]		
V6	XDPD_N[4]		
V7	VSS	Power	
V8	XDPD_N[5]		
V9	XDPD_N[3]		
V10	VSS	Power	
V11	FSB2REQ_N[2]		
V12	SPD0SMBDATA		
V13	VSS	Power	
V14	VSS	Power	
V15	VSS	Power	
V16	VSS	Power	
V17	VCC	Power	
V18	VSS	Power	
V19	VCC	Power	
V20	VSS	Power	
V21	VCC	Power	
V22	VSS	Power	
V23	VCC	Power	
V24	VSS	Power	
V25	VCC	Power	
V26	VSS	Power	
V27	VCC	Power	
V28	VSS	Power	
V29	VTT	Power	
V30	VTT	Power	
V31	VSS	Power	
V32	FBD0NBIP[0]		
V33	FBD0NBIN[0]		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 10 of 23)

Ball No.	Signal Name	Buffer Type	Direction
V34	VSS	Power	
V35	FSB0A_N[10]		
V36	VSS	Power	
V37	VCCFBD	Power	
V38	FSB0A_N[13]		
V39	FBD0SBON[8]		
V40	VSS	Power	
V41	FSB0A_N[38]		
V42	FSB0A_N[33]		
V43	VSS	Power	
V44	FSB0A_N[31]		
V45	FBD0SBON[9]		
W1	XDPODTCRES		
W2	XDPSLWCRES		
W3	VSS	Power	
W4	FSB2A_N[24]		
W5	FSB2A_N[28]		
W6	VTT	Power	
W7	FSB2A_N[17]		
W8	FSB2REQ_N[3]		
W9	VSS	Power	
W10	FSB2A_N[4]		
W11	FSB2REQ_N[1]		
W12	VSS	Power	
W13	FSB2REQ_N[4]		
W14	FSB2A_N[37]		
W15	VTT	Power	
W16	VTT	Power	
W17	VTT	Power	
W18	VCC	Power	
W19	VSS	Power	
W20	VCC	Power	
W21	VSS	Power	
W22	VCC	Power	
W23	VSS	Power	
W24	VCC	Power	
W25	VSS	Power	
W26	VCC	Power	
W27	VSS	Power	
W28	VCC	Power	
W29	VTT	Power	
W30	VTT	Power	
W31	VSS	Power	
W32	FSB0A_N[14]		
W33	VSS	Power	
W34	FSB0A_N[11]		
W35	FSB0A_N[12]		
W36	VSS	Power	
W37	FSB0A_N[15]		
W38	FSB0A_N[16]		
W39	VSS	Power	
W40	FSB0A_N[27]		
W41	FSB0A_N[34]		
W42	VTT	Power	
W43	FSB0A_N[35]		
W44	FSB0A_N[26]		
W45	VSS	Power	
Y1	FSB2A_N[29]		
Y2	VSS	Power	
Y3	FSB2A_N[23]		
Y4	FSB2A_N[25]		
Y5	VSS	Power	
Y6	FSB2A_N[19]		
Y7	FSB2A_N[39]		
Y8	VSS	Power	
Y9	FSB2REQ_N[0]		
Y10	FSB2A_N[7]		
Y11	VTT	Power	
Y12	FSB2ADSTB_N[0]		
Y13	FSB2A_N[3]		
Y14	VSS	Power	
Y15	VSS	Power	
Y16	VTT	Power	
Y17	VTT	Power	
Y18	VSS	Power	
Y19	VCC	Power	
Y20	VSS	Power	
Y21	VCC	Power	
Y22	VSS	Power	
Y23	VCC	Power	
Y24	VSS	Power	
Y25	VCC	Power	
Y26	VSS	Power	
Y27	VCC	Power	
Y28	VSS	Power	
Y29	VTT	Power	
Y30	VTT	Power	
Y31	VSS	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 11 of 23)

Ball No.	Signal Name	Buffer Type	Direction
Y32	VSS	Power	
Y33	FSB0A_N[4]		
Y34	FSB0A_N[7]		
Y35	VTT	Power	
Y36	FSB0A_N[8]		
Y37	FSB0A_N[36]		
Y38	VSS	Power	
Y39	FSB0ADSTB_N[1]		
Y40	FSB0A_N[28]		
Y41	VSS	Power	
Y42	FSB0A_N[22]		
Y43	FSB0A_N[21]		
Y44	VSS	Power	
Y45	FSB0A_N[20]		
AA1	VSS	Power	
AA2	FSB2A_N[30]		
AA3	FSB2A_N[22]		
AA4	VTT	Power	
AA5	FSB2A_N[26]		
AA6	FSB2ADSTB_N[1]		
AA7	VSS	Power	
AA8	FSB2A_N[6]		
AA9	FSB2A_N[5]		
AA10	VSS	Power	
AA11	FSB2A_N[8]		
AA12	FSB2A_N[15]		
AA13	VSS	Power	
AA14	FSB2A_N[10]		
AA15	VSS	Power	
AA16	VTT	Power	
AA17	VTT	Power	
AA18	VCC	Power	
AA19	VSS	Power	
AA20	VCC	Power	
AA21	VSS	Power	
AA22	VCC	Power	
AA23	VSS	Power	
AA24	VCC	Power	
AA25	VSS	Power	
AA26	VCC	Power	
AA27	VSS	Power	
AA28	VCC	Power	
AA29	VTT	Power	
AA30	VTT	Power	
AA31	VSS	Power	
AA32	FSB0REQ_N[4]		
AA33	FSB0REQ_N[1]		
AA34	VSS	Power	
AA35	FSB0ADSTB_N[0]		
AA36	FSB0A_N[9]		
AA37	VSS	Power	
AA38	FSB0A_N[39]		
AA39	FSB0A_N[19]		
AA40	VTT	Power	
AA41	FSB0A_N[25]		
AA42	FSB0A_N[29]		
AA43	VSS	Power	
AA44	FSB0A_N[23]		
AA45	FSB0A_N[30]		
AB1	FSB2A_N[20]		
AB2	FSB2A_N[21]		
AB3	VSS	Power	
AB4	FSB2A_N[34]		
AB5	FSB2A_N[35]		
AB6	VSS	Power	
AB7	FSB2A_N[27]		
AB8	FSB2A_N[9]		
AB9	VTT	Power	
AB10	FSB2VREF[0]		
AB11	FSB2A_N[36]		
AB12	VSS	Power	
AB13	FSB2A_N[12]		
AB14	FSB2A_N[14]		
AB15	VSS	Power	
AB16	VTT	Power	
AB17	VTT	Power	
AB18	VSS	Power	
AB19	VCC	Power	
AB20	VSS	Power	
AB21	VCC	Power	
AB22	VSS	Power	
AB23	VCC	Power	
AB24	VSS	Power	
AB25	VCC	Power	
AB26	VSS	Power	
AB27	VCC	Power	
AB28	VSS	Power	
AB29	VTT	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 12 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AB30	VTT	Power	
AB31	VSS	Power	
AB32	FSB0A_N[37]		
AB33	VTT	Power	
AB34	FSB0REQ_N[0]		
AB35	FSB0A_N[6]		
AB36	VSS	Power	
AB37	FSB0A_N[5]		
AB38	FSB0A_N[17]		
AB39	VSS	Power	
AB40	FSB0A_N[18]		
AB41	FSB0A_N[24]		
AB42	VSS	Power	
AB43	FSB0AP_N[1]		
AB44	FSB0AP_N[0]		
AB45	VSS	Power	
AC1	FSB2A_N[31]		
AC2	VTT	Power	
AC3	FSB2A_N[33]		
AC4	FSB2A_N[38]		
AC5	VSS	Power	
AC6	FSB2RS_N[1]		
AC7	FSB2A_N[32]		
AC8	VSS	Power	
AC9	FSB2HIT_N		
AC10	FSB2A_N[11]		
AC11	VSS	Power	
AC12	FSB2A_N[16]		
AC13	FSB2A_N[13]		
AC14	VTT	Power	
AC15	VSS	Power	
AC16	VTT	Power	
AC17	VTT	Power	
AC18	VCC	Power	
AC19	VSS	Power	
AC20	VCC	Power	
AC21	VSS	Power	
AC22	VCC	Power	
AC23	VSS	Power	
AC24	VCC	Power	
AC25	VSS	Power	
AC26	VCC	Power	
AC27	VSS	Power	
AC28	VCC	Power	
AC29	VTT	Power	
AC30	VTT	Power	
AC31	VSS	Power	
AC32	VSS	Power	
AC33	FSB0REQ_N[2]		
AC34	FSB0REQ_N[3]		
AC35	VSS	Power	
AC36	FSB0BREQ0_N		
AC37	FSB0VREF[0]		
AC38	VTT	Power	
AC39	FSB0RSP_N		
AC40	FSB0BPM_N[4]		
AC41	VSS	Power	
AC42	FSB0LOCK_N		
AC43	FSB0ADS_N		
AC44	VTT	Power	
AC45	FSB0DBSY_N		
AD1	VSS	Power	
AD2	FSB2ADS_N		
AD3	FSB2TRDY_N		
AD4	VSS	Power	
AD5	FSB2RS_N[0]		
AD6	FSB2RS_N[2]		
AD7	VTT	Power	
AD8	FSB2DEFER_N		
AD9	FSB2BPRI_N		
AD10	VSS	Power	
AD11	FSB2BREQ0_N		
AD12	FSB2BREQ1_N		
AD13	VSS	Power	
AD14	VSS	Power	
AD15	VSS	Power	
AD16	VTT	Power	
AD17	VTT	Power	
AD18	VSS	Power	
AD19	VCC	Power	
AD20	VSS	Power	
AD21	VCC	Power	
AD22	VSS	Power	
AD23	VCC	Power	
AD24	VSS	Power	
AD25	VCC	Power	
AD26	VSS	Power	
AD27	VCC	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 13 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AD28	VSS	Power	
AD29	VTT	Power	
AD30	VTT	Power	
AD31	VTT	Power	
AD32	VSS	Power	
AD33	FSB0A_N[3]		
AD34	VSS	Power	
AD35	FSB0D_N[59]		
AD36	FSB0BREQ1_N		
AD37	VSS	Power	
AD38	FSB0BPM_N[5]		
AD39	FSB0MCERR_N		
AD40	VSS	Power	
AD41	FSB0BINIT_N		
AD42	FSB0DRDY_N		
AD43	VSS	Power	
AD44	FSB0TRDY_N		
AD45	FSB0RESET_N		
AE1	FSB2LOCK_N		
AE2	FSB2DRDY_N		
AE3	VSS	Power	
AE4	FSB2DBSY_N		
AE5	FSB2BNR_N		
AE6	VSS	Power	
AE7	FSB2HITM_N		
AE8	FSB2DP_N[3]		
AE9	VSS	Power	
AE10	FSB2D_N[8]		
AE11	FSB2D_N[7]		
AE12	VTT	Power	
AE13	FSB2D_N[4]		
AE14	FSB2DBI_N[0]		
AE15	VSS	Power	
AE16	VTT	Power	
AE17	VTT	Power	
AE18	VCC	Power	
AE19	VSS	Power	
AE20	VCC	Power	
AE21	VSS	Power	
AE22	VCC	Power	
AE23	VSS	Power	
AE24	VCC	Power	
AE25	VSS	Power	
AE26	VCC	Power	
AE27	VSS	Power	
AE28	VCC	Power	
AE29	VTT	Power	
AE30	VTT	Power	
AE31	VSS	Power	
AE32	FSB0D_N[60]		
AE33	VSS	Power	
AE34	FSB0D_N[57]		
AE35	FSB0D_N[58]		
AE36	VTT	Power	
AE37	FSB0DP_N[1]		
AE38	FSB0DP_N[3]		
AE39	VSS	Power	
AE40	FSB0BPRI_N		
AE41	FSB0RS_N[2]		
AE42	VTT	Power	
AE43	FSB0BNR_N		
AE44	FSB0RS_N[0]		
AE45	VSS	Power	
AF1	FSB2AP_N[1]		
AF2	VSS	Power	
AF3	FSB2BPM_N[4]		
AF4	FSB2AP_N[0]		
AF5	VTT	Power	
AF6	FSB2BINIT_N		
AF7	FSB2DP_N[0]		
AF8	VSS	Power	
AF9	FSB2D_N[6]		
AF10	FSB2D_N[3]		
AF11	VSS	Power	
AF12	FSB2D_N[0]		
AF13	FSB2D_N[1]		
AF14	VSS	Power	
AF15	VCCSF	Power	
AF16	VTT	Power	
AF17	VTT	Power	
AF18	VSS	Power	
AF19	VCC	Power	
AF20	VSS	Power	
AF21	VCC	Power	
AF22	VSS	Power	
AF23	VCC	Power	
AF24	VSS	Power	
AF25	VCC	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 14 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AF26	VSS	Power	
AF27	VCC	Power	
AF28	VSS	Power	
AF29	VTT	Power	
AF30	VTT	Power	
AF31	VSS	Power	
AF32	VSS	Power	
AF33	FSB0D_N[63]		
AF34	FSB0D_N[61]		
AF35	VSS	Power	
AF36	FSB0D_N[56]		
AF37	FSB0D_N[62]		
AF38	VSS	Power	
AF39	FSB0DP_N[2]		
AF40	FSB0HIT_N		
AF41	VSS	Power	
AF42	FSB0DEFER_N		
AF43	FSB0RS_N[1]		
AF44	VSS	Power	
AF45	FSB0HITM_N		
AG1	VSS	Power	
AG2	FSB2MCERR_N		
AG3	FSB2RSP_N		
AG4	VSS	Power	
AG5	FSB2BPM_N[5]		
AG6	FSB2DP_N[1]		
AG7	VSS	Power	
AG8	FSB2VREF[2]		
AG9	FSB2D_N[9]		
AG10	VTT	Power	
AG11	FSB2D_N[2]		
AG12	FSB2D_N[5]		
AG13	VSS	Power	
AG14	FSB2D_N[13]		
AG15	VCCSF	Power	
AG16	VTT	Power	
AG17	VTT	Power	
AG18	VCC	Power	
AG19	VSS	Power	
AG20	VCC	Power	
AG21	VSS	Power	
AG22	VCC	Power	
AG23	VSS	Power	
AG24	VCC	Power	

Ball No.	Signal Name	Buffer Type	Direction
AG25	VSS	Power	
AG26	VCC	Power	
AG27	VSS	Power	
AG28	VCC	Power	
AG29	VTT	Power	
AG30	VTT	Power	
AG31	VSS	Power	
AG32	FSB0D_N[50]		
AG33	FSB0D_N[49]		
AG34	VTT	Power	
AG35	FSB0DBI_N[3]		
AG36	FSB0D_N[54]		
AG37	VSS	Power	
AG38	FSB0VREF[4]		
AG39	FSB0D_N[46]		
AG40	VTT	Power	
AG41	FSB0D_N[47]		
AG42	FSB0D_N[44]		
AG43	VSS	Power	
AG44	FSB0D_N[45]		
AG45	FSB0D_N[42]		
AH1	FSB2D_N[20]		
AH2	FSB2D_N[22]		
AH3	VTT	Power	
AH4	FSB2D_N[21]		
AH5	FSB2D_N[18]		
AH6	VSS	Power	
AH7	FSB2RESET_N		
AH8	FSB2DP_N[2]		
AH9	VSS	Power	
AH10	FSB2D_N[10]		
AH11	FSB2D_N[11]		
AH12	VSS	Power	
AH13	FSB2D_N[14]		
AH14	VCCSF	Power	
AH15	VTT	Power	
AH16	VTT	Power	
AH17	VTT	Power	
AH18	VSS	Power	
AH19	VCC	Power	
AH20	VSS	Power	
AH21	VCC	Power	
AH22	VSS	Power	
AH23	VCC	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 15 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AH24	VSS	Power	
AH25	VCC	Power	
AH26	VSS	Power	
AH27	VCC	Power	
AH28	VSS	Power	
AH29	VTT	Power	
AH30	VTT	Power	
AH31	VCCSF	Power	
AH32	VCCSF	Power	
AH33	VSS	Power	
AH34	FSB0D_N[51]		
AH35	FSB0D_N[55]		
AH36	VSS	Power	
AH37	FSB0D_N[53]		
AH38	FSB0DP_N[0]		
AH39	VSS	Power	
AH40	FSB0DBI_N[2]		
AH41	FSB0D_N[41]		
AH42	VSS	Power	
AH43	FSB0D_N[43]		
AH44	FSB0D_N[40]		
AH45	VTT	Power	
AJ1	FSB2D_N[27]		
AJ2	VSS	Power	
AJ3	FSB2D_N[23]		
AJ4	FSB2DBI_N[1]		
AJ5	VSS	Power	
AJ6	FSB2STBN_N[1]		
AJ7	FSB2STBP_N[1]		
AJ8	VSS	Power	
AJ9	FSB2STBN_N[0]		
AJ10	FSB2STBP_N[0]		
AJ11	VSS	Power	
AJ12	FSB2D_N[12]		
AJ13	FSB2D_N[15]		
AJ14	VSS	Power	
AJ15	VCCSF	Power	
AJ16	VTT	Power	
AJ17	VTT	Power	
AJ18	VTT	Power	
AJ19	VTT	Power	
AJ20	VTT	Power	
AJ21	VTT	Power	
AJ22	VTT	Power	
AJ23	VTT	Power	
AJ24	VTT	Power	
AJ25	VTT	Power	
AJ26	VTT	Power	
AJ27	VTT	Power	
AJ28	VTT	Power	
AJ29	VTT	Power	
AJ30	VTT	Power	
AJ31	VCCSF	Power	
AJ32	VSS	Power	
AJ33	VCCSF	Power	
AJ34	FSB0D_N[48]		
AJ35	VSS	Power	
AJ36	FSB0STBP_N[3]		
AJ37	FSB0STBN_N[3]		
AJ38	VTT	Power	
AJ39	FSB0D_N[34]		
AJ40	FSB0D_N[33]		
AJ41	VSS	Power	
AJ42	FSB0D_N[39]		
AJ43	FSB0D_N[35]		
AJ44	VSS	Power	
AJ45	FSB0D_N[38]		
AK1	VTT	Power	
AK2	FSB2D_N[29]		
AK3	FSB2D_N[26]		
AK4	VSS	Power	
AK5	FSB2D_N[19]		
AK6	FSB2D_N[17]		
AK7	VSS	Power	
AK8	FSB2D_N[16]		
AK9	FSB2D_N[37]		
AK10	VSS	Power	
AK11	FSB2D_N[34]		
AK12	FSB2D_N[36]		
AK13	VTT	Power	
AK14	VCCSF	Power	
AK15	VCCSF	Power	
AK16	VTT	Power	
AK17	VTT	Power	
AK18	VTT	Power	
AK19	VTT	Power	
AK20	VTT	Power	
AK21	VTT	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 16 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AK22	VTT	Power	
AK23	VTT	Power	
AK24	VTT	Power	
AK25	VTT	Power	
AK26	VTT	Power	
AK27	VTT	Power	
AK28	VTT	Power	
AK29	VTT	Power	
AK30	VTT	Power	
AK31	VSS	Power	
AK32	VCCSF	Power	
AK33	VCCSF	Power	
AK34	VSS	Power	
AK35	FSB0STBP_N[1]		
AK36	FSB0D_N[52]		
AK37	VSS	Power	
AK38	FSB0D_N[24]		
AK39	FSB0D_N[36]		
AK40	VSS	Power	
AK41	FSB0D_N[32]		
AK42	FSB0D_N[37]		
AK43	VTT	Power	
AK44	FSB0STBN_N[2]		
AK45	FSB0STBP_N[2]		
AL1	FSB2D_N[25]		
AL2	FSB2D_N[28]		
AL3	VSS	Power	
AL4	FSB2D_N[30]		
AL5	FSB2D_N[24]		
AL6	VTT	Power	
AL7	FSB2D_N[31]		
AL8	FSB2D_N[39]		
AL9	VSS	Power	
AL10	FSB2STBN_N[2]		
AL11	FSB2D_N[32]		
AL12	VSS	Power	
AL13	FSB3DBI_N[0]		
AL14	FSB3D_N[1]		
AL15	VSS	Power	
AL16	VSS	Power	
AL17	VSS	Power	
AL18	VTT	Power	
AL19	VSS	Power	
AL20	VSS	Power	

Ball No.	Signal Name	Buffer Type	Direction
AL21	VCCSF	Power	
AL22	VCCSF	Power	
AL23	VCCSF	Power	
AL24	VCCSF	Power	
AL25	VCCSF	Power	
AL26	VSS	Power	
AL27	VSS	Power	
AL28	VSS	Power	
AL29	VSS	Power	
AL30	VTT	Power	
AL31	VCCSF	Power	
AL32	VCCSF	Power	
AL33	VSS	Power	
AL34	FSB0D_N[31]		
AL35	FSB0STBN_N[1]		
AL36	VTT	Power	
AL37	FSB0D_N[30]		
AL38	FSB0D_N[25]		
AL39	VSS	Power	
AL40	FSB0D_N[28]		
AL41	FSB0D_N[15]		
AL42	VSS	Power	
AL43	FSB0D_N[14]		
AL44	FSB0D_N[12]		
AL45	VSS	Power	
AM1	FSB2D_N[51]		
AM2	VSS	Power	
AM3	FSB2D_N[48]		
AM4	FSB2D_N[52]		
AM5	VSS	Power	
AM6	FSB2D_N[49]		
AM7	FSB2D_N[50]		
AM8	VSS	Power	
AM9	FSB2D_N[35]		
AM10	FSB2STBP_N[2]		
AM11	VTT	Power	
AM12	FSB2D_N[33]		
AM13	FSB3D_N[4]		
AM14	VSS	Power	
AM15	FSB3STBP_N[0]		
AM16	FSB3D_N[15]		
AM17	VSS	Power	
AM18	FSB3D_N[36]		
AM19	FSB3D_N[46]		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 17 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AM20	VSS	Power	
AM21	VCCSF	Power	
AM22	VCCSF	Power	
AM23	VCCSF	Power	
AM24	VCCSF	Power	
AM25	VCCSF	Power	
AM26	VSS	Power	
AM27	FSB1STBP_N[1]		
AM28	FSB1D_N[31]		
AM29	VSS	Power	
AM30	FSB1D_N[50]		
AM31	FSB1D_N[52]		
AM32	VSS	Power	
AM33	FSB1D_N[63]		
AM34	FSB1D_N[60]		
AM35	VSS	Power	
AM36	FSB0D_N[26]		
AM37	FSB0D_N[29]		
AM38	VSS	Power	
AM39	FSB0VREF[2]		
AM40	FSB0DBI_N[1]		
AM41	VTT	Power	
AM42	FSB0STBP_N[0]		
AM43	FSB0STBN_N[0]		
AM44	VSS	Power	
AM45	FSB0D_N[11]		
AN1	VSS	Power	
AN2	FSB2D_N[53]		
AN3	FSB2D_N[55]		
AN4	VTT	Power	
AN5	FSB2STBP_N[3]		
AN6	FSB2STBN_N[3]		
AN7	VSS	Power	
AN8	FSB2VREF[4]		
AN9	FSB2DBI_N[2]		
AN10	VSS	Power	
AN11	FSB2D_N[38]		
AN12	VSS	Power	
AN13	VSS	Power	
AN14	FSB3D_N[7]		
AN15	FSB3STBN_N[0]		
AN16	VTT	Power	
AN17	FSB3D_N[34]		
AN18	FSB3D_N[32]		

Ball No.	Signal Name	Buffer Type	Direction
AN19	VSS	Power	
AN20	FSB3D_N[47]		
AN21	VSS	Power	
AN22	VCCSF	Power	
AN23	VCCSF	Power	
AN24	VCCSF	Power	
AN25	VSS	Power	
AN26	FSB1DBI_N[1]		
AN27	FSB1STBN_N[1]		
AN28	VTT	Power	
AN29	FSB1D_N[24]		
AN30	FSB1D_N[49]		
AN31	VSS	Power	
AN32	FSB1D_N[58]		
AN33	FSB1D_N[59]		
AN34	VTT	Power	
AN35	FSB0D_N[18]		
AN36	FSB0D_N[27]		
AN37	VSS	Power	
AN38	FSB0D_N[22]		
AN39	FSB0D_N[23]		
AN40	VSS	Power	
AN41	FSB0D_N[9]		
AN42	FSB0D_N[5]		
AN43	VSS	Power	
AN44	FSB0D_N[10]		
AN45	FSB0D_N[13]		
AP1	FSB2D_N[54]		
AP2	FSB2DBI_N[3]		
AP3	VSS	Power	
AP4	FSB2D_N[56]		
AP5	FSB2D_N[57]		
AP6	VSS	Power	
AP7	FSB2D_N[43]		
AP8	FSB2D_N[40]		
AP9	VTT	Power	
AP10	FSB2D_N[42]		
AP11	FSB2D_N[41]		
AP12	VSS	Power	
AP13	FSB3BREQ1_N		
AP14	FSB3D_N[8]		
AP15	VSS	Power	
AP16	FSB3D_N[14]		
AP17	FSB3D_N[12]		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 18 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AP18	VSS	Power	
AP19	FSB3STBN_N[2]		
AP20	FSB3D_N[44]		
AP21	VTT	Power	
AP22	VCCSF	Power	
AP23	VCCSF	Power	
AP24	VCCSF	Power	
AP25	FSB1D_N[16]		
AP26	FSB1D_N[20]		
AP27	VSS	Power	
AP28	FSB1D_N[25]		
AP29	FSB1D_N[30]		
AP30	VSS	Power	
AP31	FSB1STBN_N[3]		
AP32	FSB1STBP_N[3]		
AP33	VSS	Power	
AP34	FSB1D_N[57]		
AP35	FSB0D_N[16]		
AP36	VSS	Power	
AP37	FSB0D_N[19]		
AP38	FSB0D_N[21]		
AP39	VTT	Power	
AP40	FSB0D_N[20]		
AP41	FSB0DBI_N[0]		
AP42	VSS	Power	
AP43	FSB0D_N[6]		
AP44	FSB0D_N[2]		
AP45	VSS	Power	
AR1	FSB2D_N[62]		
AR2	VTT	Power	
AR3	FSB2D_N[61]		
AR4	FSB2D_N[58]		
AR5	VSS	Power	
AR6	FSB2D_N[63]		
AR7	FSB2D_N[45]		
AR8	VSS	Power	
AR9	FSB2D_N[44]		
AR10	FSB2D_N[46]		
AR11	VSS	Power	
AR12	FSB3BREQ0_N		
AR13	FSB3D_N[0]		
AR14	VTT	Power	
AR15	FSB3D_N[6]		
AR16	FSB3D_N[5]		

Ball No.	Signal Name	Buffer Type	Direction
AR17	VSS	Power	
AR18	FSB3D_N[33]		
AR19	FSB3STBP_N[2]		
AR20	VSS	Power	
AR21	FSB3D_N[45]		
AR22	VCCSF	Power	
AR23	VCCSF	Power	
AR24	VCCSF	Power	
AR25	FSB1D_N[19]		
AR26	VTT	Power	
AR27	FSB1D_N[23]		
AR28	FSB1D_N[29]		
AR29	VSS	Power	
AR30	FSB1D_N[48]		
AR31	FSB1D_N[51]		
AR32	VTT	Power	
AR33	FSB1D_N[56]		
AR34	FSB1D_N[61]		
AR35	VSS	Power	
AR36	FSB0D_N[17]		
AR37	FSB1A_N[10]		
AR38	VSS	Power	
AR39	FSB1A_N[12]		
AR40	FSB1A_N[16]		
AR41	VSS	Power	
AR42	FSB0D_N[1]		
AR43	FSB0D_N[8]		
AR44	VTT	Power	
AR45	FSB0D_N[3]		
AT1	VSS	Power	
AT2	FSB3REQ_N[3]		
AT3	FSB3REQ_N[1]		
AT4	VSS	Power	
AT5	FSB2D_N[59]		
AT6	FSB2D_N[60]		
AT7	VTT	Power	
AT8	FSB3A_N[37]		
AT9	FSB2D_N[47]		
AT10	VSS	Power	
AT11	FSB3HIT_N		
AT12	FSB3RESET_N		
AT13	VSS	Power	
AT14	FSB3D_N[3]		
AT15	FSB3D_N[2]		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 19 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AT16	VSS	Power	
AT17	FSB3D_N[10]		
AT18	FSB3D_N[37]		
AT19	VTT	Power	
AT20	FSB3D_N[41]		
AT21	FSB3D_N[42]		
AT22	VCCSF	Power	
AT23	VCCSF	Power	
AT24	VCCSF	Power	
AT25	VSS	Power	
AT26	FSB1D_N[21]		
AT27	FSB1D_N[22]		
AT28	VSS	Power	
AT29	FSB1D_N[26]		
AT30	FSB1D_N[28]		
AT31	VSS	Power	
AT32	FSB1D_N[55]		
AT33	FSB1DBI_N[3]		
AT34	VSS	Power	
AT35	FSB1BPM_N[5]		
AT36	FSB1A_N[14]		
AT37	VTT	Power	
AT38	FSB1A_N[13]		
AT39	FSB1A_N[15]		
AT40	VSS	Power	
AT41	FSB1A_N[11]		
AT42	FSB1A_N[36]		
AT43	VSS	Power	
AT44	FSB0D_N[7]		
AT45	FSB0D_N[0]		
AU1	FSB3A_N[7]		
AU2	FSB3A_N[6]		
AU3	VSS	Power	
AU4	FSB3REQ_N[0]		
AU5	FSB3A_N[4]		
AU6	VSS	Power	
AU7	FSB3A_N[3]		
AU8	FSB3REQ_N[4]		
AU9	VSS	Power	
AU10	FSB3BPRI_N		
AU11	FSB3DEFER_N		
AU12	VTT	Power	
AU13	FSB3HITM_N		
AU14	FSB3VREF[2]		

Ball No.	Signal Name	Buffer Type	Direction
AU15	VSS	Power	
AU16	FSB3D_N[13]		
AU17	FSB3D_N[11]		
AU18	VSS	Power	
AU19	FSB3D_N[39]		
AU20	FSB3D_N[38]		
AU21	VSS	Power	
AU22	VSS	Power	
AU23	VCCSF	Power	
AU24	VSS	Power	
AU25	FSB1D_N[17]		
AU26	FSB1D_N[18]		
AU27	VSS	Power	
AU28	FSB1D_N[27]		
AU29	FSB1VREF[2]		
AU30	VTT	Power	
AU31	FSB1D_N[53]		
AU32	FSB1D_N[54]		
AU33	VSS	Power	
AU34	FSB1MCERR_N		
AU35	FSB1RSP_N		
AU36	VSS	Power	
AU37	FSB1A_N[4]		
AU38	FSB1A_N[8]		
AU39	VSS	Power	
AU40	FSB1ADSTB_N[0]		
AU41	FSB1A_N[9]		
AU42	VTT	Power	
AU43	FSB1A_N[32]		
AU44	FSB0D_N[4]		
AU45	VSS	Power	
AV1	FSB3ADSTB_N[0]		
AV2	VSS	Power	
AV3	FSB3A_N[5]		
AV4	FSB3A_N[9]		
AV5	VTT	Power	
AV6	FSB3A_N[16]		
AV7	FSB3REQ_N[2]		
AV8	VSS	Power	
AV9	FSB3A_N[14]		
AV10	FSB3VREF[0]		
AV11	VSS	Power	
AV12	FSB3RS_N[2]		
AV13	FSB3RSP_N		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 20 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AV14	VSS	Power	
AV15	FSB3D_N[9]		
AV16	FSB3D_N[31]		
AV17	VTT	Power	
AV18	FSB3VREF[4]		
AV19	FSB3D_N[35]		
AV20	VSS	Power	
AV21	FSB3D_N[40]		
AV22	VCCSF	Power	
AV23	VSS	Power	
AV24	VCCSF	Power	
AV25	FSB1DBI_N[0]		
AV26	VSS	Power	
AV27	FSB1D_N[14]		
AV28	FSB1D_N[15]		
AV29	VSS	Power	
AV30	FSB1D_N[36]		
AV31	FSB1D_N[46]		
AV32	VSS	Power	
AV33	FSB1D_N[62]		
AV34	FSB1AP_N[1]		
AV35	VTT	Power	
AV36	FSB1A_N[37]		
AV37	FSB1REQ_N[4]		
AV38	VSS	Power	
AV39	FSB1REQ_N[0]		
AV40	FSB1A_N[6]		
AV41	VSS	Power	
AV42	FSB1A_N[5]		
AV43	FSB1A_N[33]		
AV44	VSS	Power	
AV45	FSB1A_N[38]		
AW1	VSS	Power	
AW2	FSB3A_N[8]		
AW3	FSB3A_N[36]		
AW4	VSS	Power	
AW5	FSB3A_N[12]		
AW6	FSB3A_N[13]		
AW7	VSS	Power	
AW8	FSB3A_N[10]		
AW9	RSVD		
AW10	VTT	Power	
AW11	FSB3RS_N[1]		
AW12	FSB3BNR_N		
AW13	VSS	Power	
AW14	FSB3D_N[16]		
AW15	FSB3D_N[17]		
AW16	VSS	Power	
AW17	FSB3DBI_N[1]		
AW18	FSB3D_N[50]		
AW19	VSS	Power	
AW20	FSB3DBI_N[2]		
AW21	FSB3D_N[43]		
AW22	VSS	Power	
AW23	VSS	Power	
AW24	VSS	Power	
AW25	VSS	Power	
AW26	FSB1STBP_N[0]		
AW27	FSB1D_N[12]		
AW28	VTT	Power	
AW29	FSB1D_N[34]		
AW30	FSB1D_N[32]		
AW31	VSS	Power	
AW32	FSB1D_N[47]		
AW33	FSB1BINIT_N		
AW34	VSS	Power	
AW35	FSB1AP_N[0]		
AW36	FSB1BPM_N[4]		
AW37	VSS	Power	
AW38	FSB1REQ_N[2]		
AW39	FSB1REQ_N[1]		
AW40	VTT	Power	
AW41	FSB1A_N[7]		
AW42	FSB1A_N[35]		
AW43	VSS	Power	
AW44	FSB1A_N[31]		
AW45	FSB1A_N[27]		
AY1	FSB3A_N[11]		
AY2	FSB3A_N[15]		
AY3	VTT	Power	
AY4	FSB3A_N[18]		
AY5	FSB3A_N[19]		
AY6	VSS	v	
AY7	FSB3A_N[39]		
AY8	FSB3A_N[17]		
AY9	VSS	Power	
AY10	FSB3DRVCRES		
AY11	FSB3RS_N[0]		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 21 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AY12	VSS	Power	
AY13	FSB3BPM_N[5]		
AY14	FSB3D_N[19]		
AY15	VTT	Power	
AY16	FSB3STBP_N[1]		
AY17	FSB3D_N[29]		
AY18	VSS	Power	
AY19	FSB3D_N[49]		
AY20	FSB3D_N[59]		
AY21	VSS	Power	
AY22	FSB3D_N[60]		
AY23	VSS	Power	
AY24	VSS	Power	
AY25	FSB1D_N[4]		
AY26	FSB1STBN_N[0]		
AY27	VSS	Power	
AY28	FSB1D_N[33]		
AY29	FSB1D_N[37]		
AY30	VSS	Power	
AY31	FSB1D_N[44]		
AY32	FSB1VREF[4]		
AY33	VTT	Power	
AY34	FSB1DRDY_N		
AY35	FSB1BREQ0_N		
AY36	VSS	Power	
AY37	RSVD		
AY38	FSB1A_N[3]		
AY39	VSS	Power	
AY40	FSB1REQ_N[3]		
AY41	FSB1A_N[20]		
AY42	VSS	Power	
AY43	FSB1A_N[34]		
AY44	FSB1A_N[26]		
AY45	VTT	Power	
BA1	VTT	Power	
BA2	VSS	Power	
BA3	FSB3A_N[24]		
BA4	FSB3A_N[28]		
BA5	VSS	Power	
BA6	FSB3A_N[25]		
BA7	FSB3ADSTB_N[1]		
BA8	VTT	Power	
BA9	FSB3ODTCRES		
BA10	FSB3TRDY_N		
BA11	VSS	Power	
BA12	FSB3ADS_N		
BA13	FSB3DP_N[2]		
BA14	VSS	Power	
BA15	FSB3D_N[18]		
BA16	FSB3STBN_N[1]		
BA17	VSS	Power	
BA18	FSB3D_N[52]		
BA19	FSB3D_N[51]		
BA20	VTT	Power	
BA21	FSB3D_N[57]		
BA22	FSB3D_N[63]		
BA23	VSS	Power	
BA24	VSS	Power	
BA25	FSB1D_N[1]		
BA26	VTT	Power	
BA27	FSB1D_N[13]		
BA28	FSB1D_N[11]		
BA29	VSS	Power	
BA30	FSB1STBN_N[2]		
BA31	FSB1D_N[45]		
BA32	VSS	Power	
BA33	FSB1RS_N[2]		
BA34	FSB1BREQ1_N		
BA35	VSS	Power	
BA36	FSB1LOCK_N		
BA37	FSB1VREF[0]		
BA38	VTT	Power	
BA39	FSB1A_N[30]		
BA40	FSB1ADSTB_N[1]		
BA41	VSS	Power	
BA42	FSB1A_N[22]		
BA43	FSB1A_N[21]		
BA44	VSS	Power	
BA45	VTT	Power	
BB1	VSS	Power	
BB2	VTT	Power	
BB3	FSB3A_N[29]		
BB4	VSS	Power	
BB5	FSB3A_N[23]		
BB6	FSB3A_N[21]		
BB7	VSS	Power	
BB8	FSB3A_N[20]		
BB9	FSB3SLWCRES		

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 22 of 23)

Ball No.	Signal Name	Buffer Type	Direction
BB10	VSS	Power	
BB11	FSB3DBSY_N		
BB12	FSB3AP_N[1]		
BB13	VTT	Power	
BB14	FSB3D_N[21]		
BB15	FSB3D_N[22]		
BB16	VSS	Power	
BB17	FSB3D_N[24]		
BB18	FSB3D_N[48]		
BB19	VSS	Power	
BB20	FSB3D_N[54]		
BB21	FSB3D_N[58]		
BB22	VSS	Power	
BB23	VSS	Power	
BB24	VSS	Power	
BB25	VSS	Power	
BB26	FSB1D_N[6]		
BB27	FSB1D_N[2]		
BB28	VSS	Power	
BB29	FSB1DBI_N[2]		
BB30	FSB1STBP_N[2]		
BB31	VTT	Power	
BB32	FSB1DP_N[3]		
BB33	FSB1BPRI_N		
BB34	VSS	Power	
BB35	FSB1ADS_N		
BB36	FSB1RESET_N		
BB37	VSS	Power	
BB38	FSB1A_N[17]		
BB39	FSB1A_N[18]		
BB40	VSS	Power	
BB41	FSB1A_N[29]		
BB42	FSB1A_N[23]		
BB43	VTT	Power	
BB44	VTT	Power	
BB45	VSS	Power	
BC1	VSS	Power	
BC2	VSS	Power	
BC3	VTT	Power	
BC4	FSB3A_N[30]		
BC5	FSB3A_N[22]		
BC6	VTT	Power	
BC7	FSB3A_N[35]		
BC8	FSB3A_N[31]		

Ball No.	Signal Name	Buffer Type	Direction
BC9	VSS	Power	
BC10	FSB3DRDY_N		
BC11	FSB3BINIT_N		
BC12	VSS	Power	
BC13	FSB3DP_N[0]		
BC14	FSB3D_N[20]		
BC15	VSS	Power	
BC16	FSB3D_N[28]		
BC17	FSB3D_N[30]		
BC18	VTT	Power	
BC19	FSB3D_N[53]		
BC20	FSB3D_N[62]		
BC21	VSS	Power	
BC22	CORECLKP		
BC23	RSVD		
BC24	VSS	Power	
BC25	FSB1D_N[7]		
BC26	FSB1D_N[8]		
BC27	VSS	Power	
BC28	FSB1D_N[39]		
BC29	FSB1D_N[35]		
BC30	VSS	Power	
BC31	FSB1D_N[43]		
BC32	FSB1DP_N[0]		
BC33	VSS	Power	
BC34	FSB1HITM_N		
BC35	FSB1TRDY_N		
BC36	VTT	Power	
BC37	FSB1DBSY_N		
BC38	FSB1DRVCRES		
BC39	VSS	Power	
BC40	FSB1A_N[28]		
BC41	FSB1A_N[25]		
BC42	VSS	Power	
BC43	VTT	Power	
BC44	VSS	Power	
BC45	VSS	Power	
BD2	VSS	Power	
BD3	VSS	Power	
BD4	VTT	Power	
BD5	VSS	Power	
BD6	FSB3A_N[34]		
BD7	FSB3A_N[27]		
BD8	VSS	Power	

Table 5-1. Intel® 7300 Chipset MCH Signals By Ball Number (Sheet 23 of 23)

Ball No.	Signal Name	Buffer Type	Direction
BD9	FSB3A_N[38]		
BD10	FSB3LOCK_N		
BD11	VTT	Power	
BD12	FSB3BPM_N[4]		
BD13	FSB3DP_N[1]		
BD14	VSS	Power	
BD15	FSB3D_N[23]		
BD16	FSB3D_N[26]		
BD17	VSS	Power	
BD18	FSB3STBN_N[3]		
BD19	FSB3D_N[55]		
BD20	VSS	Power	
BD21	FSB3D_N[56]		
BD22	CORECLKN		
BD23	VSS	Power	
BD24	COREVCCA		
BD25	FSB1D_N[0]		
BD26	VSS	Power	
BD27	FSB1D_N[5]		
BD28	FSB1D_N[10]		
BD29	VTT	Power	
BD30	FSB1D_N[41]		
BD31	FSB1D_N[42]		
BD32	VSS	Power	
BD33	FSB1HIT_N		
BD34	FSB1DEFER_N		
BD35	VSS	Power	
BD36	FSB1BNR_N		
BD37	FSB1SLWCRES		
BD38	VSS	Power	
BD39	FSB1A_N[39]		
BD40	FSB1A_N[24]		
BD41	VTT	Power	
BD42	VTT	Power	
BD43	VSS	Power	
BD44	VSS	Power	
BE3	VSS	Power	
BE4	VSS	Power	
BE5	VTT	Power	
BE6	FSB3A_N[26]		
BE7	VSS	Power	
BE8	FSB3A_N[33]		
BE9	FSB3A_N[32]		
BE10	VSS	Power	
BE11	FSB3AP_N[0]		
BE12	FSB3MCERR_N		
BE13	VSS	Power	
BE14	FSB3DP_N[3]		
BE15	FSB3D_N[27]		
BE16	VTT	Power	
BE17	FSB3D_N[25]		
BE18	FSB3STBP_N[3]		
BE19	VSS	Power	
BE20	FSB3DBI_N[3]		
BE21	FSB3D_N[61]		
BE22	VSS	Power	
BE23	COREVSSA		
BE24	FSBVCCA		
BE25	VSS	Power	
BE26	FSB1D_N[3]		
BE27	FSB1D_N[9]		
BE28	VSS	Power	
BE29	FSB1D_N[38]		
BE30	FSB1D_N[40]		
BE31	VSS	Power	
BE32	FSB1DP_N[1]		
BE33	FSB1DP_N[2]		
BE34	VTT	Power	
BE35	FSB1RS_N[1]		
BE36	FSB1RS_N[0]		
BE37	VSS	Power	
BE38	FSB1ODTCRES		
BE39	FSB1A_N[19]		
BE40	VSS	Power	
BE41	VTT	Power	
BE42	VSS	Power	
BE43	VSS	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 1 of 23)

Ball No.	Signal Name	Buffer Type	Direction
P6	CFGSMBCLK		
P7	CFGSMBDATA		
BD22	CORECLKN		
BC22	CORECLKP		
BD24	COREVCCA		
BE23	COREVSSA		
N4	ERR_N[0]		
N5	ERR_N[1]		
P4	ERR_N[2]		
A38	FBD01BGBIASEXT		
J45	FBD01CLKN		
H45	FBD01CLKP		
A37	FBD01ICOMPBIAS		
A40	FBD01RESIN		
F45	FBD01VCCA		
F44	FBD01VSSA		
V33	FBD0NBIN[0]		
U34	FBD0NBIN[1]		
P40	FBD0NBIN[10]		
R39	FBD0NBIN[11]		
M36	FBD0NBIN[12]		
N35	FBD0NBIN[13]		
P31	FBD0NBIN[2]		
N32	FBD0NBIN[3]		
R33	FBD0NBIN[4]		
P34	FBD0NBIN[5]		
R36	FBD0NBIN[6]		
N38	FBD0NBIN[7]		
M39	FBD0NBIN[8]		
P37	FBD0NBIN[9]		
V32	FBD0NBIP[0]		
U33	FBD0NBIP[1]		
N40	FBD0NBIP[10]		
P39	FBD0NBIP[11]		
M35	FBD0NBIP[12]		
N34	FBD0NBIP[13]		
P30	FBD0NBIP[2]		
N31	FBD0NBIP[3]		
R32	FBD0NBIP[4]		
P33	FBD0NBIP[5]		
R35	FBD0NBIP[6]		
N37	FBD0NBIP[7]		
M38	FBD0NBIP[8]		
P36	FBD0NBIP[9]		

Ball No.	Signal Name	Buffer Type	Direction
N44	FBD0SBON[0]		
P43	FBD0SBON[1]		
R45	FBD0SBON[2]		
T44	FBD0SBON[3]		
U43	FBD0SBON[4]		
R42	FBD0SBON[5]		
T41	FBD0SBON[6]		
U40	FBD0SBON[7]		
V39	FBD0SBON[8]		
V45	FBD0SBON[9]		
M44	FBD0SBOP[0]		
N43	FBD0SBOP[1]		
P45	FBD0SBOP[2]		
R44	FBD0SBOP[3]		
T43	FBD0SBOP[4]		
P42	FBD0SBOP[5]		
R41	FBD0SBOP[6]		
T40	FBD0SBOP[7]		
U39	FBD0SBOP[8]		
U45	FBD0SBOP[9]		
M33	FBD1NBIN[0]		
L34	FBD1NBIN[1]		
E43	FBD1NBIN[10]		
F42	FBD1NBIN[11]		
E40	FBD1NBIN[12]		
F39	FBD1NBIN[13]		
K35	FBD1NBIN[2]		
J36	FBD1NBIN[3]		
H37	FBD1NBIN[4]		
G38	FBD1NBIN[5]		
B40	FBD1NBIN[6]		
D41	FBD1NBIN[7]		
C42	FBD1NBIN[8]		
G41	FBD1NBIN[9]		
M32	FBD1NBIP[0]		
L33	FBD1NBIP[1]		
E42	FBD1NBIP[10]		
F41	FBD1NBIP[11]		
E39	FBD1NBIP[12]		
F38	FBD1NBIP[13]		
K34	FBD1NBIP[2]		
J35	FBD1NBIP[3]		
H36	FBD1NBIP[4]		
G37	FBD1NBIP[5]		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 2 of 23)

Ball No.	Signal Name	Buffer Type	Direction
B39	FBD1NBIP[6]		
D40	FBD1NBIP[7]		
C41	FBD1NBIP[8]		
G40	FBD1NBIP[9]		
G44	FBD1SBON[0]		
H43	FBD1SBON[1]		
J42	FBD1SBON[2]		
K44	FBD1SBON[3]		
L43	FBD1SBON[4]		
M42	FBD1SBON[5]		
N41	FBD1SBON[6]		
K40	FBD1SBON[7]		
L39	FBD1SBON[8]		
M45	FBD1SBON[9]		
G43	FBD1SBOP[0]		
H42	FBD1SBOP[1]		
J41	FBD1SBOP[2]		
J44	FBD1SBOP[3]		
K43	FBD1SBOP[4]		
L42	FBD1SBOP[5]		
M41	FBD1SBOP[6]		
K41	FBD1SBOP[7]		
L40	FBD1SBOP[8]		
L45	FBD1SBOP[9]		
B22	FBD23BGBIASEXT		
A32	FBD23CLKN		
A31	FBD23CLKP		
B21	FBD23ICOMPBIAS		
A22	FBD23RESIN		
A28	FBD23VCCA		
A29	FBD23VSSA		
M29	FBD2NBIN[0]		
N28	FBD2NBIN[1]		
F33	FBD2NBIN[10]		
E34	FBD2NBIN[11]		
G31	FBD2NBIN[12]		
H30	FBD2NBIN[13]		
L30	FBD2NBIN[2]		
L27	FBD2NBIN[3]		
K28	FBD2NBIN[4]		
J29	FBD2NBIN[5]		
G29	FBD2NBIN[6]		
F30	FBD2NBIN[7]		
E31	FBD2NBIN[8]		

Ball No.	Signal Name	Buffer Type	Direction
D32	FBD2NBIN[9]		
M30	FBD2NBIP[0]		
N29	FBD2NBIP[1]		
F32	FBD2NBIP[10]		
E33	FBD2NBIP[11]		
H31	FBD2NBIP[12]		
J30	FBD2NBIP[13]		
L31	FBD2NBIP[2]		
M27	FBD2NBIP[3]		
L28	FBD2NBIP[4]		
K29	FBD2NBIP[5]		
G28	FBD2NBIP[6]		
F29	FBD2NBIP[7]		
E30	FBD2NBIP[8]		
D31	FBD2NBIP[9]		
A35	FBD2SBON[0]		
D35	FBD2SBON[1]		
B37	FBD2SBON[2]		
C36	FBD2SBON[3]		
C39	FBD2SBON[4]		
E37	FBD2SBON[5]		
F36	FBD2SBON[6]		
G35	FBD2SBON[7]		
H34	FBD2SBON[8]		
D38	FBD2SBON[9]		
A34	FBD2SBOP[0]		
D34	FBD2SBOP[1]		
B36	FBD2SBOP[2]		
C35	FBD2SBOP[3]		
C38	FBD2SBOP[4]		
E36	FBD2SBOP[5]		
F35	FBD2SBOP[6]		
G34	FBD2SBOP[7]		
H33	FBD2SBOP[8]		
D37	FBD2SBOP[9]		
N25	FBD3NBIN[0]		
L24	FBD3NBIN[1]		
G26	FBD3NBIN[10]		
E25	FBD3NBIN[11]		
H24	FBD3NBIN[12]		
J23	FBD3NBIN[13]		
N22	FBD3NBIN[2]		
M23	FBD3NBIN[3]		
L21	FBD3NBIN[4]		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 3 of 23)

Ball No.	Signal Name	Buffer Type	Direction
K22	FBD3NBIN[5]		
H22	FBD3NBIN[6]		
G23	FBD3NBIN[7]		
E22	FBD3NBIN[8]		
F24	FBD3NBIN[9]		
N26	FBD3NBIP[0]		
L25	FBD3NBIP[1]		
G25	FBD3NBIP[10]		
E24	FBD3NBIP[11]		
J24	FBD3NBIP[12]		
K23	FBD3NBIP[13]		
N23	FBD3NBIP[2]		
M24	FBD3NBIP[3]		
M21	FBD3NBIP[4]		
L22	FBD3NBIP[5]		
H21	FBD3NBIP[6]		
G22	FBD3NBIP[7]		
E21	FBD3NBIP[8]		
F23	FBD3NBIP[9]		
B25	FBD3SBON[0]		
A26	FBD3SBON[1]		
D26	FBD3SBON[2]		
B28	FBD3SBON[3]		
C27	FBD3SBON[4]		
B31	FBD3SBON[5]		
D29	FBD3SBON[6]		
E28	FBD3SBON[7]		
F27	FBD3SBON[8]		
C30	FBD3SBON[9]		
B24	FBD3SBOP[0]		
A25	FBD3SBOP[1]		
D25	FBD3SBOP[2]		
B27	FBD3SBOP[3]		
C26	FBD3SBOP[4]		
B30	FBD3SBOP[5]		
D28	FBD3SBOP[6]		
E27	FBD3SBOP[7]		
F26	FBD3SBOP[8]		
C29	FBD3SBOP[9]		
V35	FSB0A_N[10]		
W34	FSB0A_N[11]		
W35	FSB0A_N[12]		
V38	FSB0A_N[13]		
W32	FSB0A_N[14]		
W37	FSB0A_N[15]		
W38	FSB0A_N[16]		
AB38	FSB0A_N[17]		
AB40	FSB0A_N[18]		
AA39	FSB0A_N[19]		
Y45	FSB0A_N[20]		
Y43	FSB0A_N[21]		
Y42	FSB0A_N[22]		
AA44	FSB0A_N[23]		
AB41	FSB0A_N[24]		
AA41	FSB0A_N[25]		
W44	FSB0A_N[26]		
W40	FSB0A_N[27]		
Y40	FSB0A_N[28]		
AA42	FSB0A_N[29]		
AD33	FSB0A_N[3]		
AA45	FSB0A_N[30]		
V44	FSB0A_N[31]		
U42	FSB0A_N[32]		
V42	FSB0A_N[33]		
W41	FSB0A_N[34]		
W43	FSB0A_N[35]		
Y37	FSB0A_N[36]		
AB32	FSB0A_N[37]		
V41	FSB0A_N[38]		
AA38	FSB0A_N[39]		
Y33	FSB0A_N[4]		
AB37	FSB0A_N[5]		
AB35	FSB0A_N[6]		
Y34	FSB0A_N[7]		
Y36	FSB0A_N[8]		
AA36	FSB0A_N[9]		
AC43	FSB0ADS_N		
AA35	FSB0ADSTB_N[0]		
Y39	FSB0ADSTB_N[1]		
AB44	FSB0AP_N[0]		
AB43	FSB0AP_N[1]		
AD41	FSB0BINIT_N		
AE43	FSB0BNR_N		
AC40	FSB0BPM_N[4]		
AD38	FSB0BPM_N[5]		
AE40	FSB0BPRI_N		
AC36	FSB0BREQ0_N		
AD36	FSB0BREQ1_N		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 4 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AT45	FSB0D_N[0]		
AR42	FSB0D_N[1]		
AN44	FSB0D_N[10]		
AM45	FSB0D_N[11]		
AL44	FSB0D_N[12]		
AN45	FSB0D_N[13]		
AL43	FSB0D_N[14]		
AL41	FSB0D_N[15]		
AP35	FSB0D_N[16]		
AR36	FSB0D_N[17]		
AN35	FSB0D_N[18]		
AP37	FSB0D_N[19]		
AP44	FSB0D_N[2]		
AP40	FSB0D_N[20]		
AP38	FSB0D_N[21]		
AN38	FSB0D_N[22]		
AN39	FSB0D_N[23]		
AK38	FSB0D_N[24]		
AL38	FSB0D_N[25]		
AM36	FSB0D_N[26]		
AN36	FSB0D_N[27]		
AL40	FSB0D_N[28]		
AM37	FSB0D_N[29]		
AR45	FSB0D_N[3]		
AL37	FSB0D_N[30]		
AL34	FSB0D_N[31]		
AK41	FSB0D_N[32]		
AJ40	FSB0D_N[33]		
AJ39	FSB0D_N[34]		
AJ43	FSB0D_N[35]		
AK39	FSB0D_N[36]		
AK42	FSB0D_N[37]		
AJ45	FSB0D_N[38]		
AJ42	FSB0D_N[39]		
AU44	FSB0D_N[4]		
AH44	FSB0D_N[40]		
AH41	FSB0D_N[41]		
AG45	FSB0D_N[42]		
AH43	FSB0D_N[43]		
AG42	FSB0D_N[44]		
AG44	FSB0D_N[45]		
AG39	FSB0D_N[46]		
AG41	FSB0D_N[47]		
AJ34	FSB0D_N[48]		
AG33	FSB0D_N[49]		
AN42	FSB0D_N[5]		
AG32	FSB0D_N[50]		
AH34	FSB0D_N[51]		
AK36	FSB0D_N[52]		
AH37	FSB0D_N[53]		
AG36	FSB0D_N[54]		
AH35	FSB0D_N[55]		
AF36	FSB0D_N[56]		
AE34	FSB0D_N[57]		
AE35	FSB0D_N[58]		
AD35	FSB0D_N[59]		
AP43	FSB0D_N[6]		
AE32	FSB0D_N[60]		
AF34	FSB0D_N[61]		
AF37	FSB0D_N[62]		
AF33	FSB0D_N[63]		
AT44	FSB0D_N[7]		
AR43	FSB0D_N[8]		
AN41	FSB0D_N[9]		
AP41	FSB0DBI_N[0]		
AM40	FSB0DBI_N[1]		
AH40	FSB0DBI_N[2]		
AG35	FSB0DBI_N[3]		
AC45	FSB0DBSY_N		
AF42	FSB0DEFER_N		
AH38	FSB0DP_N[0]		
AE37	FSB0DP_N[1]		
AF39	FSB0DP_N[2]		
AE38	FSB0DP_N[3]		
AD42	FSB0DRDY_N		
AF40	FSB0HIT_N		
AF45	FSB0HITM_N		
AC42	FSB0LOCK_N		
AD39	FSB0MCERR_N		
AB34	FSB0REQ_N[0]		
AA33	FSB0REQ_N[1]		
AC33	FSB0REQ_N[2]		
AC34	FSB0REQ_N[3]		
AA32	FSB0REQ_N[4]		
AD45	FSB0RESET_N		
AE44	FSB0RS_N[0]		
AF43	FSB0RS_N[1]		
AE41	FSB0RS_N[2]		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 5 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AC39	FSB0RSP_N		
AM43	FSB0STBN_N[0]		
AL35	FSB0STBN_N[1]		
AK44	FSB0STBN_N[2]		
AJ37	FSB0STBN_N[3]		
AM42	FSB0STBP_N[0]		
AK35	FSB0STBP_N[1]		
AK45	FSB0STBP_N[2]		
AJ36	FSB0STBP_N[3]		
AD44	FSB0TRDY_N		
AC37	FSB0VREF[0]		
AM39	FSB0VREF[2]		
AG38	FSB0VREF[4]		
AR37	FSB1A_N[10]		
AT41	FSB1A_N[11]		
AR39	FSB1A_N[12]		
AT38	FSB1A_N[13]		
AT36	FSB1A_N[14]		
AT39	FSB1A_N[15]		
AR40	FSB1A_N[16]		
BB38	FSB1A_N[17]		
BB39	FSB1A_N[18]		
BE39	FSB1A_N[19]		
AY41	FSB1A_N[20]		
BA43	FSB1A_N[21]		
BA42	FSB1A_N[22]		
BB42	FSB1A_N[23]		
BD40	FSB1A_N[24]		
BC41	FSB1A_N[25]		
AY44	FSB1A_N[26]		
AW45	FSB1A_N[27]		
BC40	FSB1A_N[28]		
BB41	FSB1A_N[29]		
AY38	FSB1A_N[3]		
BA39	FSB1A_N[30]		
AW44	FSB1A_N[31]		
AU43	FSB1A_N[32]		
AV43	FSB1A_N[33]		
AY43	FSB1A_N[34]		
AW42	FSB1A_N[35]		
AT42	FSB1A_N[36]		
AV36	FSB1A_N[37]		
AV45	FSB1A_N[38]		
BD39	FSB1A_N[39]		
AU37	FSB1A_N[4]		
AV42	FSB1A_N[5]		
AV40	FSB1A_N[6]		
AW41	FSB1A_N[7]		
AU38	FSB1A_N[8]		
AU41	FSB1A_N[9]		
BB35	FSB1ADS_N		
AU40	FSB1ADSTB_N[0]		
BA40	FSB1ADSTB_N[1]		
AW35	FSB1AP_N[0]		
AV34	FSB1AP_N[1]		
AW33	FSB1BINIT_N		
BD36	FSB1BNR_N		
AW36	FSB1BPM_N[4]		
AT35	FSB1BPM_N[5]		
BB33	FSB1BPRI_N		
AY35	FSB1BREQ0_N		
BA34	FSB1BREQ1_N		
BD25	FSB1D_N[0]		
BA25	FSB1D_N[1]		
BD28	FSB1D_N[10]		
BA28	FSB1D_N[11]		
AW27	FSB1D_N[12]		
BA27	FSB1D_N[13]		
AV27	FSB1D_N[14]		
AV28	FSB1D_N[15]		
AP25	FSB1D_N[16]		
AU25	FSB1D_N[17]		
AU26	FSB1D_N[18]		
AR25	FSB1D_N[19]		
BB27	FSB1D_N[2]		
AP26	FSB1D_N[20]		
AT26	FSB1D_N[21]		
AT27	FSB1D_N[22]		
AR27	FSB1D_N[23]		
AN29	FSB1D_N[24]		
AP28	FSB1D_N[25]		
AT29	FSB1D_N[26]		
AU28	FSB1D_N[27]		
AT30	FSB1D_N[28]		
AR28	FSB1D_N[29]		
BE26	FSB1D_N[3]		
AP29	FSB1D_N[30]		
AM28	FSB1D_N[31]		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 6 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AW30	FSB1D_N[32]		
AY28	FSB1D_N[33]		
AW29	FSB1D_N[34]		
BC29	FSB1D_N[35]		
AV30	FSB1D_N[36]		
AY29	FSB1D_N[37]		
BE29	FSB1D_N[38]		
BC28	FSB1D_N[39]		
AY25	FSB1D_N[4]		
BE30	FSB1D_N[40]		
BD30	FSB1D_N[41]		
BD31	FSB1D_N[42]		
BC31	FSB1D_N[43]		
AY31	FSB1D_N[44]		
BA31	FSB1D_N[45]		
AV31	FSB1D_N[46]		
AW32	FSB1D_N[47]		
AR30	FSB1D_N[48]		
AN30	FSB1D_N[49]		
BD27	FSB1D_N[5]		
AM30	FSB1D_N[50]		
AR31	FSB1D_N[51]		
AM31	FSB1D_N[52]		
AU31	FSB1D_N[53]		
AU32	FSB1D_N[54]		
AT32	FSB1D_N[55]		
AR33	FSB1D_N[56]		
AP34	FSB1D_N[57]		
AN32	FSB1D_N[58]		
AN33	FSB1D_N[59]		
BB26	FSB1D_N[6]		
AM34	FSB1D_N[60]		
AR34	FSB1D_N[61]		
AV33	FSB1D_N[62]		
AM33	FSB1D_N[63]		
BC25	FSB1D_N[7]		
BC26	FSB1D_N[8]		
BE27	FSB1D_N[9]		
AV25	FSB1DBI_N[0]		
AN26	FSB1DBI_N[1]		
BB29	FSB1DBI_N[2]		
AT33	FSB1DBI_N[3]		
BC37	FSB1DBSY_N		
BD34	FSB1DEFER_N		

Ball No.	Signal Name	Buffer Type	Direction
BC32	FSB1DP_N[0]		
BE32	FSB1DP_N[1]		
BE33	FSB1DP_N[2]		
BB32	FSB1DP_N[3]		
AY34	FSB1DRDY_N		
BC38	FSB1DRVCRES		
BD33	FSB1HIT_N		
BC34	FSB1HITM_N		
BA36	FSB1LOCK_N		
AU34	FSB1MCERR_N		
BE38	FSB1ODTCRES		
AV39	FSB1REQ_N[0]		
AW39	FSB1REQ_N[1]		
AW38	FSB1REQ_N[2]		
AY40	FSB1REQ_N[3]		
AV37	FSB1REQ_N[4]		
BB36	FSB1RESET_N		
BE36	FSB1RS_N[0]		
BE35	FSB1RS_N[1]		
BA33	FSB1RS_N[2]		
AU35	FSB1RSP_N		
BD37	FSB1SLWCRES		
AY26	FSB1STBN_N[0]		
AN27	FSB1STBN_N[1]		
BA30	FSB1STBN_N[2]		
AP31	FSB1STBN_N[3]		
AW26	FSB1STBP_N[0]		
AM27	FSB1STBP_N[1]		
BB30	FSB1STBP_N[2]		
AP32	FSB1STBP_N[3]		
BC35	FSB1TRDY_N		
BA37	FSB1VREF[0]		
AU29	FSB1VREF[2]		
AY32	FSB1VREF[4]		
AA14	FSB2A_N[10]		
AC10	FSB2A_N[11]		
AB13	FSB2A_N[12]		
AC13	FSB2A_N[13]		
AB14	FSB2A_N[14]		
AA12	FSB2A_N[15]		
AC12	FSB2A_N[16]		
W7	FSB2A_N[17]		
V5	FSB2A_N[18]		
Y6	FSB2A_N[19]		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 7 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AB1	FSB2A_N[20]		
AB2	FSB2A_N[21]		
AA3	FSB2A_N[22]		
Y3	FSB2A_N[23]		
W4	FSB2A_N[24]		
Y4	FSB2A_N[25]		
AA5	FSB2A_N[26]		
AB7	FSB2A_N[27]		
W5	FSB2A_N[28]		
Y1	FSB2A_N[29]		
Y13	FSB2A_N[3]		
AA2	FSB2A_N[30]		
AC1	FSB2A_N[31]		
AC7	FSB2A_N[32]		
AC3	FSB2A_N[33]		
AB4	FSB2A_N[34]		
AB5	FSB2A_N[35]		
AB11	FSB2A_N[36]		
W14	FSB2A_N[37]		
AC4	FSB2A_N[38]		
Y7	FSB2A_N[39]		
W10	FSB2A_N[4]		
AA9	FSB2A_N[5]		
AA8	FSB2A_N[6]		
Y10	FSB2A_N[7]		
AA11	FSB2A_N[8]		
AB8	FSB2A_N[9]		
AD2	FSB2ADS_N		
Y12	FSB2ADSTB_N[0]		
AA6	FSB2ADSTB_N[1]		
AF4	FSB2AP_N[0]		
AF1	FSB2AP_N[1]		
AF6	FSB2BINIT_N		
AE5	FSB2BNR_N		
AF3	FSB2BPM_N[4]		
AG5	FSB2BPM_N[5]		
AD9	FSB2BPRI_N		
AD11	FSB2BREQ0_N		
AD12	FSB2BREQ1_N		
AF12	FSB2D_N[0]		
AF13	FSB2D_N[1]		
AH10	FSB2D_N[10]		
AH11	FSB2D_N[11]		
AJ12	FSB2D_N[12]		
AG14	FSB2D_N[13]		
AH13	FSB2D_N[14]		
AJ13	FSB2D_N[15]		
AK8	FSB2D_N[16]		
AK6	FSB2D_N[17]		
AH5	FSB2D_N[18]		
AK5	FSB2D_N[19]		
AG11	FSB2D_N[2]		
AH1	FSB2D_N[20]		
AH4	FSB2D_N[21]		
AH2	FSB2D_N[22]		
AJ3	FSB2D_N[23]		
AL5	FSB2D_N[24]		
AL1	FSB2D_N[25]		
AK3	FSB2D_N[26]		
AJ1	FSB2D_N[27]		
AL2	FSB2D_N[28]		
AK2	FSB2D_N[29]		
AF10	FSB2D_N[3]		
AL4	FSB2D_N[30]		
AL7	FSB2D_N[31]		
AL11	FSB2D_N[32]		
AM12	FSB2D_N[33]		
AK11	FSB2D_N[34]		
AM9	FSB2D_N[35]		
AK12	FSB2D_N[36]		
AK9	FSB2D_N[37]		
AN11	FSB2D_N[38]		
AL8	FSB2D_N[39]		
AE13	FSB2D_N[4]		
AP8	FSB2D_N[40]		
AP11	FSB2D_N[41]		
AP10	FSB2D_N[42]		
AP7	FSB2D_N[43]		
AR9	FSB2D_N[44]		
AR7	FSB2D_N[45]		
AR10	FSB2D_N[46]		
AT9	FSB2D_N[47]		
AM3	FSB2D_N[48]		
AM6	FSB2D_N[49]		
AG12	FSB2D_N[5]		
AM7	FSB2D_N[50]		
AM1	FSB2D_N[51]		
AM4	FSB2D_N[52]		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 8 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AN2	FSB2D_N[53]		
AP1	FSB2D_N[54]		
AN3	FSB2D_N[55]		
AP4	FSB2D_N[56]		
AP5	FSB2D_N[57]		
AR4	FSB2D_N[58]		
AT5	FSB2D_N[59]		
AF9	FSB2D_N[6]		
AT6	FSB2D_N[60]		
AR3	FSB2D_N[61]		
AR1	FSB2D_N[62]		
AR6	FSB2D_N[63]		
AE11	FSB2D_N[7]		
AE10	FSB2D_N[8]		
AG9	FSB2D_N[9]		
AE14	FSB2DBI_N[0]		
AJ4	FSB2DBI_N[1]		
AN9	FSB2DBI_N[2]		
AP2	FSB2DBI_N[3]		
AE4	FSB2DBSY_N		
AD8	FSB2DEFER_N		
AF7	FSB2DP_N[0]		
AG6	FSB2DP_N[1]		
AH8	FSB2DP_N[2]		
AE8	FSB2DP_N[3]		
AE2	FSB2DRDY_N		
AC9	FSB2HIT_N		
AE7	FSB2HITM_N		
AE1	FSB2LOCK_N		
AG2	FSB2MCERR_N		
Y9	FSB2REQ_N[0]		
W11	FSB2REQ_N[1]		
V11	FSB2REQ_N[2]		
W8	FSB2REQ_N[3]		
W13	FSB2REQ_N[4]		
AH7	FSB2RESET_N		
AD5	FSB2RS_N[0]		
AC6	FSB2RS_N[1]		
AD6	FSB2RS_N[2]		
AG3	FSB2RSP_N		
AJ9	FSB2STBN_N[0]		
AJ6	FSB2STBN_N[1]		
AL10	FSB2STBN_N[2]		
AN6	FSB2STBN_N[3]		
AJ10	FSB2STBP_N[0]		
AJ7	FSB2STBP_N[1]		
AM10	FSB2STBP_N[2]		
AN5	FSB2STBP_N[3]		
AD3	FSB2TRDY_N		
AB10	FSB2VREF[0]		
AG8	FSB2VREF[2]		
AN8	FSB2VREF[4]		
AW8	FSB3A_N[10]		
AY1	FSB3A_N[11]		
AW5	FSB3A_N[12]		
AW6	FSB3A_N[13]		
AV9	FSB3A_N[14]		
AY2	FSB3A_N[15]		
AV6	FSB3A_N[16]		
AY8	FSB3A_N[17]		
AY4	FSB3A_N[18]		
AY5	FSB3A_N[19]		
BB8	FSB3A_N[20]		
BB6	FSB3A_N[21]		
BC5	FSB3A_N[22]		
BB5	FSB3A_N[23]		
BA3	FSB3A_N[24]		
BA6	FSB3A_N[25]		
BE6	FSB3A_N[26]		
BD7	FSB3A_N[27]		
BA4	FSB3A_N[28]		
BB3	FSB3A_N[29]		
AU7	FSB3A_N[3]		
BC4	FSB3A_N[30]		
BC8	FSB3A_N[31]		
BE9	FSB3A_N[32]		
BE8	FSB3A_N[33]		
BD6	FSB3A_N[34]		
BC7	FSB3A_N[35]		
AW3	FSB3A_N[36]		
AT8	FSB3A_N[37]		
BD9	FSB3A_N[38]		
AY7	FSB3A_N[39]		
AU5	FSB3A_N[4]		
AV3	FSB3A_N[5]		
AU2	FSB3A_N[6]		
AU1	FSB3A_N[7]		
AW2	FSB3A_N[8]		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 9 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AV4	FSB3A_N[9]		
BA12	FSB3ADS_N		
AV1	FSB3ADSTB_N[0]		
BA7	FSB3ADSTB_N[1]		
BE11	FSB3AP_N[0]		
BB12	FSB3AP_N[1]		
BC11	FSB3BINIT_N		
AW12	FSB3BNR_N		
BD12	FSB3BPM_N[4]		
AY13	FSB3BPM_N[5]		
AU10	FSB3BPRI_N		
AR12	FSB3BREQ0_N		
AP13	FSB3BREQ1_N		
AR13	FSB3D_N[0]		
AL14	FSB3D_N[1]		
AT17	FSB3D_N[10]		
AU17	FSB3D_N[11]		
AP17	FSB3D_N[12]		
AU16	FSB3D_N[13]		
AP16	FSB3D_N[14]		
AM16	FSB3D_N[15]		
AW14	FSB3D_N[16]		
AW15	FSB3D_N[17]		
BA15	FSB3D_N[18]		
AY14	FSB3D_N[19]		
AT15	FSB3D_N[2]		
BC14	FSB3D_N[20]		
BB14	FSB3D_N[21]		
BB15	FSB3D_N[22]		
BD15	FSB3D_N[23]		
BB17	FSB3D_N[24]		
BE17	FSB3D_N[25]		
BD16	FSB3D_N[26]		
BE15	FSB3D_N[27]		
BC16	FSB3D_N[28]		
AY17	FSB3D_N[29]		
AT14	FSB3D_N[3]		
BC17	FSB3D_N[30]		
AV16	FSB3D_N[31]		
AN18	FSB3D_N[32]		
AR18	FSB3D_N[33]		
AN17	FSB3D_N[34]		
AV19	FSB3D_N[35]		
AM18	FSB3D_N[36]		
AT18	FSB3D_N[37]		
AU20	FSB3D_N[38]		
AU19	FSB3D_N[39]		
AM13	FSB3D_N[4]		
AV21	FSB3D_N[40]		
AT20	FSB3D_N[41]		
AT21	FSB3D_N[42]		
AW21	FSB3D_N[43]		
AP20	FSB3D_N[44]		
AR21	FSB3D_N[45]		
AM19	FSB3D_N[46]		
AN20	FSB3D_N[47]		
BB18	FSB3D_N[48]		
AY19	FSB3D_N[49]		
AR16	FSB3D_N[5]		
AW18	FSB3D_N[50]		
BA19	FSB3D_N[51]		
BA18	FSB3D_N[52]		
BC19	FSB3D_N[53]		
BB20	FSB3D_N[54]		
BD19	FSB3D_N[55]		
BD21	FSB3D_N[56]		
BA21	FSB3D_N[57]		
BB21	FSB3D_N[58]		
AY20	FSB3D_N[59]		
AR15	FSB3D_N[6]		
AY22	FSB3D_N[60]		
BE21	FSB3D_N[61]		
BC20	FSB3D_N[62]		
BA22	FSB3D_N[63]		
AN14	FSB3D_N[7]		
AP14	FSB3D_N[8]		
AV15	FSB3D_N[9]		
AL13	FSB3DBI_N[0]		
AW17	FSB3DBI_N[1]		
AW20	FSB3DBI_N[2]		
BE20	FSB3DBI_N[3]		
BB11	FSB3DBSY_N		
AU11	FSB3DEFER_N		
BC13	FSB3DP_N[0]		
BD13	FSB3DP_N[1]		
BA13	FSB3DP_N[2]		
BE14	FSB3DP_N[3]		
BC10	FSB3DRDY_N		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 10 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AY10	FSB3DRVCRES		
AT11	FSB3HIT_N		
AU13	FSB3HITM_N		
BD10	FSB3LOCK_N		
BE12	FSB3MCERR_N		
BA9	FSB3ODTCRES		
AU4	FSB3REQ_N[0]		
AT3	FSB3REQ_N[1]		
AV7	FSB3REQ_N[2]		
AT2	FSB3REQ_N[3]		
AU8	FSB3REQ_N[4]		
AT12	FSB3RESET_N		
AY11	FSB3RS_N[0]		
AW11	FSB3RS_N[1]		
AV12	FSB3RS_N[2]		
AV13	FSB3RSP_N		
BB9	FSB3SLWCRES		
AN15	FSB3STBN_N[0]		
BA16	FSB3STBN_N[1]		
AP19	FSB3STBN_N[2]		
BD18	FSB3STBN_N[3]		
AM15	FSB3STBP_N[0]		
AY16	FSB3STBP_N[1]		
AR19	FSB3STBP_N[2]		
BE18	FSB3STBP_N[3]		
BA10	FSB3TRDY_N		
AV10	FSB3VREF[0]		
AU14	FSB3VREF[2]		
AV18	FSB3VREF[4]		
K5	FSBSLWCTRL		
BE24	FSBVCCA		
K4	INT_N[0]		
L3	INT_N[1]		
L1	INT_N[2]		
L4	INT_N[3]		
M3	INT_N[4]		
M2	INT_N[5]		
M5	INT_N[6]		
G1	PE0RN[0]		
H4	PE0RN[1]		
K8	PE0RN[2]		
L10	PE0RN[3]		
G2	PE0RP[0]		
H3	PE0RP[1]		

Ball No.	Signal Name	Buffer Type	Direction
K7	PE0RP[2]		
L9	PE0RP[3]		
G5	PE0TN[0]		
J6	PE0TN[1]		
L7	PE0TN[2]		
N11	PE0TN[3]		
G4	PE0TP[0]		
J5	PE0TP[1]		
L6	PE0TP[2]		
M11	PE0TP[3]		
C8	PE1RN[0]		
F12	PE1RN[1]		
K14	PE1RN[2]		
N14	PE1RN[3]		
C9	PE1RP[0]		
F11	PE1RP[1]		
J14	PE1RP[2]		
M14	PE1RP[3]		
E9	PE1TN[0]		
D7	PE1TN[1]		
H12	PE1TN[2]		
L13	PE1TN[3]		
E10	PE1TP[0]		
D8	PE1TP[1]		
H13	PE1TP[2]		
K13	PE1TP[3]		
G10	PE2RN[0]		
F8	PE2RN[1]		
E6	PE2RN[2]		
D4	PE2RN[3]		
G11	PE2RP[0]		
F9	PE2RP[1]		
E7	PE2RP[2]		
D5	PE2RP[3]		
J11	PE2TN[0]		
H9	PE2TN[1]		
G7	PE2TN[2]		
C5	PE2TN[3]		
J12	PE2TP[0]		
H10	PE2TP[1]		
G8	PE2TP[2]		
C6	PE2TP[3]		
F3	PE3RN[0]		
H7	PE3RN[1]		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 11 of 23)

Ball No.	Signal Name	Buffer Type	Direction
K11	PE3RN[2]		
P13	PE3RN[3]		
F2	PE3RP[0]		
H6	PE3RP[1]		
K10	PE3RP[2]		
N13	PE3RP[3]		
E4	PE3TN[0]		
F6	PE3TN[1]		
J9	PE3TN[2]		
M12	PE3TN[3]		
E3	PE3TP[0]		
F5	PE3TP[1]		
J8	PE3TP[2]		
L12	PE3TP[3]		
J20	PE4RN[0]		
F20	PE4RN[1]		
D19	PE4RN[2]		
A19	PE4RN[3]		
K20	PE4RP[0]		
G20	PE4RP[1]		
E19	PE4RP[2]		
A20	PE4RP[3]		
K19	PE4TN[0]		
G19	PE4TN[1]		
E18	PE4TN[2]		
C20	PE4TN[3]		
L19	PE4TP[0]		
H19	PE4TP[1]		
F18	PE4TP[2]		
D20	PE4TP[3]		
C17	PE5RN[0]		
B15	PE5RN[1]		
G17	PE5RN[2]		
K17	PE5RN[3]		
C18	PE5RP[0]		
B16	PE5RP[1]		
F17	PE5RP[2]		
J17	PE5RP[3]		
B18	PE5TN[0]		
A16	PE5TN[1]		
D16	PE5TN[2]		
J18	PE5TN[3]		
B19	PE5TP[0]		
A17	PE5TP[1]		
D17	PE5TP[2]		
H18	PE5TP[3]		
G16	PE6RN[0]		
E15	PE6RN[1]		
C14	PE6RN[2]		
B12	PE6RN[3]		
H16	PE6RP[0]		
E16	PE6RP[1]		
C15	PE6RP[2]		
B13	PE6RP[3]		
K16	PE6TN[0]		
F14	PE6TN[1]		
D13	PE6TN[2]		
A13	PE6TN[3]		
L16	PE6TP[0]		
F15	PE6TP[1]		
D14	PE6TP[2]		
A14	PE6TP[3]		
A10	PE7RN[0]		
E12	PE7RN[1]		
J15	PE7RN[2]		
N17	PE7RN[3]		
A11	PE7RP[0]		
E13	PE7RP[1]		
H15	PE7RP[2]		
M17	PE7RP[3]		
C11	PE7TN[0]		
B9	PE7TN[1]		
G13	PE7TN[2]		
M15	PE7TN[3]		
C12	PE7TP[0]		
B10	PE7TP[1]		
G14	PE7TP[2]		
L15	PE7TP[3]		
D10	PECLKN		
D11	PECLKP		
N20	PEICOMPI		
N19	PERCOMPO		
A7	PEVCCA		
B6	PEVCCBG		
A8	PEVSSA		
B7	PEVSSBG		
H1	PSEL[0]		
J3	PSEL[1]		

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 12 of 23)

Ball No.	Signal Name	Buffer Type	Direction
J2	PSEL[2]		
R6	PWRGOOD		
R9	RESET_N		
A23	RSVD		
D43	RSVD		
K1	RSVD		
K2	RSVD		
AW9	RSVD		
AY37	RSVD		
BC23	RSVD		
U12	SPD0SMBCLK		
V12	SPD0SMBDATA		
R11	SPD1SMBCLK		
T11	SPD1SMBDATA		
N10	SPD2SMBCLK		
P10	SPD2SMBDATA		
N8	SPD3SMBCLK		
M8	SPD3SMBDATA		
N2	TCK		
P3	TDI		
R3	TDO		
P12	TESTHI_V3REF		
R8	TESTHI_V3REF		
R12	TESTHI_V3REF		
N1	TMS		
R2	TRST_N		
P1	V3REF	Power	
T21	VCC	Power	
T23	VCC	Power	
T25	VCC	Power	
T27	VCC	Power	
U16	VCC	Power	
U20	VCC	Power	
U22	VCC	Power	
U24	VCC	Power	
U26	VCC	Power	
U28	VCC	Power	
V17	VCC	Power	
V19	VCC	Power	
V21	VCC	Power	
V23	VCC	Power	
V25	VCC	Power	
V27	VCC	Power	
W18	VCC	Power	
W20	VCC	Power	
W22	VCC	Power	
W24	VCC	Power	
W26	VCC	Power	
W28	VCC	Power	
Y19	VCC	Power	
Y21	VCC	Power	
Y23	VCC	Power	
Y25	VCC	Power	
Y27	VCC	Power	
AA18	VCC	Power	
AA20	VCC	Power	
AA22	VCC	Power	
AA24	VCC	Power	
AA26	VCC	Power	
AA28	VCC	Power	
AB19	VCC	Power	
AB21	VCC	Power	
AB23	VCC	Power	
AB25	VCC	Power	
AB27	VCC	Power	
AC18	VCC	Power	
AC20	VCC	Power	
AC22	VCC	Power	
AC24	VCC	Power	
AC26	VCC	Power	
AC28	VCC	Power	
AD19	VCC	Power	
AD21	VCC	Power	
AD23	VCC	Power	
AD25	VCC	Power	
AD27	VCC	Power	
AE18	VCC	Power	
AE20	VCC	Power	
AE22	VCC	Power	
AE24	VCC	Power	
AE26	VCC	Power	
AE28	VCC	Power	
AF19	VCC	Power	
AF21	VCC	Power	
AF23	VCC	Power	
AF25	VCC	Power	
AF27	VCC	Power	
AG18	VCC	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 13 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AG20	VCC	Power	
AG22	VCC	Power	
AG24	VCC	Power	
AG26	VCC	Power	
AG28	VCC	Power	
AH19	VCC	Power	
AH21	VCC	Power	
AH23	VCC	Power	
AH25	VCC	Power	
AH27	VCC	Power	
A24	VCCFBD	Power	
A41	VCCFBD	Power	
B29	VCCFBD	Power	
B41	VCCFBD	Power	
B42	VCCFBD	Power	
C22	VCCFBD	Power	
C34	VCCFBD	Power	
C43	VCCFBD	Power	
D27	VCCFBD	Power	
D39	VCCFBD	Power	
D44	VCCFBD	Power	
E32	VCCFBD	Power	
E44	VCCFBD	Power	
E45	VCCFBD	Power	
F25	VCCFBD	Power	
F37	VCCFBD	Power	
G30	VCCFBD	Power	
G42	VCCFBD	Power	
H23	VCCFBD	Power	
H35	VCCFBD	Power	
J28	VCCFBD	Power	
J40	VCCFBD	Power	
K21	VCCFBD	Power	
K33	VCCFBD	Power	
K45	VCCFBD	Power	
L26	VCCFBD	Power	
L38	VCCFBD	Power	
M31	VCCFBD	Power	
M43	VCCFBD	Power	
N24	VCCFBD	Power	
N36	VCCFBD	Power	
P24	VCCFBD	Power	
P28	VCCFBD	Power	
P29	VCCFBD	Power	
P41	VCCFBD	Power	
R22	VCCFBD	Power	
R24	VCCFBD	Power	
R26	VCCFBD	Power	
R29	VCCFBD	Power	
R30	VCCFBD	Power	
R34	VCCFBD	Power	
T32	VCCFBD	Power	
T39	VCCFBD	Power	
U32	VCCFBD	Power	
U44	VCCFBD	Power	
V37	VCCFBD	Power	
A5	VCCPE	Power	
A12	VCCPE	Power	
B4	VCCPE	Power	
B5	VCCPE	Power	
B17	VCCPE	Power	
C3	VCCPE	Power	
C10	VCCPE	Power	
D2	VCCPE	Power	
D3	VCCPE	Power	
D15	VCCPE	Power	
E1	VCCPE	Power	
E8	VCCPE	Power	
E20	VCCPE	Power	
F1	VCCPE	Power	
F13	VCCPE	Power	
G6	VCCPE	Power	
G18	VCCPE	Power	
H11	VCCPE	Power	
J4	VCCPE	Power	
J16	VCCPE	Power	
K9	VCCPE	Power	
L14	VCCPE	Power	
L18	VCCPE	Power	
M7	VCCPE	Power	
M18	VCCPE	Power	
M19	VCCPE	Power	
N12	VCCPE	Power	
N16	VCCPE	Power	
P15	VCCPE	Power	
P16	VCCPE	Power	
P17	VCCPE	Power	
P18	VCCPE	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 14 of 23)

Ball No.	Signal Name	Buffer Type	Direction
P19	VCCPE	Power	
R15	VCCPE	Power	
R18	VCCPE	Power	
T19	VCCPE	Power	
AF15	VCCSF	Power	
AG15	VCCSF	Power	
AH14	VCCSF	Power	
AH31	VCCSF	Power	
AH32	VCCSF	Power	
AJ15	VCCSF	Power	
AJ31	VCCSF	Power	
AJ33	VCCSF	Power	
AK14	VCCSF	Power	
AK15	VCCSF	Power	
AK32	VCCSF	Power	
AK33	VCCSF	Power	
AL21	VCCSF	Power	
AL22	VCCSF	Power	
AL23	VCCSF	Power	
AL24	VCCSF	Power	
AL25	VCCSF	Power	
AL31	VCCSF	Power	
AL32	VCCSF	Power	
AM21	VCCSF	Power	
AM22	VCCSF	Power	
AM23	VCCSF	Power	
AM24	VCCSF	Power	
AM25	VCCSF	Power	
AN22	VCCSF	Power	
AN23	VCCSF	Power	
AN24	VCCSF	Power	
AP22	VCCSF	Power	
AP23	VCCSF	Power	
AP24	VCCSF	Power	
AR22	VCCSF	Power	
AR23	VCCSF	Power	
AR24	VCCSF	Power	
AT22	VCCSF	Power	
AT23	VCCSF	Power	
AT24	VCCSF	Power	
AU23	VCCSF	Power	
AV22	VCCSF	Power	
AV24	VCCSF	Power	
U18	VCCSPL	Power	
U13	VPPSMBCLK	Power	
T13	VPPSMBDATA	Power	
A3	VSS	Power	
A4	VSS	Power	
A6	VSS	Power	
A9	VSS	Power	
A15	VSS	Power	
A18	VSS	Power	
A21	VSS	Power	
A27	VSS	Power	
A30	VSS	Power	
A33	VSS	Power	
A36	VSS	Power	
A39	VSS	Power	
A42	VSS	Power	
A43	VSS	Power	
B2	VSS	Power	
B3	VSS	Power	
B8	VSS	Power	
B11	VSS	Power	
B14	VSS	Power	
B20	VSS	Power	
B23	VSS	Power	
B26	VSS	Power	
B32	VSS	Power	
B33	VSS	Power	
B34	VSS	Power	
B35	VSS	Power	
B38	VSS	Power	
B43	VSS	Power	
B44	VSS	Power	
C1	VSS	Power	
C2	VSS	Power	
C4	VSS	Power	
C7	VSS	Power	
C13	VSS	Power	
C16	VSS	Power	
C19	VSS	Power	
C21	VSS	Power	
C23	VSS	Power	
C24	VSS	Power	
C25	VSS	Power	
C28	VSS	Power	
C31	VSS	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 15 of 23)

Ball No.	Signal Name	Buffer Type	Direction
C32	VSS	Power	
C33	VSS	Power	
C37	VSS	Power	
C40	VSS	Power	
C44	VSS	Power	
C45	VSS	Power	
D1	VSS	Power	
D6	VSS	Power	
D9	VSS	Power	
D12	VSS	Power	
D18	VSS	Power	
D21	VSS	Power	
D22	VSS	Power	
D23	VSS	Power	
D24	VSS	Power	
D30	VSS	Power	
D33	VSS	Power	
D36	VSS	Power	
D42	VSS	Power	
D45	VSS	Power	
E2	VSS	Power	
E5	VSS	Power	
E11	VSS	Power	
E14	VSS	Power	
E17	VSS	Power	
E23	VSS	Power	
E26	VSS	Power	
E29	VSS	Power	
E35	VSS	Power	
E38	VSS	Power	
E41	VSS	Power	
F4	VSS	Power	
F7	VSS	Power	
F10	VSS	Power	
F16	VSS	Power	
F19	VSS	Power	
F21	VSS	Power	
F22	VSS	Power	
F28	VSS	Power	
F31	VSS	Power	
F34	VSS	Power	
F40	VSS	Power	
F43	VSS	Power	
G3	VSS	Power	
G9	VSS	Power	
G12	VSS	Power	
G15	VSS	Power	
G21	VSS	Power	
G24	VSS	Power	
G27	VSS	Power	
G32	VSS	Power	
G33	VSS	Power	
G36	VSS	Power	
G39	VSS	Power	
G45	VSS	Power	
H2	VSS	Power	
H5	VSS	Power	
H8	VSS	Power	
H14	VSS	Power	
H17	VSS	Power	
H20	VSS	Power	
H25	VSS	Power	
H26	VSS	Power	
H27	VSS	Power	
H28	VSS	Power	
H29	VSS	Power	
H32	VSS	Power	
H38	VSS	Power	
H39	VSS	Power	
H40	VSS	Power	
H41	VSS	Power	
H44	VSS	Power	
J1	VSS	Power	
J7	VSS	Power	
J10	VSS	Power	
J13	VSS	Power	
J19	VSS	Power	
J21	VSS	Power	
J22	VSS	Power	
J25	VSS	Power	
J26	VSS	Power	
J27	VSS	Power	
J31	VSS	Power	
J32	VSS	Power	
J33	VSS	Power	
J34	VSS	Power	
J37	VSS	Power	
J38	VSS	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 16 of 23)

Ball No.	Signal Name	Buffer Type	Direction
J39	VSS	Power	
J43	VSS	Power	
K3	VSS	Power	
K6	VSS	Power	
K12	VSS	Power	
K15	VSS	Power	
K18	VSS	Power	
K24	VSS	Power	
K25	VSS	Power	
K26	VSS	Power	
K27	VSS	Power	
K30	VSS	Power	
K31	VSS	Power	
K32	VSS	Power	
K36	VSS	Power	
K37	VSS	Power	
K38	VSS	Power	
K39	VSS	Power	
K42	VSS	Power	
L2	VSS	Power	
L5	VSS	Power	
L8	VSS	Power	
L11	VSS	Power	
L17	VSS	Power	
L20	VSS	Power	
L23	VSS	Power	
L29	VSS	Power	
L32	VSS	Power	
L35	VSS	Power	
L36	VSS	Power	
L37	VSS	Power	
L41	VSS	Power	
L44	VSS	Power	
M1	VSS	Power	
M4	VSS	Power	
M6	VSS	Power	
M9	VSS	Power	
M10	VSS	Power	
M13	VSS	Power	
M16	VSS	Power	
M20	VSS	Power	
M22	VSS	Power	
M25	VSS	Power	
M26	VSS	Power	
M28	VSS	Power	
M34	VSS	Power	
M37	VSS	Power	
M40	VSS	Power	
N3	VSS	Power	
N6	VSS	Power	
N7	VSS	Power	
N9	VSS	Power	
N15	VSS	Power	
N18	VSS	Power	
N21	VSS	Power	
N27	VSS	Power	
N30	VSS	Power	
N33	VSS	Power	
N39	VSS	Power	
N42	VSS	Power	
N45	VSS	Power	
P2	VSS	Power	
P5	VSS	Power	
P8	VSS	Power	
P9	VSS	Power	
P11	VSS	Power	
P14	VSS	Power	
P20	VSS	Power	
P21	VSS	Power	
P22	VSS	Power	
P23	VSS	Power	
P25	VSS	Power	
P26	VSS	Power	
P27	VSS	Power	
P32	VSS	Power	
P35	VSS	Power	
P38	VSS	Power	
P44	VSS	Power	
R1	VSS	Power	
R4	VSS	Power	
R7	VSS	Power	
R10	VSS	Power	
R13	VSS	Power	
R14	VSS	Power	
R16	VSS	Power	
R17	VSS	Power	
R19	VSS	Power	
R20	VSS	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 17 of 23)

Ball No.	Signal Name	Buffer Type	Direction
R21	VSS	Power	
R23	VSS	Power	
R25	VSS	Power	
R27	VSS	Power	
R28	VSS	Power	
R31	VSS	Power	
R37	VSS	Power	
R38	VSS	Power	
R40	VSS	Power	
R43	VSS	Power	
T3	VSS	Power	
T6	VSS	Power	
T9	VSS	Power	
T12	VSS	Power	
T14	VSS	Power	
T15	VSS	Power	
T16	VSS	Power	
T17	VSS	Power	
T18	VSS	Power	
T20	VSS	Power	
T22	VSS	Power	
T24	VSS	Power	
T26	VSS	Power	
T28	VSS	Power	
T29	VSS	Power	
T30	VSS	Power	
T31	VSS	Power	
T33	VSS	Power	
T34	VSS	Power	
T35	VSS	Power	
T36	VSS	Power	
T37	VSS	Power	
T38	VSS	Power	
T42	VSS	Power	
T45	VSS	Power	
U2	VSS	Power	
U5	VSS	Power	
U8	VSS	Power	
U11	VSS	Power	
U14	VSS	Power	
U15	VSS	Power	
U17	VSS	Power	
U19	VSS	Power	
U21	VSS	Power	
U23	VSS	Power	
U25	VSS	Power	
U27	VSS	Power	
U29	VSS	Power	
U30	VSS	Power	
U31	VSS	Power	
U35	VSS	Power	
U36	VSS	Power	
U37	VSS	Power	
U38	VSS	Power	
U41	VSS	Power	
V1	VSS	Power	
V4	VSS	Power	
V7	VSS	Power	
V10	VSS	Power	
V13	VSS	Power	
V14	VSS	Power	
V15	VSS	Power	
V16	VSS	Power	
V18	VSS	Power	
V20	VSS	Power	
V22	VSS	Power	
V24	VSS	Power	
V26	VSS	Power	
V28	VSS	Power	
V31	VSS	Power	
V34	VSS	Power	
V36	VSS	Power	
V40	VSS	Power	
V43	VSS	Power	
W3	VSS	Power	
W9	VSS	Power	
W12	VSS	Power	
W19	VSS	Power	
W21	VSS	Power	
W23	VSS	Power	
W25	VSS	Power	
W27	VSS	Power	
W31	VSS	Power	
W33	VSS	Power	
W36	VSS	Power	
W39	VSS	Power	
W45	VSS	Power	
Y2	VSS	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 18 of 23)

Ball No.	Signal Name	Buffer Type	Direction
Y5	VSS	Power	
Y8	VSS	Power	
Y14	VSS	Power	
Y15	VSS	Power	
Y18	VSS	Power	
Y20	VSS	Power	
Y22	VSS	Power	
Y24	VSS	Power	
Y26	VSS	Power	
Y28	VSS	Power	
Y31	VSS	Power	
Y32	VSS	Power	
Y38	VSS	Power	
Y41	VSS	Power	
Y44	VSS	Power	
AA1	VSS	Power	
AA7	VSS	Power	
AA10	VSS	Power	
AA13	VSS	Power	
AA15	VSS	Power	
AA19	VSS	Power	
AA21	VSS	Power	
AA23	VSS	Power	
AA25	VSS	Power	
AA27	VSS	Power	
AA31	VSS	Power	
AA34	VSS	Power	
AA37	VSS	Power	
AA43	VSS	Power	
AB3	VSS	Power	
AB6	VSS	Power	
AB12	VSS	Power	
AB15	VSS	Power	
AB18	VSS	Power	
AB20	VSS	Power	
AB22	VSS	Power	
AB24	VSS	Power	
AB26	VSS	Power	
AB28	VSS	Power	
AB31	VSS	Power	
AB36	VSS	Power	
AB39	VSS	Power	
AB42	VSS	Power	
AB45	VSS	Power	
AC5	VSS	Power	
AC8	VSS	Power	
AC11	VSS	Power	
AC15	VSS	Power	
AC19	VSS	Power	
AC21	VSS	Power	
AC23	VSS	Power	
AC25	VSS	Power	
AC27	VSS	Power	
AC31	VSS	Power	
AC32	VSS	Power	
AC35	VSS	Power	
AC41	VSS	Power	
AD1	VSS	Power	
AD4	VSS	Power	
AD10	VSS	Power	
AD13	VSS	Power	
AD14	VSS	Power	
AD15	VSS	Power	
AD18	VSS	Power	
AD20	VSS	Power	
AD22	VSS	Power	
AD24	VSS	Power	
AD26	VSS	Power	
AD28	VSS	Power	
AD32	VSS	Power	
AD34	VSS	Power	
AD37	VSS	Power	
AD40	VSS	Power	
AD43	VSS	Power	
AE3	VSS	Power	
AE6	VSS	Power	
AE9	VSS	Power	
AE15	VSS	Power	
AE19	VSS	Power	
AE21	VSS	Power	
AE23	VSS	Power	
AE25	VSS	Power	
AE27	VSS	Power	
AE31	VSS	Power	
AE33	VSS	Power	
AE39	VSS	Power	
AE45	VSS	Power	
AF2	VSS	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 19 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AF8	VSS	Power	
AF11	VSS	Power	
AF14	VSS	Power	
AF18	VSS	Power	
AF20	VSS	Power	
AF22	VSS	Power	
AF24	VSS	Power	
AF26	VSS	Power	
AF28	VSS	Power	
AF31	VSS	Power	
AF32	VSS	Power	
AF35	VSS	Power	
AF38	VSS	Power	
AF41	VSS	Power	
AF44	VSS	Power	
AG1	VSS	Power	
AG4	VSS	Power	
AG7	VSS	Power	
AG13	VSS	Power	
AG19	VSS	Power	
AG21	VSS	Power	
AG23	VSS	Power	
AG25	VSS	Power	
AG27	VSS	Power	
AG31	VSS	Power	
AG37	VSS	Power	
AG43	VSS	Power	
AH6	VSS	Power	
AH9	VSS	Power	
AH12	VSS	Power	
AH18	VSS	Power	
AH20	VSS	Power	
AH22	VSS	Power	
AH24	VSS	Power	
AH26	VSS	Power	
AH28	VSS	Power	
AH33	VSS	Power	
AH36	VSS	Power	
AH39	VSS	Power	
AH42	VSS	Power	
AJ2	VSS	Power	
AJ5	VSS	Power	
AJ8	VSS	Power	
AJ11	VSS	Power	
AJ14	VSS	Power	
AJ32	VSS	Power	
AJ35	VSS	Power	
AJ41	VSS	Power	
AJ44	VSS	Power	
AK4	VSS	Power	
AK7	VSS	Power	
AK10	VSS	Power	
AK31	VSS	Power	
AK34	VSS	Power	
AK37	VSS	Power	
AK40	VSS	Power	
AL3	VSS	Power	
AL9	VSS	Power	
AL12	VSS	Power	
AL15	VSS	Power	
AL16	VSS	Power	
AL17	VSS	Power	
AL19	VSS	Power	
AL20	VSS	Power	
AL26	VSS	Power	
AL27	VSS	Power	
AL28	VSS	Power	
AL29	VSS	Power	
AL33	VSS	Power	
AL39	VSS	Power	
AL42	VSS	Power	
AL45	VSS	Power	
AM2	VSS	Power	
AM5	VSS	Power	
AM8	VSS	Power	
AM14	VSS	Power	
AM17	VSS	Power	
AM20	VSS	Power	
AM26	VSS	Power	
AM29	VSS	Power	
AM32	VSS	Power	
AM35	VSS	Power	
AM38	VSS	Power	
AM44	VSS	Power	
AN1	VSS	Power	
AN7	VSS	Power	
AN10	VSS	Power	
AN12	VSS	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 20 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AN13	VSS	Power	
AN19	VSS	Power	
AN21	VSS	Power	
AN25	VSS	Power	
AN31	VSS	Power	
AN37	VSS	Power	
AN40	VSS	Power	
AN43	VSS	Power	
AP3	VSS	Power	
AP6	VSS	Power	
AP12	VSS	Power	
AP15	VSS	Power	
AP18	VSS	Power	
AP27	VSS	Power	
AP30	VSS	Power	
AP33	VSS	Power	
AP36	VSS	Power	
AP42	VSS	Power	
AP45	VSS	Power	
AR5	VSS	Power	
AR8	VSS	Power	
AR11	VSS	Power	
AR17	VSS	Power	
AR20	VSS	Power	
AR29	VSS	Power	
AR35	VSS	Power	
AR38	VSS	Power	
AR41	VSS	Power	
AT1	VSS	Power	
AT4	VSS	Power	
AT10	VSS	Power	
AT13	VSS	Power	
AT16	VSS	Power	
AT25	VSS	Power	
AT28	VSS	Power	
AT31	VSS	Power	
AT34	VSS	Power	
AT40	VSS	Power	
AT43	VSS	Power	
AU3	VSS	Power	
AU6	VSS	Power	
AU9	VSS	Power	
AU15	VSS	Power	
AU18	VSS	Power	
AU21	VSS	Power	
AU22	VSS	Power	
AU24	VSS	Power	
AU27	VSS	Power	
AU33	VSS	Power	
AU36	VSS	Power	
AU39	VSS	Power	
AU45	VSS	Power	
AV2	VSS	Power	
AV8	VSS	Power	
AV11	VSS	Power	
AV14	VSS	Power	
AV20	VSS	Power	
AV23	VSS	Power	
AV26	VSS	Power	
AV29	VSS	Power	
AV32	VSS	Power	
AV38	VSS	Power	
AV41	VSS	Power	
AV44	VSS	Power	
AW1	VSS	Power	
AW4	VSS	Power	
AW7	VSS	Power	
AW13	VSS	Power	
AW16	VSS	Power	
AW19	VSS	Power	
AW22	VSS	Power	
AW23	VSS	Power	
AW24	VSS	Power	
AW25	VSS	Power	
AW31	VSS	Power	
AW34	VSS	Power	
AW37	VSS	Power	
AW43	VSS	Power	
AY6	VSS	Power	
AY9	VSS	Power	
AY12	VSS	Power	
AY18	VSS	Power	
AY21	VSS	Power	
AY23	VSS	Power	
AY24	VSS	Power	
AY27	VSS	Power	
AY30	VSS	Power	
AY36	VSS	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 21 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AY39	VSS	Power	
AY42	VSS	Power	
BA2	VSS	Power	
BA5	VSS	Power	
BA11	VSS	Power	
BA14	VSS	Power	
BA17	VSS	Power	
BA23	VSS	Power	
BA24	VSS	Power	
BA29	VSS	Power	
BA32	VSS	Power	
BA35	VSS	Power	
BA41	VSS	Power	
BA44	VSS	Power	
BB1	VSS	Power	
BB4	VSS	Power	
BB7	VSS	Power	
BB10	VSS	Power	
BB16	VSS	Power	
BB19	VSS	Power	
BB22	VSS	Power	
BB23	VSS	Power	
BB24	VSS	Power	
BB25	VSS	Power	
BB28	VSS	Power	
BB34	VSS	Power	
BB37	VSS	Power	
BB40	VSS	Power	
BB45	VSS	Power	
BC1	VSS	Power	
BC2	VSS	Power	
BC9	VSS	Power	
BC12	VSS	Power	
BC15	VSS	Power	
BC21	VSS	Power	
BC24	VSS	Power	
BC27	VSS	Power	
BC30	VSS	Power	
BC33	VSS	Power	
BC39	VSS	Power	
BC42	VSS	Power	
BC44	VSS	Power	
BC45	VSS	Power	
BD2	VSS	Power	
BD3	VSS	Power	
BD5	VSS	Power	
BD8	VSS	Power	
BD14	VSS	Power	
BD17	VSS	Power	
BD20	VSS	Power	
BD23	VSS	Power	
BD26	VSS	Power	
BD32	VSS	Power	
BD35	VSS	Power	
BD38	VSS	Power	
BD43	VSS	Power	
BD44	VSS	Power	
BE3	VSS	Power	
BE4	VSS	Power	
BE7	VSS	Power	
BE10	VSS	Power	
BE13	VSS	Power	
BE19	VSS	Power	
BE22	VSS	Power	
BE25	VSS	Power	
BE28	VSS	Power	
BE31	VSS	Power	
BE37	VSS	Power	
BE40	VSS	Power	
BE42	VSS	Power	
BE43	VSS	Power	
V29	VTT	Power	
V30	VTT	Power	
W6	VTT	Power	
W15	VTT	Power	
W16	VTT	Power	
W17	VTT	Power	
W29	VTT	Power	
W30	VTT	Power	
W42	VTT	Power	
Y11	VTT	Power	
Y16	VTT	Power	
Y17	VTT	Power	
Y29	VTT	Power	
Y30	VTT	Power	
Y35	VTT	Power	
AA4	VTT	Power	
AA16	VTT	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 22 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AA17	VTT	Power	
AA29	VTT	Power	
AA30	VTT	Power	
AA40	VTT	Power	
AB9	VTT	Power	
AB16	VTT	Power	
AB17	VTT	Power	
AB29	VTT	Power	
AB30	VTT	Power	
AB33	VTT	Power	
AC2	VTT	Power	
AC14	VTT	Power	
AC16	VTT	Power	
AC17	VTT	Power	
AC29	VTT	Power	
AC30	VTT	Power	
AC38	VTT	Power	
AC44	VTT	Power	
AD7	VTT	Power	
AD16	VTT	Power	
AD17	VTT	Power	
AD29	VTT	Power	
AD30	VTT	Power	
AD31	VTT	Power	
AE12	VTT	Power	
AE16	VTT	Power	
AE17	VTT	Power	
AE29	VTT	Power	
AE30	VTT	Power	
AE36	VTT	Power	
AE42	VTT	Power	
AF5	VTT	Power	
AF16	VTT	Power	
AF17	VTT	Power	
AF29	VTT	Power	
AF30	VTT	Power	
AG10	VTT	Power	
AG16	VTT	Power	
AG17	VTT	Power	
AG29	VTT	Power	
AG30	VTT	Power	
AG34	VTT	Power	
AG40	VTT	Power	
AH3	VTT	Power	
AH15	VTT	Power	
AH16	VTT	Power	
AH17	VTT	Power	
AH29	VTT	Power	
AH30	VTT	Power	
AH45	VTT	Power	
AJ16	VTT	Power	
AJ17	VTT	Power	
AJ18	VTT	Power	
AJ19	VTT	Power	
AJ20	VTT	Power	
AJ21	VTT	Power	
AJ22	VTT	Power	
AJ23	VTT	Power	
AJ24	VTT	Power	
AJ25	VTT	Power	
AJ26	VTT	Power	
AJ27	VTT	Power	
AJ28	VTT	Power	
AJ29	VTT	Power	
AJ30	VTT	Power	
AJ38	VTT	Power	
AK1	VTT	Power	
AK13	VTT	Power	
AK16	VTT	Power	
AK17	VTT	Power	
AK18	VTT	Power	
AK19	VTT	Power	
AK20	VTT	Power	
AK21	VTT	Power	
AK22	VTT	Power	
AK23	VTT	Power	
AK24	VTT	Power	
AK25	VTT	Power	
AK26	VTT	Power	
AK27	VTT	Power	
AK28	VTT	Power	
AK29	VTT	Power	
AK30	VTT	Power	
AK43	VTT	Power	
AL6	VTT	Power	
AL18	VTT	Power	
AL30	VTT	Power	
AL36	VTT	Power	

Table 5-2. Intel® 7300 Chipset MCH Signals By Signal Name (Sheet 23 of 23)

Ball No.	Signal Name	Buffer Type	Direction
AM11	VTT	Power	
AM41	VTT	Power	
AN4	VTT	Power	
AN16	VTT	Power	
AN28	VTT	Power	
AN34	VTT	Power	
AP9	VTT	Power	
AP21	VTT	Power	
AP39	VTT	Power	
AR2	VTT	Power	
AR14	VTT	Power	
AR26	VTT	Power	
AR32	VTT	Power	
AR44	VTT	Power	
AT7	VTT	Power	
AT19	VTT	Power	
AT37	VTT	Power	
AU12	VTT	Power	
AU30	VTT	Power	
AU42	VTT	Power	
AV5	VTT	Power	
AV17	VTT	Power	
AV35	VTT	Power	
AW10	VTT	Power	
AW28	VTT	Power	
AW40	VTT	Power	
AY3	VTT	Power	
AY15	VTT	Power	
AY33	VTT	Power	
AY45	VTT	Power	
BA1	VTT	Power	
BA8	VTT	Power	
BA20	VTT	Power	
BA26	VTT	Power	
BA38	VTT	Power	
BA45	VTT	Power	
BB2	VTT	Power	
BB13	VTT	Power	
BB31	VTT	Power	
BB43	VTT	Power	
BB44	VTT	Power	
BC3	VTT	Power	
BC6	VTT	Power	
BC18	VTT	Power	
BC36	VTT	Power	
BC43	VTT	Power	
BD4	VTT	Power	
BD11	VTT	Power	
BD29	VTT	Power	
BD41	VTT	Power	
BD42	VTT	Power	
BE5	VTT	Power	
BE16	VTT	Power	
BE34	VTT	Power	
BE41	VTT	Power	
U1	XDPCOMCRES		
T5	XDPD_N[0]		
V2	XDPD_N[1]		
U7	XDPD_N[10]		
T7	XDPD_N[11]		
R5	XDPD_N[12]		
T10	XDPD_N[13]		
T1	XDPD_N[14]		
T2	XDPD_N[15]		
V3	XDPD_N[2]		
V9	XDPD_N[3]		
V6	XDPD_N[4]		
V8	XDPD_N[5]		
U3	XDPD_N[6]		
U10	XDPD_N[7]		
U9	XDPD_N[8]		
U6	XDPD_N[9]		
T4	XDPDSTBN_N		
U4	XDPDSTBP_N		
W1	XDPODTCRES		
T8	XDPRDY_N		
W2	XDPSLWCRES		

§

6 System Address Map

This chapter describes the system address maps in memory space, I/O space, and PCI configuration space.

6.1 Memory Map

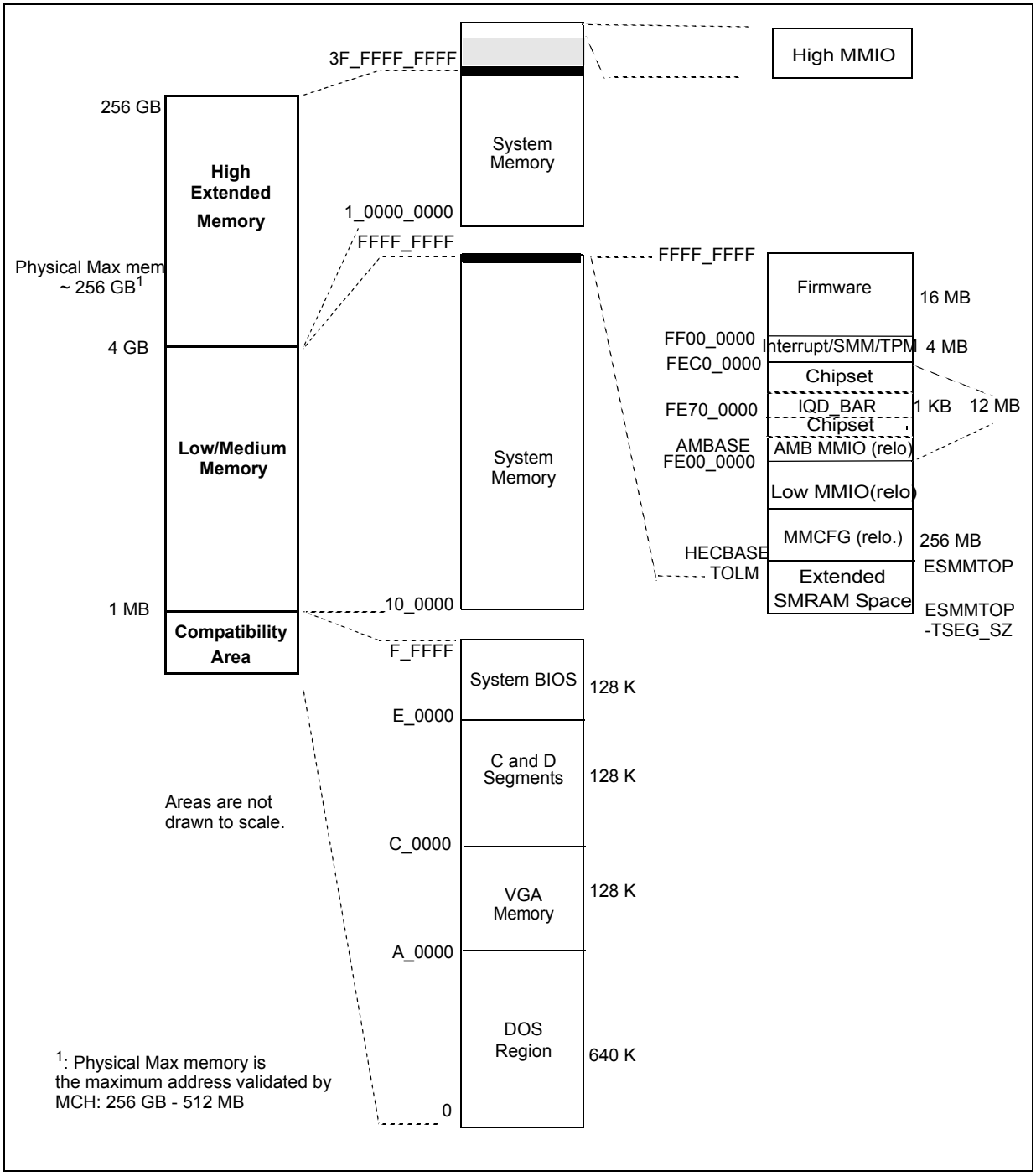
The platform supports 40 bits of memory address space. All key components and interfaces also support the 40 bit address space.

- 39-bit local address supported over the FBD channels for the memory space.
- 32 and 64 bit address bit formats supported for PCI Express interface.

The chipset treats accesses to various address ranges in different ways. There are fixed ranges like the compatibility region below 1 MB, interrupt delivery range, and the system region located in the 32 MB directly below 4 GB. In addition, there is a variable region for Memory Mapped I/O. The locations of these ranges in the memory map are illustrated in Figure 6-1.

The system address space is divided into non-overlapped regions by system software. All regions should not overlap with other region, otherwise, it is a programming error and Intel® 7300 Chipset's behavior is undefined. For example, setting of HECBASE to all zeros will overlap MMCFG region and Compatibility region.

Figure 6-1. System Memory Address Space



6.1.1 Compatibility Region

This is the range from 0-1 MB (0_0000 to F_FFFF). Requests to the Compatibility region are directed to main memory, the Compatibility Bus (ESI), or the VGA device. Any physical DRAM that would be addressed by requests in this region that are mapped

to the Compatibility Bus (ESI) is not recovered. This region is divided into four ranges. Regions below 1M that are mapped to memory are accessible by the processors and by any I/O bus.

Note: The DRAM that has a physical address between 0-1 MB must not be recovered or relocated or reflected. This range must always be available to the OS as DRAM, even if at times addresses in this range are sent to the Compatibility Bus (ESI) or VGA or other non-DRAM areas.

6.1.1.1 DOS Region

DOS applications execute in the lowest 640 KB, in the address range 0h to 9_FFFFh. This range is always mapped to main memory.

6.1.1.2 VGA Memory Range

The 128 KB Video Graphics Adapter Memory range (A_0000h to B_FFFFh) can be mapped to the VGA device which may be on any PCI Express or ESI or it can be mapped to main memory (if it is mapped to SMM space). At power-on this space is mapped to the ESI port.

Note that the Intel® 7300 Chipsett does not support the historical MDA (monochrome display adapter) space which would have appeared in the VGA region.

This region can be redirected by BIOS to point to any bus which has a VGA card. If the VGAEN bit is set in one of the BCTRL configuration registers associated with the PCI Express port, then transactions in this space are sent to that PCI Express port. Note that the VGAEN bit can only be set in one and only one of the BCTRL registers. For more information on the BCTRL registers, please refer to Section 4.8.10.28, “BCTRL[7:1]: Bridge Control Register”.

If the VGAEN bit of a PCI Express port y in the Intel® 7300 Chipset is set and BCTRL[y].VGA16bdecode is set to zero, ISAEN bits of all peer PCI Express ports with valid I/O range (PEXCMD.IOAE = 1, IOLIMIT >= IOBASE) in the MCH must be set by software. Otherwise, it is a programming error due to the routing conflict.

If the VGAEN bit of a PCI Express port y in the MCH is set and BCTRL[y].VGA16bdecode is set to one and if there is another PCI Express port x (y != x) with valid I/O range including the lowest 4K I/O addresses (PEXCMD[x].IOAE = 1, IOLIMIT[x] >= IOBASE[x] = 0000h), BCTRL[x].ISAEN bit must be set to 1 by software. Otherwise, it is a programming error.

This 128 KB region may also be used for mapping SMM space. The SMM range can overlay the VGA range in the A and B segments. If the SMM range overlaps an enabled VGA range then the state of the SMMEM# signal determines where accesses to the SMM Range are directed. SMMEM# asserted directs the accesses to the memory and SMMEM# deasserted directs the access to the PCI Express bus where VGA has been mapped.

This region cannot be cached. It is a programming error if it is cached.

6.1.1.3 C and D Segments

Writes and reads may be directed to different destinations in the range C_0000 to D_FFFF. Typically, these blocks were used to shadow ISA device BIOS code. For the MCH, these regions are used to provide address space to PCI devices requiring memory

space below 1 MB. The range is divided into 8 sub-ranges. These ranges are defined by PAM registers. There is a PAM register for each sub-range that defines the routing of reads and writes.

Table 6-1. PAM Settings

PAM [5:4]/[1:0]	writes go to	reads go to	Result
00	ESI	ESI	Mapped to the ESI
01	ESI	Main Memory	Memory Write Protect
10	Main Memory	ESI	In-line Shadowed
11	Main Memory	Main Memory	Mapped to main memory

The power-on default for these segments is mapped read/write to the ESI (Intel® 631xESB/632xESB I/O Controller Hub). Software should not set cacheable memory attributes for any of these ranges, unless both reads and writes are mapped to main memory. Chipset functionality is not guaranteed if this region is cached in any mode other than both reads and writes being mapped to main memory.

For locks to this region, the MCH will complete, but does not guarantee the atomicity of locked access to this range when writes and reads are mapped to separate destinations. If a PAM region is configured as "write only" and a locked transaction is targeted by the CPU to system memory, then the MCH will aborts the entire lock sequence since MemRdLk will not be allowed due to the "write only" restriction. It is important for Software/BIOS to adhere to these restrictions before using the PAM region.

If inbound accesses are expected, the C and D segments MUST be programmed to send accesses to DRAM.

6.1.1.4 System BIOS (E and F Segments)

The 128 KB region from E0000h to F_FFFFh is divided into 5 sub-ranges. Read/write attributes defined in the PAM registers may be used to direct accesses to the ESI or main memory. The PAM settings for accesses in this region are defined in Table 6-1, "PAM Settings".

The power-on default for these segments is mapped read/write to the ESI (Intel® 631xESB/632xESB I/O Controller Hub). Software should not set cacheable memory attributes for any of these ranges, unless both reads and writes are mapped to main memory. Chipset functionality is not guaranteed if this region is cached in any mode other than both reads and writes being mapped to main memory.

For locks to this region, the MCH will complete, but does not guarantee the atomicity of locked access to this range when writes and reads are mapped to separate destinations.

If inbound accesses are expected, the E and F segments MUST be programmed to send accesses to DRAM.

6.1.2 Low/Medium Memory

The low/medium memory regions range from 1MB to 4GB. It consists of sub-regions for Firmware, Processor memory mapped functions, and MCH specific registers.

6.1.2.1 System Memory

See Section 6.1.4, "Main Memory Region".

6.1.2.2 ISA Hole (Not Applicable)

6.1.2.3 Extended SMRAM Space (TSEG)

SMM space allows system management software to partition a region in main memory that is only accessible by system management software. When the SMM range is enabled, memory in this range is not exposed to the Operating System. The SMM range is accessed only when the processor is in SMM-mode. This is an extended SMM range that is different than the SMM space that may overlap VGA space. The register fields that define this extended SMM range are the EXSMRC.TSEG_SZ and EXSMRTOP.ESMMTOP. The TSEG SMM space starts at ESMMTOP - TSEG_SZ and ends at ESMMTOP. This region may be 512KB, 1MB, 2MB, or 4MB in size, depending on the TSEG_SZ field. ESMMTOP is relocatable to accommodate software that wishes to configure the TSEG SMM space before MMIO space is known. The ESMMTOP will default to the same default value as TOLM (Top Of Low Memory, defined by TOLM register) for Cayuse compatibility. It is a programming error if TOLM is programmed less than ESMMTOP.

If SMM is enabled, the chipset allows accesses to this range only when the SMMEM# signal on the processor bus is asserted with the request.

If SMMEM# is deasserted, accesses to the SMM Range are master aborted. If SMMEM# is asserted the access is routed to main memory. The MCH will use the SMM enable and range registers to determine where to route the access.

The Intel® 7300 Chipset will not support a locked access that crosses an SMM boundary. Firmware should not create data structures that span this boundary.

SMM main memory is protected from Inbound accesses.

Note: In order to make cacheable SMM possible, the chipset must accept EWB's (BWL's) and must absorb IWB (HITM) data regardless of the condition of the SMMEM# pin. The MCH will not set the error bit EXSMRAMC.E_SMERR in this case. Because of this, care must be used when attempting to cache SMM space. The chipset/platform **cannot protect** against processors that attempt to illegally access SMM space that is modified in another processor's cache. Any software that creates such a condition (for example, by corrupting the page table) will jeopardize the protective properties of SMM.

Note: The MCH does not support locks to SMM region in MMIO space (uncacheable) that elicit an IWB response from the Processor. Technically, such locked MMIO based SMM accesses should not be treated as cacheable by the symmetric agent and is considered a programming error. If such a scenario arises, the MCH will not guarantee consistency and the system can potentially hang.

See Table 6-9, "Decoding Processor Requests to SMM and VGA Spaces" for details on SMM decoding.

6.1.2.4 Memory Mapped Configuration (MMCFG) Region

There is one relocatable memory mapped configuration region in the MCH. The processor bus address defines the configuration register to be accessed and the processor bus data either returns or provides register contents. As opposed to CF8/CFC based configuration accesses, this mechanism is atomic.The memory mapped configuration region is compatible with the PCI Express enhanced configuration mechanism. It is a 256MB window that maps to PCI Express registers. (Both in the

chipset and south of the chipset.) The location of this MMCFG window is defined by the HECBASE register. The HECBASE register could also be accessed through a fixed location.

The default value of HECBASE maps this region such that there will be no wasted memory that is lost behind it. The default value for the PCI Express registers is the same as the default value of TOLM. If this range is moved, the following recommendations will enable reclaiming the memory that is lost to MMCFG accesses.

1. MMCFG range is mapped to a legal location within the range between TOLM and 4GB. Since ranges must not overlap other legal ranges, it is safest to put this range between TOLM and the lowest real MMIO range. (The current default is in these ranges) OR

2. Put the region above 4GB (above the top of memory and not overlapping above 4GB MMIO space.). BIOS/software must ensure there are no outstanding configuration accesses or memory accesses to the old and new MMCFG range addresses when relocating this range.

Note: An SMM program can address up to 4GB of memory. SMM is similar to real-address mode in that there are no privilege levels or address mapping. The MCH allows the relocation of HECBASE above 4GB. However, SMM code cannot access extended configuration space if HECBASE is relocated above 4GB. This is a CPU limitation and has not been functionally validated. Page Size Extension (PSE) is supported in SMM but Page Address Extension (PAE) is currently not in P4 families.

For more information on the memory mapped configuration mechanism described here, please see the Configuration Map and Access Chapter. For more information on any of the MCH registers, such as HECBASE.

6.1.2.5 Low Memory Mapped I/O (MMIO)

This is the first of two MCH memory mapped I/O ranges. The low memory mapped I/O range is defined to be between TOLM and FE00_0000. This low MMIO region is further subdivided between the PCI Express and ESI (Intel® 631xESB/632xESB I/O Controller Hub) ports. The following table shows the registers used to define the MMIO ranges for each PCI Express/ESI device. These registers are compatible with PCI Express and the PCI to PCI bridge specifications. Note that all subranges must be contained in the low memory mapped I/O range (between TOLM and FE00_0000). In other words, the lowest base address must be above TOLM and the highest LIMIT register must be below FE00_0000. Subranges must also not overlap each other.

Table 6-2. Low Memory Mapped I/O[a] (Sheet 1 of 2)

I/O Port	Intel® 7300 Chipset Base	Intel® 7300 Chipset Limit
ESI (Intel® 631xESB/632xESB I/O Controller Hub)	N/A[b]	
PEX1 Memory	MBASE1	MLIMIT1
PEX1 Prefetchable Memory	PMBASE1	PMLIMIT1
PEX2 Memory	MBASE2	MLIMIT2
PEX2 Prefetchable Memory	PMBASE2	PMLIMIT2
PEX3 Memory	MBASE3	MLIMIT3
PEX3 Prefetchable Memory	PMBASE3	PMLIMIT3
PEX4 Memory	MBASE4	MLIMIT4
PEX4 Prefetchable Memory	PMBASE4	PMLIMIT4

Table 6-2. Low Memory Mapped I/O[a] (Sheet 2 of 2)

I/O Port	Intel® 7300 Chipset Base	Intel® 7300 Chipset Limit
PEX5 Memory	MBASE5	MLIMIT5
PEX5 Prefetchable Memory	PMBASE5	PMLIMIT5
PEX6 Memory	MBASE6	MLIMIT6
PEX6 Prefetchable Memory	PMBASE6	PMLIMIT6
PEX7 Memory	MBASE7	MLIMIT7
PEX7 Prefetchable Memory	PMBASE7	PMLIMIT7

Notes:

a. This table assumes PMLU and PMBU are 0's. Otherwise, the prefetchable memory space will be located in high MMIO space.

b. The MCH does not need base/limit for Intel® 631xESB/632xESB I/O Controller Hub because subtractive decoding will send the accesses to the Intel® 631xESB/632xESB I/O Controller Hub. This is OK for software also, since the Intel® 631xESB/632xESB I/O Controller Hub is considered part of the same bus as the MCH.

The Intel® 7300 Chipset will decode addresses in this range and route them to the appropriate ESI or PCI Express port. If the address is in the low MMIO range, but is not contained in any of the PCI Express base and limit ranges, it will be routed to the ESI.

If the PMLU and PMBU registers are greater than 0, then the corresponding prefetchable region will be located in the high MMIO range instead.

6.1.2.6 Chipset Specific Range

The address range FE00_0000 - FEBF_FFFF region is reserved for chipset specific functions.

- FE00_0000 - FE01_FFFF: This range (with size of 128 KB for four FBD channels, 16 AMB per channel, 2KB per AMB) is used for accessing AMB registers. These AMB registers can only be accessed thru memory mapped register access mechanism as MMIO. Notice that they are not accessible thru CF8/CFC or MMCFG which are for PCI/PCI Express configuration space registers. This range could be relocated by programming AMBASE register. The AMBASE register could also be accessed through a fixed location.
- FE60_0000 - FE6F_FFFF: This range is used for fixed memory mapped MCH registers. They are accessible only from the processor bus. These registers are fixed since they are needed early during the boot process. The registers include:
  a. 4 Scratch Pad Registers
  b. 4 Sticky Scratch Pad Registers
  c. 4 Boot flag registers
  d. HECBASE register for MMCFG
  e. AMBASE register for AMB memory mapped registers

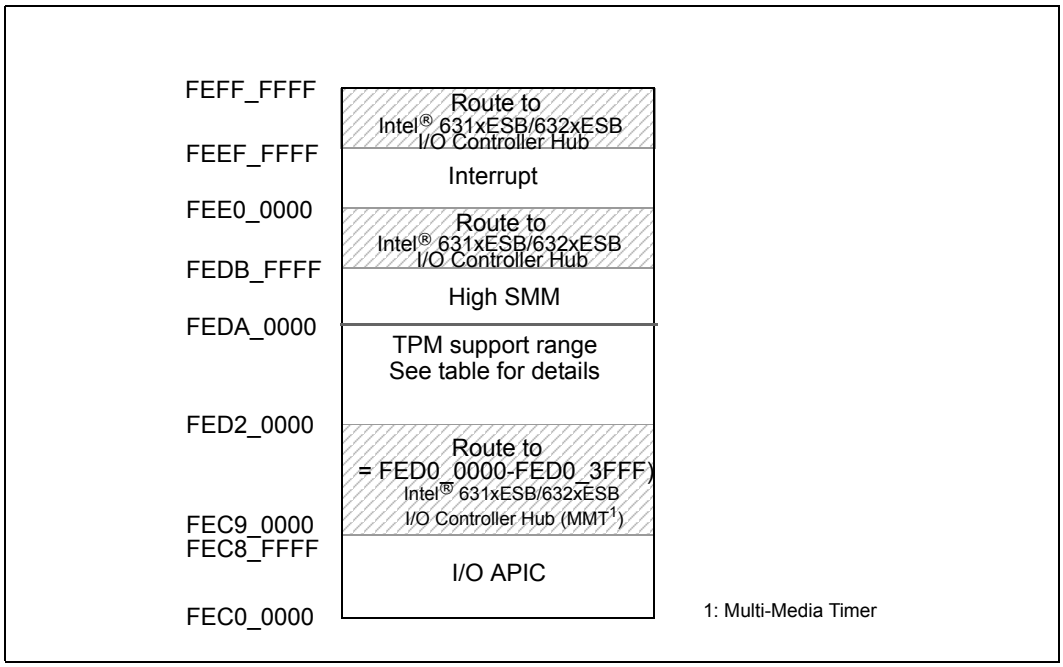
  These registers are described in the MCH Configuration Register Chapter.
- FE70_0000 - FE70_03FF: IQD_BAR MMIO. The integrated DMA device has a 1 KB MMIO space with a default range from FE70_0000 to FE70_03FF. This range could be relocated by programming IQD_BAR register. This range could be used as a private MMIO space by software.

The Intel® 7300 Chipset will master abort requests to the remainder and undefined locations of this region.

6.1.2.7 Interrupt Region

This 4 MB range is used for processor specific applications. It lies between FEC0_0000h and FEFF_FFFFh.

Figure 6-2. Interrupt/High SMM Region



6.1.2.7.1 I/O APIC Controller Range

This address range FEC0_0000 to FEC8_FFFF is used to communicate with the IOAPIC controllers in the PXH, IXH, Intel® 631xESB/632xESB I/O Controller Hub.

The APIC ranges are hard coded. Reads and writes to each IOAPIC region should be sent to the appropriate ESI or PCI Express as indicated below.

Table 6-3. I/O APIC Address Mapping

IOAPIC0 (ESI)	0_FEC0_0000 to 0_FEC7_FFFF[a]
IOAPIC1 (PEX1)	0_FEC8_0000 to 0_FEC8_0FFF
IOAPIC2 (PEX2)	0_FEC8_1000 to 0_FEC8_1FFF
IOAPIC3 (PEX3)	0_FEC8_2000 to 0_FEC8_2FFF
IOAPIC4 (PEX4)	0_FEC8_3000 to 0_FEC8_3FFF
IOAPIC5 (PEX5)	0_FEC8_4000 to 0_FEC8_4FFF
IOAPIC6 (PEX6)	0_FEC8_5000 to 0_FEC8_5FFF
IOAPIC7 (PEX7)	0_FEC8_6000 to 0_FEC8_6FFF
Reserved (Intel® 631xESB/632xESB I/O Controller Hub for master abort)	0_FEC8_7000 to 0_FEC8_FFFF

Notes:

a. IOAPIC range is hard coded on the Intel® 631xESB/632xESB I/O Controller Hub

For hot plug I/O APIC support, it is recommended that software use the standard MMIO range to communicate with the PXH or IXH. To accomplish this, the PXH.MBAR and/or IXH.XAPIC_BASE_ADDRESS_REG must be programmed within the PCI Express device MMIO region.

Inbound accesses to this memory range should also be routed to the I/O APIC controllers. This could happen if software configures MSI devices to send MSI's to an I/O APIC controller.

6.1.2.7.2 TPM Support Range

Intel® 7300 Chipset supports Trusted Platform Module (TPM) 1.2 locality 0 only, i.e. FED4_0xxx memory mapped accesses. TPM sits on the LPC bus of the Intel® 631xESB/632xESB I/O Controller Hub. TPM access from the processor (without LOCK#) is supported regardless of whether VT-d is enabled. Processor issued memory mapped access to FED4_0xxx will be converted to ESI LT_READ/ESI LT_WRITE for TPM accesses by the MCH. Intel® 631xESB/632xESB I/O Controller Hub will convert LT_READ/LT_WRITE to LPC TPM_Read and LPC TPM_Write. The MCH access control for the range to support TPM is defined in Table 6-4, "TPM support range address routing" on page 339.

Table 6-4. TPM support range address routing

Address range	FSB Access	Inbound Access
FED2_0000 - FED3_FFFF	Normal route to ESI	Normal route to ESI
FED4_0000 - FED4_0FFF	LT_Mem_Rd/Wr to ESI if LOCK# deasserted; Lock aborted in the MCH if LOCK# asserted.	Normal route to ESI
FED4_1000 - FED9_FFFF	Normal route to ESI	Normal route to ESI

All inbound accesses from external/internal I/O Devices or SMBus to FED4_0000 thru FED4_0FFF will be routed to ESI.

Locked transaction to the FED4_0xxx from FSB will be master aborted. Note: There will be no LT support in TNB although LT transaction to ESI is used internally to communicate TPM access.

6.1.2.7.3 High SMM Range

If high SMM space is enabled by EXSMRC.H_SMRAME, then requests to the address range from FEDA_0000 to FEDB_FFFF will be aliased down to the physical address of A_0000 to B_FFFF. The HIGHSMM space allows cacheable accesses to the compatible (legacy) SMM space and Intel® 7300 Chipset will track its caching state. In this range, the chipset will accept EWB's (BWL's) regardless of the SMMEM# pin. Also, if there is an implicit writeback (HITM with data), the chipset will update memory with the new data (regardless of the SMMEM# pin). This is the same behavior as the Cayuse chipset. Note that if the HIGHSMM space is enabled, the aliased SMM space of A_0000-B_FFFF will be disabled.

Note: In order to make cacheable SMM possible, the chipset must accept EWB's (BWL's) and must absorb IWB (HITM) data regardless of the condition of the SMMEM# pin. Because of this, care must be used when attempting to cache SMM space. The chipset/platform **cannot protect** against processors who attempt to illegally access SMM space that is modified in another processor's cache. Any software that creates such a condition (for example, by corrupting the page table) will jeopardize the protective properties of SMM.

6.1.2.7.4 Interrupt Range

Requests to the address range FEE0_0000 to FEEF_FFFF are used to deliver interrupts. Memory reads or write transactions to this range are illegal from the processor. The processor issues interrupt transactions to this range. Inbound interrupts from the PCI Express devices in the form of memory writes to this range are converted by the Intel® 7300 Chipset to processor bus interrupt requests. See the Interrupt Chapter for details.

6.1.2.7.5 Reserved Ranges

The MCH will master abort requests to the remaining addresses in the interrupt range (FEC0_0000 - FEFF_FFFF) which is not specified. This can be done by sending the request to the compatibility bus (ESI) to be master aborted. In the event the Intel® 631xESB/632xESB I/O Controller Hub adds new registers or functionality in this range in the future, this provides the most likely chance that it will work.

6.1.2.8 Firmware Range

The MCH allocates 16MB of firmware space from FF00_0000 to FFFF_FFFF. Requests in this range are directed to the Compatibility Bus (ESI). The Intel® 631xESB/632xESB I/O Controller Hub will route these to its FWH interface. This range is only accessible from any processor bus. No peer to peer access is allowed for FWH. See Table 6-10, "Address Disposition for Inbound Transactions" on page 347

6.1.3 High Extended Memory

This is the range above 4GB. The range from 4GB to MIR[2].LIMIT is mapped to system memory. There can also be a memory mapped I/O region that is located at the top of the address space. (Just below 1TB). This is described in Section 6.1.3.2.

6.1.3.1 System Memory

See Section 6.1.4, "Main Memory Region".

6.1.3.2 High MMIO

The high memory mapped I/O region is located above the top of memory as defined by MIR[2].LIMIT. These PMBU and PMLU registers in each PCI Express configuration device determine whether there is memory mapped I/O space above the top of memory. If an access is above MIR[2].LIMIT and it falls within the PMBU+PMBASE and PMLU+PMLIMIT range, it should be routed to the appropriate PCI Express port. For accesses above MIR[2].LIMIT (and above 4 GB) that are not in a high MMIO region, they should be master aborted.

6.1.3.3 Extended Memory (not applicable)

The range of memory just below 4 GB from TOLM to 4 GB (Low MMIO, Chipset, Interrupt) does not map to memory. If nothing is done, the DRAM memory behind the TOLM to 4 GB range will be unused.

6.1.4 Main Memory Region

6.1.4.1 Application of Coherency Protocol

The Intel® 7300 Chipset applies the coherency protocol to any accesses to main memory. Table 6-6 defines the conditions under which processor transactions are routed to main memory. Table 6-10 defines the conditions under which inbound transactions are routed to main memory. Application of the coherency protocol includes snooping the other processor bus.

Two exceptions are the C_0000 - F_FFFF and the legacy SMM ranges. (C_0000-F_FFFF may not necessarily route both reads and writes to memory, legacy SMM range may target non-memory when not in SMM mode.) These exceptions will not apply the coherency protocol. The MCH may malfunction if processors issue coherent

transactions such as BRILs, IWBs, and EWBs to ranges that do not call for application of the coherency protocol to both reads and writes. In addition, the chipset can't guarantee coherency for these ranges regardless of the types of transactions that are issued. Software must not set cacheable attributes for these areas (Only UC and WC attributes are not cacheable).

6.1.4.2 Routing Memory Requests

When a request appears on the processor bus, ESI port, or PCI Express link, and it does not fall in any of the previously mentioned regions, it is compared against the MIR.LIMIT registers on the Intel® 7300 Chipset.

The MIR.LIMIT registers will decode an access into a specific interleaving range. Within the interleaving range, the MIR.LIMIT register indicates which FBD memory branch the address is associated with. In the event that a mirroring event is occurring, memory writes are associated with two branches of FBDs.

It is a programming error if the highest MIR's LIMIT is less than TOLM.

In addition, the MCH will decode the address to determine which rank the access is targeted for. It also helps with converting the memory (compressing gaps due to MMIO or FBD interleaving policies). The ESIR.LIMIT registers also decode the access into a specific interleaving range, which helps decode which rank the access will be directed to. The MCH will also convert the address into the appropriate DDR address, RAS, and CAS signals.

6.2 Memory Address Disposition

6.2.1 Registers Used for Address Routing

Table 6-5. Intel® 7300 Chipset Memory Mapping Registers

Name	Function
MIR[2:0]	Memory Interleaving Registers (FBD Branch Interleaving)
AMIR[2:0]	Scratch pad register for software to use related to memory interleaving. For example, software can write MMIO gap adjusted limits here to aid in subsequent memory RAS operations.
PAM[6:0]	Defines attributes for ranges in the C and D segments. Supports shadowing by routing reads and writes to memory of I/O
SMRAMC	SMM Control
EXSMRC, EXSMRAMC	Extended SMM Control
EXSMRTOP	Top of extended SMM memory
BCTRL	Contains VGAEN and ISAEN for each PCI Express.
TOLM	Top of low memory. Everything between TOLM and 4GB will not be sent to memory.
HECBASE	Base of the memory mapped configuration region that maps to all PCI Express registers
MBASE (dev 1-7)	Base address for memory mapped I/O to PCI Express ports 1- 7
MLIMIT (dev 1-7)	Limit address for memory mapped I/O to PCI Express ports 1- 7
PMBASE (dev 1-7)	Base address for memory mapped I/O to prefetchable memory of PCI Express ports 1-7[a]
PMLIMIT (dev 1-7)	Limit address for memory mapped I/O to prefetchable memory of PCI Express ports 1-7
PMBU (dev 1-7)	Prefetchable Memory Base (Upper 32 bits) - Upper address bits to the base address of prefetchable memory space. If the prefetchable memory is below 4GB, this register will be set to all 0's.
PMLU (dev 1-7)	Prefetchable Memory Limit (Upper 32 bits) - Upper address bits to the limit address of prefetchable memory space. If the prefetchable memory is below 4GB, this register will be set to all 0's.
PEXCMD (dev 1-7)	MSE (Memory Space Enable) bit enables the memory and pre-fetchable ranges.

Notes:

a. The chipset treats memory and prefetchable memory the same. These are just considered 2 apertures to the PCI Express port.

6.2.2 Address Disposition for Processor

The following tables define the address disposition for the Intel® 7300 Chipset. Table 6-6 defines the disposition of outbound requests entering the Intel® 7300 Chipset on the processor bus. Table 6-10 defines the disposition of inbound requests entering the MCH on I/O bus. For address dispositions of PCI-4Express/ESI devices, please refer

to the specifications for the PXH, IXH, Intel® 631xESB/632xESB I/O Controller Hub, etc. In these tables, an address listed as "A to B" can be interpreted as A <= Address <= B

Table 6-6. Address Disposition for Processor (Sheet 1 of 2)

Address Range	Conditions	Intel® 7300 Chipset Behavior
DOS	0 to 09FFFFh	Coherent Request to Main Memory. Route to main memory according to MIR registers. Apply Coherence Protocol.
SMM/VGA	0A0000h to 0BFFFFh	see Table 6-8 and Table 6-9.
C and D BIOS segments (see Table 6-1 for a definition of PAM encoding)	0C0000h to 0DFFFFh and PAM=11	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation) Route to appropriate FBD according to MIR registers.
	Write to 0C0000h to 0DFFFFh and PAM=10	
	Read to 0C0000h to 0DFFFFh and PAM=01	
	Read to 0C0000h to 0DFFFFh and PAM=10	Issue request to ESI.
	Write to 0C0000h to 0DFFFFh and PAM=01	
	0C0000h to 0DFFFFh and PAM=00	
E and F BIOS segments (see Table 6-1 for a definition of PAM encoding)	0E0000h to 0FFFFFh and PAM=11	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation) Route to appropriate FBD according to MIR registers.
	Write to 0E0000h to 0FFFFFh and PAM=10	
	Read to 0E0000h to 0FFFFFh and PAM=01	
	Read to 0E0000h to 0FFFFFh and PAM=10	Issue request to ESI.
	Write to 0E0000h to 0FFFFFh and PAM=01	
	0E0000h to 0FFFFFh and PAM=00	
Low/Medium Memory	10_0000 <= Addr < TOLM	Coherent Request to Main Memory. Route to main memory according to MIR registers. Apply Coherence Protocol. Note: the extended SMRAM space is within this range.
Extended SMRAM Space	ESMMTOP-TSEG_SZ <= Addr < ESMMTOP	see Table 6-8 and Table 6-9.
Low MMIO	TOLM <= Addr < FE00_0000 and falls into a legal BASE/LIMIT range	Request to PCI Express based on <MBASE/MLIMIT and PMBASE/PMLIMIT> registers.
	TOLM <= Addr < FE00_0000 and not in a legal BASE/LIMIT range	Send to ESI to be master aborted.
PCI Express MMCFG	HECBASE <= Addr < HECBASE+256MB	Convert to a configuration access and route according to the Configuration Access Disposition.
MCH specific	FE00_0000h to FEBF_FFFFh AND valid MCH memory mapped register address plus AMB targeted addresses	Issue configuration access to memory mapped register inside MCH or to the FBD based on the context.
	FE00_0000h to FEBF_FFFFh AND (NOT a valid MCH memory mapped register address or NOT a valid AMB targeted address)	Send to ESI to be master aborted.

Table 6-6. Address Disposition for Processor (Sheet 2 of 2)

Address Range	Conditions	Intel® 7300 Chipset Behavior
I/O APIC registers	FEC0_0000 to FEC8_FFFFh	Non-coherent request to PCI Express or ESI based on Table 6-3, "I/O APIC Address Mapping".
ICH/ICH timers	FEC9_0000h to FED1_FFFF	Issue request to ESI.
TPM support range	FED2_0000 to FED9_FFFF	see Table 6-4, "TPM support range address routing" on page 339
High SMM	FEDA_0000h to FEDB_FFFF	see Table 6-8 and Table 6-9.
Interrupt	interrupt transaction to FEE0_0000h to FEEF_FFFFh (not really memory space)	Route to appropriate FSB(s). See Interrupt Chapter for details on interrupt routing.
	memory transaction to FEE0_0000h to FEEF_FFFFh	Send to ESI to be master aborted.
Firmware	FF00_0000h to FFFF_FFFFh	Issue request to ESI.
High Memory	1_0000_0000 to MIR[2].LIMIT (max FF_FFFF_FFFF)	Coherent Request to Main Memory. Route to main memory according to MIR registers. Apply Coherence Protocol.
High MMIO	PMBU+PMBASE <= Addr <= PMLU+PMLIMIT	Route request to appropriate PCI Express port
All others	All Others (subtractive decoding)	Issue request to ESI.

6.2.2.1 Access to SMM Space (Processor only)

Only the processor is allowed to access SMM space. Inbound transactions to enabled SMM space are not allowed and Intel® 7300 Chipset will set EXSMRAMC.E_SMERR bit. The following table defines when a SMM range is enabled. All the enable bits: G_SMRAME, H_SMRAM_EN, and TSEG_EN are located in the EXSMRC register.

Table 6-7. Enabled SMM Ranges

Global Enable G_SMRAME	High SMM Enable H_SMRAM_EN	TSEG Enable TSEG_EN	Legacy SMM Enabled?	HIGH SMM Enabled?	Extended SMRAM Space (TSEG) Enabled?
0	X	X	No	No	No
1	0	0	Yes	No	No
1	0	1	Yes	No	Yes
1	1	0	No	Yes	No
1	1	1	No	Yes	Yes

The processor bus has a SMMEM# signal that qualifies the request asserted as having access to a system management memory. The SMM register defines SMM space that may fall in one of three ranges: legacy SMRAM, Extended SMRAM Space (TSEG), or High SMRAM Space (H_SMM). Table 6-8 defines the access control of SMM memory regions from processors.

Table 6-8. SMM Memory Region Access Control from Processor

G_SMRAME	D_LCK	D_CLS	D_OPEN	SMMEM#	Code access to SMM memory[a]	Data access to SMM memory[b]
0[c]	x	x	x	x	no	no
1	0	x	0	0	no	no
1	0	0	0	1	yes	yes
1	0	0	1	x	yes	yes
1	0	1	0	1	yes	no (legacy SMM) yes (H_SMM, TSEG)
1[d]	x	1	1	x	illegal settings	illegal settings
1	1	0	x	0	no	no
1	1	1	0	0	no	no
1	1	0	x	1	yes	yes
1	1	1	0	1	yes	no (legacy SMM) yes (H_SMM, TSEG)

Notes:

a. BRLC
b. Data access transaction other than BRLC
c. For access to TSEG region (address range between ESMMTOP - TSEG_SZ and ESMMTOP), The MCH will route to identical system memory by definition (as TSEG is not enabled).
d. It is a programming error if D_CLS and D_OPEN are both set to 1, the MCH's behavior is undefined. The MCH could master abort SMM access.

The Intel® 7300 Chipset prevents illegal processor access to SMM memory. This is accomplished by routing memory requests from processors as a function of transaction request address, code or data access, the SMMEM# signal accompanying request and the settings of the SMRAMC, EXSMRC, and BCTRL registers. Table 6-9 defines the MCH's routing for each case. Illegal accesses are either routed to the ESI bus where they are Master Aborted or are blocked with error flagging. SMMEM# only affects the MCH behavior if it falls in an enabled SMM space. Note that the D_CLS only applies to the legacy (A_0000-B_FFFF) SMM region. The bold values indicate the reason SMM access was granted or denied.

Note: If a spurious inbound access targets the enabled SMM range (viz., legacy, High SMM Memory and Extended SMRAM (T-segment)), then it will be Master-aborted. The EXSMRAMC.E_SMERR register field (Invalid SMRAM) is set for accesses to the High SMM Memory and Extended SMRAM (T-segment). Refer to Section 6-10, "Address Disposition for Inbound Transactions" on page 347.

Table 6-9. Decoding Processor Requests to SMM and VGA Spaces

SMM region	Transaction Address Range	SMM Memory Address Range	SMM Access Control [a]	G_SMRAME	H_SMRAME	T_EN	EWB/IWB	Routing
Legacy VGA/SMM [b]	A_0000h to B_FFFFh	A_0000h to B_FFFFh	x	**0**	x	x	x	to the VGA-enabled port (in BCTRL); otherwise, ESI[c]
			yes	1	**1**	x	x	
			no	1	x	x	x	
			yes	1	**0**	x	x	to SMM memory
Extended SMRAM (TSEG)	ESMMTOP -TSEG_SZ to ESMMTOP	ESMMTOP -TSEG_SZ to ESMMTOP	x	**0**	x	x	x	to identical system memory by definition
			x	1	x	**0**	x	
			yes	1	x	**1**	x	to SMM memory
			no	1	x	1	**1**	
			no	1	x	1	0	block access: master abort set EXSMRAMC.E_SMERR
High SMM	FEDA_0000h to FEDB_FFFFh	A_0000h to B_FFFFh	x	**0**	x	x	x	to ESI (where access will be master aborted)
			x	1	**0**	x	x	
			yes	1	**1**	x	x	to SMM memory[d]
			no	1	1	x	**1**	
			no	1	1	x	0	block access: master abort set EXSMRAMC.E_SMERR

Notes:

a. SMM memory access control, see Table 6-8.
b. Software must not cache this region.
c. One and only one BCTRL can set the VGAEN; otherwise, send to ESI.
d. Notice this range is mapped into legacy SMM range (A_0000h to B_FFFFh).

6.2.3 Inbound Transactions

In general, inbound I/O transactions are decoded and dispositioned similarly to processor transactions. The key differences are in SMM space, memory mapped configuration space, and interrupts. Inbound transaction targeting at itself will be master aborted.

Note that inbound accesses to the SMM region must be handled in such a way that FSB snooping and associated potential implicit writebacks are avoided. This is necessary to prevent compromising SMM data by returning real content to the I/O subsystem. Note also that DMA engine is treated as an I/O device, thus accesses initiated by the DMA engine are considered as inbound accesses.

For all table entries where an access is forwarded to ESI to be master aborted, if an access comes from ESI, the MCH may master abort a transaction without forwarding it back to the ESI.

Table 6-10. Address Disposition for Inbound Transactions

Address Range	Conditions	Intel® 7300 Chipset Behavior
DOS	0 to 09FFFFh	Coherent Request to Main Memory. Route to main memory according to MIR registers. Apply Coherence Protocol.
SMM/VGA	0A0000h to 0BFFFFh, and VGAEN=0	Send to ESI to be master aborted. Set EXSMRAMC.E_SMERR
	0A0000h to 0BFFFFh and VGAEN=1	Non-coherent read/write request to the decoded PCI Express or to ESI based on BCTRL[a]
C, D, E, and F BIOS segments (see Table 6-1 for a definition of PAM encoding)	0C0000h to 0FFFFFh and PAM=11[b]	Non-coherent request to main memory. (Coherency does not need to be guaranteed. Coherency protocol can be followed if it simplifies implementation.) Route to appropriate FBD according to MIR registers.
Low/Medium Memory	10_0000 <= Addr < ESMMTOP -TSEG_SZ	Coherent Request to Main Memory. Route to main memory according to MIR registers. Apply Coherence Protocol.
Extended SMRAM Space	ESMMTOP -TSEG_SZ <= Addr < ESMMTOP	Send to system memory if G_SMRAME = 0 or (G_SMRAME = 1 and T_EN = 0); otherwise Send to ESI to be master aborted. Set EXSMRAMC.E_SMERR
Low MMIO	TOLM <= Addr < FE00_0000 and falls into a legal BASE/LIMIT range	Request to PCI Express based on <MBASE/MLIMIT and PMBASE/PMLIMIT> registers.
	TOLM <= Addr < FE00_0000 and not in a legal BASE/LIMIT range	Send to ESI to be master aborted.
PCI Express MMCFG	HECBASE <= Addr < HECBASE+256MB	Inbound MMCFG access is not allowed and will be aborted.
MCH specific	FE00_0000h to FEBF_FFFFh AND valid MCH memory mapped register address	Inbound MMCFG access is not allowed and will be aborted.
	FE00_0000h to FEBF_FFFFh AND NOT a valid MCH memory mapped register address	Send to ESI to be master aborted.
I/O APIC registers	FEC0_0000 to FEC8_FFFFh	Non-coherent request to PCI Express or ESI based on Table 6-3, "I/O APIC Address Mapping"
ICH/ICH timers	FEC9_0000h to FED1_FFFF	Issue request to ESI.
TPM support range	FED2_0000 to FED9_FFFF	see Table 6-4, "TPM support range address routing" on page 339
High SMM	FEDA_0000h to FEDB_FFFF	Send to ESI to be master aborted. Set EXSMRAMC.E_SMERR
Interrupt	Inbound write to FEE0_0000h - FEEF_FFFFh	Route to appropriate FSB(s). See Interrupt Chapter for details on interrupt routing.
	memory transaction (other than write) to FEE0_0000h - FEEF_FFFFh	Send to ESI to be master aborted.
Firmware	FF00_0000h to FFFF_FFFFh	Master abort
High Memory	1_0000_0000 to MIR[2].LIMIT (max FF_FFFF_FFFF)	Coherent Request to Main Memory. Route to main memory according to MIR registers. Apply Coherence Protocol.
High MMIO	PMBU+PMBASE <= Addr <= PMLU+PMLIMIT	Route request to appropriate PCI Express port
All others	All Others (subtractive decoding)	Issue request to ESI.

Notes:

a. One and only one BCTRL can set the VGAEN; otherwise, send to ESI for master abort.
b. Other combinations of PAM's are not allowed if inbound accesses to this region can occur. Just like Cayuse, chipset functionality is not guaranteed.

6.3 I/O Address Map

The I/O address map is separate from the memory map and is primarily used to support legacy code/drivers that use I/O mapped accesses rather than memory mapped I/O accesses. Except for the special addresses listed in Section 6.3.1, "Special I/O addresses", I/O accesses are decoded by range and sent to the appropriate ESI/PCI Express port, which will route the I/O access to the appropriate device.

6.3.1 Special I/O addresses

There are two classes of I/O addresses that are specifically decoded by the MCH:

- I/O addresses used for VGA controllers.
- I/O addresses used for the PCI Configuration Space Enable (CSE) protocol. The I/O addresses 0CF8h and 0CFCh are specifically decoded as part of the CSE protocol.

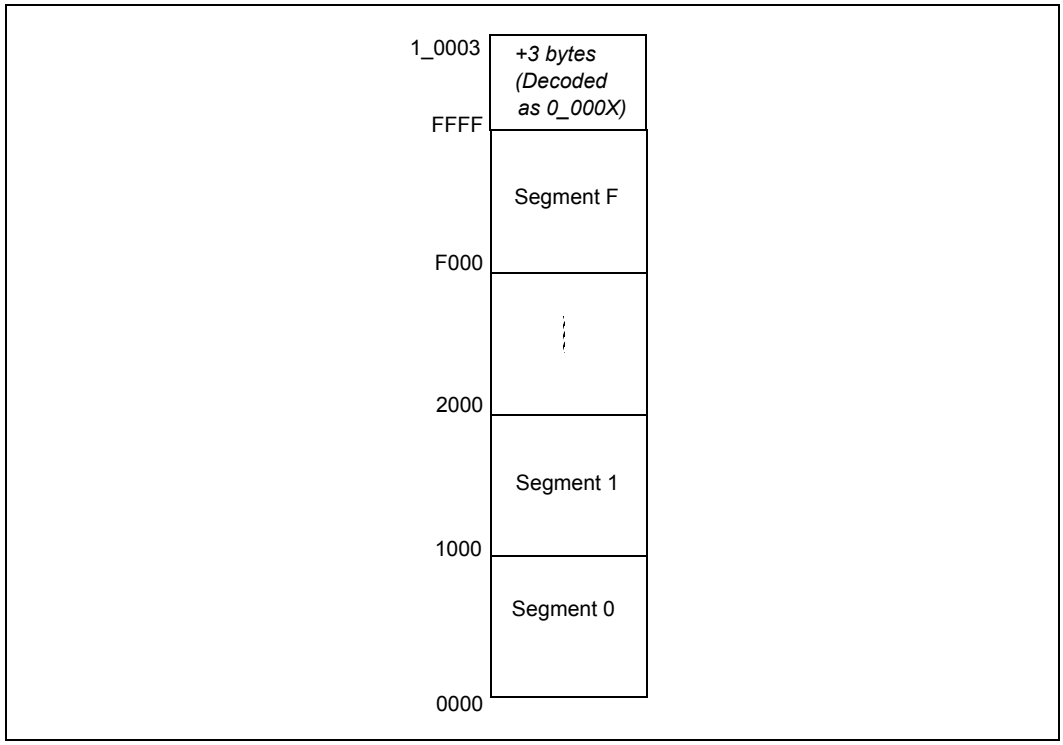
Historically, the 64 K I/O space actually was 64 K+3 bytes. For the extra three bytes, A#[16] is asserted on FSB. The MCH decodes only A#[15:3] when the request encoding indicates an I/O cycle. Therefore first three byte I/O accesses with A#[16] asserted are decoded as if they were accesses to the first three bytes starting from I/O addresses 0 (wrap-around the 64KB line). A[16] is not forwarded by the MCH.

At power-on, all I/O accesses are mapped to the ESI.

6.3.2 Outbound I/O Access

The Intel® 7300 Chipset allows I/O addresses to be mapped to resources supported on the I/O buses underneath the MCH. This I/O space is partitioned into 16 4 KB segments. Each of PCI Express port can have from 1 to 16 consecutive segments mapped to it by programming its IOBASE and IOLIM registers. Each PCI Express port must be assigned contiguous segments. The lowest segment, from 0 to 0FFFh, should be programmed to send to the ESI for compatibility.

Figure 6-3. System I/O Address Space



6.3.2.1 Outbound I/O accesses routing

The Intel® 7300 Chipset applies these routing rules **in the following order**: (A[2:0] for the following is not physically present on the processor bus, but are calculated from BE[7:0]).

1. I/O addresses used for VGA controllers on PCI Express:
   If PEXCMD[y].IOAE and BCTRL[y].VGAEN of PCI Express port y are set to 1 and BCTRL[y].VGA16bdecode = **0**, then I/O accesses with the following VGA addresses will be forwarded to PCI Express port y: A[9:0] (A[15:10] are ignored for this decode since BCTRL[y].VGA16bdecode is set to 0) = 3B0h - 3BBh, 3C0h - 3DFh if every addressed byte is within these two ranges. For example, a two byte read starting at X3BBh includes X3BB -X3BCh. (X can be any hex number since A[15:10] are ignored) Since the second byte with A[9:0] = 3BCh is not within these ranges, the access is not routed to port y.
   If PEXCMD[y].IOAE and BCTRL[y].VGAEN of port y are set to 1 and BCTRL[y].VGA16bdecode = **1**, then I/O accesses with the following VGA addresses will be forwarded to PCI Express port y: A[**15**:0] = **0**3B0h - **0**3BBh, **0**3C0h - **0**3DFh if every addressed byte is within these two ranges. For example, a four byte I/O read starting at F3B0h includes F3B0 - F3B3h are not within these ranges, the access is not routed to port y.
   Note that software should program PEXCMDs and BCTRLs to ensure that **at most only one** port is allowed to forward these accesses with VGA addresses. It is a programming error if more than one port are programmed to forward accesses with VGA addresses.
2. Configuration accesses: If a request is a DW accesses to 0CF8h (See CFGADR register) or 1-4B accesses to 0CFCh (See CFGDAT register) with configuration space enabled (See CFGE bit, bit 31, of CFGADR register), the request is

considered a configuration access. Configuration accesses are routed based on the bus and device numbers as programmed by software.

3. ISA Aliases: If the PEXCMD[y].IOAE and BCTRL[y].ISAEN are set to 1 for a PCI Express port y and the I/O address falls within (IOBASE[y], IOLIMIT[y]) and if the addresses are X100-X3FFh, X500-X7FFh, X900-XBFF, and XD00-XFFFh (X can be any hex number) will result in the access being sent out to the ESI (Intel® 631xESB/632xESB I/O Controller Hub). This is the top 768B in each 1KB block.
4. I/O defined by IOBASE/IOLIMIT: If PEXCMD[y].IOAE is set for a given PCI Express port and the I/O address falls in this range: (IOBASE[y] <= address <= IOLIMIT[y]) for that port, then the access will be routed to the PCI Express port y.
5. Otherwise, the I/O Read/Write is sent to ESI (Intel® 631xESB/632xESB I/O Controller Hub).

6.3.3 Inbound I/O Access

Inbound I/Os are supported only for peer to peer accesses and are decoded the same as processor initiated I/Os. Inbound I/O transactions for configuration access to the MCH (i.e. CF8/CFC as in step 2. of Section 6.3.2.1.) are not supported and will be routed as normal inbound I/O transaction.

Inbound requests (memory, I/O) received from a PCI Express port and which target the same port as destination will be Master Aborted.

6.4 Configuration Space

All chipset registers are represented in the memory address map. In addition, some registers are also mapped as PCI registers in PCI configuration space.These adhere to the *PCI Local Bus Specification*, Revision 2.2

The memory mapped configuration space is described in Section 6.1.2.4, “Memory Mapped Configuration (MMCFG) Region”. Individual register maps are in Section 4, “Register Description”.

If a Processor issues a zero length configuration cycle accessing the MCH’s internal configuration space registers or the IQD_BAR/AMB Memory mapped area, then it will be completed on the FSB “inorder” with no data.

§

7 Functional Description

This chapter describes each of the MCH interfaces and functional units including the Processor Front Side Bus interfaces, PCI Express ports, system memory controller, power management, and clocking.

7.1 Processor Front Side Bus

The MCH supports four Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series processors, based on 65 nmprocess technologies, in a 604pin FC-uPGA4 package. Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Seriesare first generation MP processors based on Intel® Core™ microarchitecture.

The MCH supports four 1066 MHz Front Side Bus (FSB) interfaces which are quad-pumped buses running off of a 266 MHz system clock. Each processor FSB supports peak address generation rates of 533 Million Addresses/second (MA/s). All FSB data buses are quad-pumped 64 bits which allows peak bandwidths of 8.5 GB/s (1066 MT/s). The MCH supports 40-bit physical addressing, capable of decoding up to 1 Terabyte (TB) of system address space. Host-initiated I/O cycles are decoded to PCI, PCI Express, ESI or MCH configuration space. Host-initiated memory cycles are decoded to PCI, PCI Express, ESI or system memory.

7.1.1 Front Side Bus Overview

The MCH is the only priority agent for four point-to-point, independent, processor FSBs. The MCH maintains coherency across these four buses. The MCH may complete deferrable transactions with either deferred-replies or in-order responses. Intel® 7300 Chipset contains an internal Snoop Filter (SF) to remove unnecessary snoops on the remote FSB, and to be able to complete transactions in-order without deferring for transactions that do not need to have a remote snoop. Data transactions on the FSBs are optimized to support 64-byte cache lines.

Each processor FSB contains a 40-bit address bus, a 64-bit data bus, and associated control signals. The FSB supports a split-transaction, deferred reply protocol. The FSB uses source-synchronous transfer of address and data to improve performance. The FSB address bus is double pumped (2X) with ADS being sourced every other clock. The address bus generates a maximum bandwidth of 533 MA/s. The FSB data bus is quad-pumped (4X) and supports peak bandwidths of 8.5 GB/s (1066 MT/s). Parity protection is applied to the address and data bus. This yields a combined bandwidth of 34 GB/s for all FSBs.

Interrupts are also delivered via the FSB.

7.1.2 FSB Dynamic Bus Inversion

The MCH supports Dynamic Bus Inversion (DBI) when driving and when receiving data from the processor. DBI limits the number of data signals that are driven to a low voltage on each quad-pumped data phase. This decreases the worst-case power consumption of the MCH. The FSB{0-3}DBI_N[3:0] signals for FSB0-FSB3, respectively, indicate if the corresponding 16 bits of data are inverted on the bus for each quad-pumped data phase.

Table 7-1. FSB{0-3}DBI_N[3:0] / Data Bit Correspondence

FSB{0-3}DBI_N[3:0]	Data Bits
FSB{0-3}DBI_N[0]	FSB{0-3}D_N[15:0]
FSB{0-3}DBI_N[1]	FSB{0-3}D_N[31:16]
FSB{0-3}DBI_N[2]	FSB{0-3}D_N[47:32]
FSB{0-3}DBI_N[3]	FSB{0-3}D_N[63:48]

When the processor or the MCH drives data, each 16-bit segment is analyzed. If more than 8 of the 16 signals would normally be driven low on the bus, the corresponding FSB{0-3}DBI_N[{0-3} signal will be asserted and the data will be inverted prior to being driven on the bus. When the processor or the MCH receives data, it monitors FSB{0-3}DBI_N[3:0] to determine if the corresponding data segment should be inverted.

7.1.3 FSB Interrupt Overview

The Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Seriesprocessors supports FSB interrupt delivery. The legacy APIC serial bus interrupt delivery mechanism is not supported. Interrupt-related messages are encoded on the FSB as "Interrupt Message Transactions." In the platform, FSB interrupts may originate from the processor on the FSB, or from a downstream device on the Enterprise South Bridge Interface (ESI) or PCI Express ports. In the case of interrupts generated by downstream agents, the MCH drives the Interrupt Message Transaction onto the FSB.

In the Intel® 7300 Chipset chipset the Intel® 631xESB/632xESB I/O Controller Hub contains IOxAPICs, and its interrupts are generated as upstream ESI memory writes. Furthermore, PCI 2.3 defines Message Signaled Interrupts (MSI) that are also in the form of memory writes. A PCI 2.3 device may generate an interrupt as an MSI cycle on its PCI bus instead of asserting a hardware signal to the IOxAPIC. The MSI may be directed to the IOxAPIC which in turn generates an interrupt as an upstream ESI memory write. Alternatively, the MSI may be directed directly to the FSB. The target of an MSI is dependent on the address of the interrupt memory write. The MCH forwards inbound ESI and PCI (PCI semantic only) memory writes to address 0FEEx_xxxxh to the FSB as Interrupt Message Transactions.

7.1.3.1 Upstream Interrupt Messages

The MCH accepts message-based interrupts from PCI (PCI semantics only) or ESI and forwards them to the FSB as Interrupt Message Transactions. The interrupt messages presented to the MCH are in the form of memory writes to address 0FEEx_xxxxh. At the ESI or PCI Express interface, the memory write interrupt message is treated like any other memory write; it is either posted into the inbound data buffer (if space is available) or retried (if data buffer space is not immediately available). Once posted, the memory write from PCI or ESI to address 0FEEx_xxxxh is decoded as a cycle that needs to be propagated by the MCH to the FSB as an Interrupt Message Transaction.

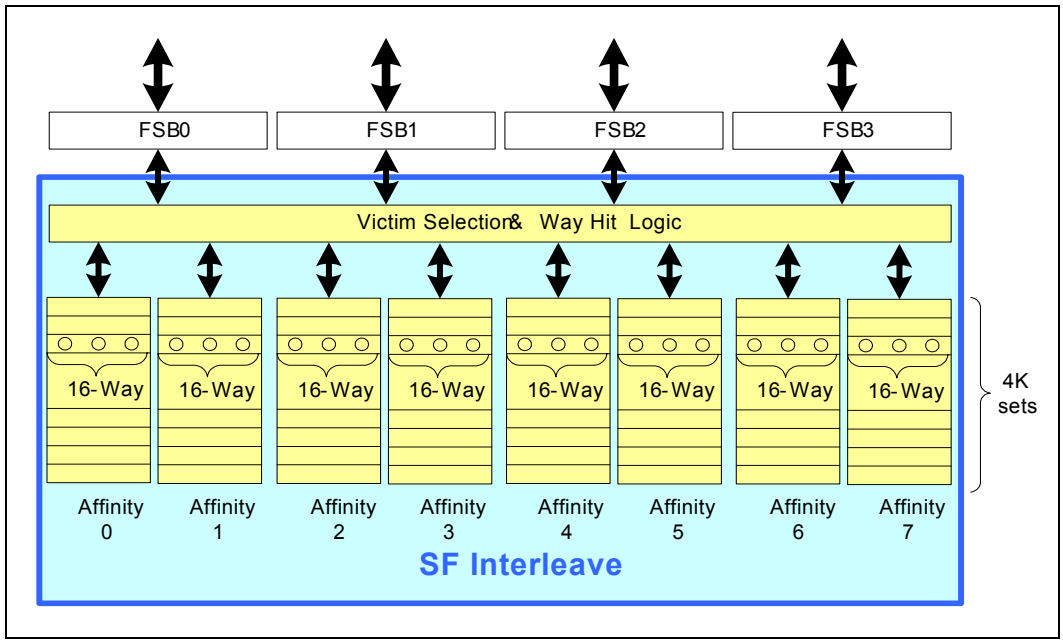
7.2 Snoop Filter

The Snoop Filter (SF) offers significant performance enhancements by eliminating unnecessary FSB snoop traffic. By removing snoops from the snooped bus, the full bandwidth is available for other transaction processing. Supporting concurrent snoops effectively reduces performance degradation attributable to multiple snoop stall.

The SF is composed of eight affinity groups each containing 4 K sets of x16-way associative entries. The SF tracks 64 MB of processor L2 cachelines. Each affinity group supports an Active Way management algorithm. Lookups are done on a 128-way lookup, full 16-way per set for 8 sets for hit/miss checks.

The SF is organized as shown in Figure 7-1.

Figure 7-1. Snoop Filter



The SF stores the tags and coherency state information for all cache lines in the system. The SF is used to determine if a cache line associated with an address is cached in the system and where. The coherency protocol engine (CE) accesses the SF to look-up entries for inbound coherent transactions, update/add an entry, or invalidate an entry in the snoop filter.

The SF has the following features:

- Snoop Filter tracks a total of 64MB of processor L2 cachelines, this is equivalent to: (64 * (2^20)byte) / 64 byte CL = 1,048,576 cache lines.
- The SF is partitioned into two SF interleaves which are selected based on A[6].
- Each SF interleave is configured in 4K sets organized as an 8 Affinity x 16Way x 4K (128Way x 4K) Set-Associativity array. This is equivalent to (2^12 Sets) x 16Way x 8 DID = 524,288 tag entries
- 8 x 16 Affinity Set-Associativity will allocate/evict entries within the 16-way corresponding to the assigned affinity group if the SF look up is a miss. Each SF look up will be based on 128-way (8x16 ways) look up
- The size of the snoop filter interleave Tag RAM is: 4096 sets * 8 affinities * 16 ways * 35 bits/affinity/ set/way = 2,293,760 bytes. The total SF size (2 interleaves) is 4,587,520 bytes.
- The size of the snoop filter Victim Ram is: 4096 sets * 8 affinities * 8 bits = 32,768 bytes

- The size of the snoop filter Random ROM is: 1024 addresses * 16 bits = 2,048 bytes
- Each Snoop Filter interleave is operated at 2x of the MCH core frequency, i.e. 533MHz to provide 267 MLUU/s (where a Look-Up-Update operation is a read followed by a write operation to the tag array).
  - The maximum lookup and update bandwidth of the Snoop Filter is equal to the max request bandwidth from two FSB's. The lookup and update bandwidth from I/O coherent transactions have to share the bandwidth with all four FSBs per request weighted-round-robin arbitration
  - The SF interleave lookup latency is four SF-clocks or two MCH core clocks to support minimum number of snoop stall in idle condition.
- Active Way / Invalid replacement algorithm, with updates on lookups and invalidates.
- Tag entries supporting a 40-bit physical address space. The space is much larger than the physical memory can provide.
- Stores coherency state (EM) and Bus Presence Vector BusPV[3:0] for each valid cache line in the system. Here is the summary of the snoop-filter state definitions:
  - Coherency state: the cache line is in E/M state if the bit is set; else, the line is in the Shared state.
  - If BusPV[3:0]=0000, the entry is invalid, i.e. unoccupied (EM is a don't care).
  - If BusPV[3:0]=0001, the FSB0 processor(s) has ownership of the line. If the EM is set, the line ownership is in Exclusive/Modified state; otherwise, the line is in Shared state.
  - If BusPV[3:0]=0010, the FSB1 processor(s) has ownership of the line. If the EM is set, the line ownership is in Exclusive/Modified state; otherwise, the line is in the Shared state.
  - If BusPV[3:0]=0100, the FSB2 processor(s) has ownership of the line. If the EM is set, the line ownership is in Exclusive/Modified state; otherwise, the line is in the Shared state.
  - If BusPV[3:0]=1000, the FSB3 processor(s) has ownership of the line. If the EM is set, the line ownership is in Exclusive/Modified state; otherwise, the line is in the Shared state.
  - If BusPV[3:0] has more than one bit set and the EM bit is deasserted, the line is in the Shared state with those processor(s) on each of those buses with the corresponding Bus Presence Vector bit set.
  - EM and more than one BusPV[3:0] are set concurrently is a reserved definition.
- Tag Array ECC coverage, with correction of single bit errors, detection of double bit errors (SEC-DED).
- Snoop-Filter Fast array initialization through configuration register access.

7.3 System Memory Controller

The MCH masters four Fully-Buffered DIMM (FB-DIMM) memory channels. FB-DIMM memory utilizes a narrow high speed frame oriented interface referred to as a channel.

The four FB-DIMM channels are organized into two branches of two channels per branch. Each branch is supported by a separate Memory Controller (MC). The two channels on each branch operate in lock step to increase FB-DIMM bandwidth. A branch transfers 16 bytes of payload/frame on Southbound lanes and 32 bytes of payload/frame on Northbound lanes.

The MCH will comply with the FB-DIMM specification definition of a host and will be compatible with any FB-DIMM-compliant DIMM that supports the following parameter limitations:

Table 7-1. Intel® 7300 Chipset FB-DIMM Limitations

Area	Intel® 7300 Chipset Limitation
Frequency	The MCH supports the following channel frequencies: 3.2 GHz for DDR 533 MHz, 4.0 GHz for DDR 667 MHz
Cache-line length	64B
Commands	The MCH supports access to main memory and DIMM configuration space over the FBD channel.
Link Training	At least every 42nd southbound packet must be an FBD "SYNC".
Function Support	The MCH supports configuration functions [7:0] for each DIMM.
Low Power Mode	The MCH will not enter low-power mode on failure to detect connected component, but software may force low power mode by forcing FBDHPC.STATE to Reset, and S1 will push FBD and PXP I/O pads into L1.
States	Not all of the initialization states defined in the *FBD channel specification* are software visible.

The key features of the FB-DIMM memory interface are summarized in the following list.

- Four Fully Buffered DDR2 (FB-DIMM) memory channels.
- Branch channels are paired together in lock step to match FSB bandwidth requirement.
- Each FB-DIMM Channel can link up to eight Fully Buffered DDR2 DIMMs
- Supports up to 32 dual-ranked DDR2 8 GB DIMMs (2 Gbit memory technology), i.e. 256 GB of physical memory in non-mirrored configuration or 128 GB of physical memory in mirrored configuration
- The FB-DIMM link speed is at 6x the DDR2 data transfer speed. A 3.2 GHz FB-DIMM link supports DDR2-533 (FSB@1067 MT/s). A 4.0 GHz FB-DIMM link supports DDR2-667 (FSB@1067 MT/s).
- Special single channel, single DIMM operation mode (Branch 0, Channel 0, Slot 0 position only).
- All memory devices must be DDR2.
- Reliability, Availability and Serviceability (RAS) features supported in the System Memory controller include
  - ECC protection on memory reads
  - CRC protection on FB-DIMM lanes
  - SDDC support
  - Demand and Patrol Scrubbing
  - Memory Mirroring
  - DIMM Sparing (Normal and Manual)

Table 7-2 and Figure 7-6 present system memory capacity as a function of DRAM device capacity and MCH operating mode.

Table 7-2. Minimum System Memory Configurations & Upgrade Increments

DRAM Technology	Smallest System Configuration - One DIMM	Smallest Upgrade Increment - Two DIMM
256 Mb	256 MB	512 MB
512 Mb	512 MB	1024 MB
1024 Mb	1024 MB	2048 MB
2048 Mb	2048 MB	4096 MB

The *Smallest System Configuration - One DIMM* column represents the smallest possible single DIMM capacity for a given technology (MCH operating in single channel, single DIMM mode with x8 single rank (x8SR) DIMM populated). The *Smallest Upgrade Increment - Two DIMMs* column represents the smallest possible memory upgrade capacity for a given technology using two x8 single rank DIMMs.

Table 7-3. Maximum 32 DIMM System Memory Configurations with x8 Single Rank DIMMs

DRAM Technology x8 Single Rank	Maximum Capacity Mirrored Mode	Maximum Capacity Non-Mirrored Mode
256 Mb	4 GB	8 GB
512 Mb	8 GB	16 GB
1024 Mb	16 GB	32 GB
2048 Mb	32 GB	64 GB

Note: The *Maximum Capacity Mirrored Mode* and *Maximum Capacity Non-Mirrored Mode* columns represent the system memory available when all DIMM slots are populated with identical x8 Single Rank (x8DR) DIMMs using the DRAM Technology indicated.

Table 7-4. Maximum 32 DIMM System Memory Configurations with x4 Dual Rank DIMMs

DRAM Technology x4 Dual Rank	Maximum Capacity Mirrored Mode	Maximum Capacity Non-Mirrored Mode
256 Mb	16 GB	32 GB
512 Mb	32 GB	64 GB
1024 Mb	64 GB	128 GB
2048 Mb	128 GB	256 GB

Note: The *Maximum Capacity Mirrored Mode* and *Maximum Capacity Non-Mirrored Mode* columns represent the system memory available when all DIMM slots are populated with identical x4 Double Rank (x4DR) DIMMs using the DRAM Technology indicated.

7.3.1 Fully Buffered DIMM Technology and Organization

Fully Buffered DIMM technology was developed to address the higher performance needs of server and workstation platforms. FB-DIMM addresses both the need for higher bandwidth and larger memory sizes.

FB-DIMM memory DIMMs contain an Advanced Memory Buffer (AMB) device that serves as an interface between the point to point FB-DIMM Channel links and the DDR2 DRAM devices. Each AMB is capable of buffering up to two ranks of DRAM devices. Each AMB supports two complete FB-DIMM channel interfaces. The first FB-DIMM interface is the incoming interface between the AMB and its proceeding device. The second

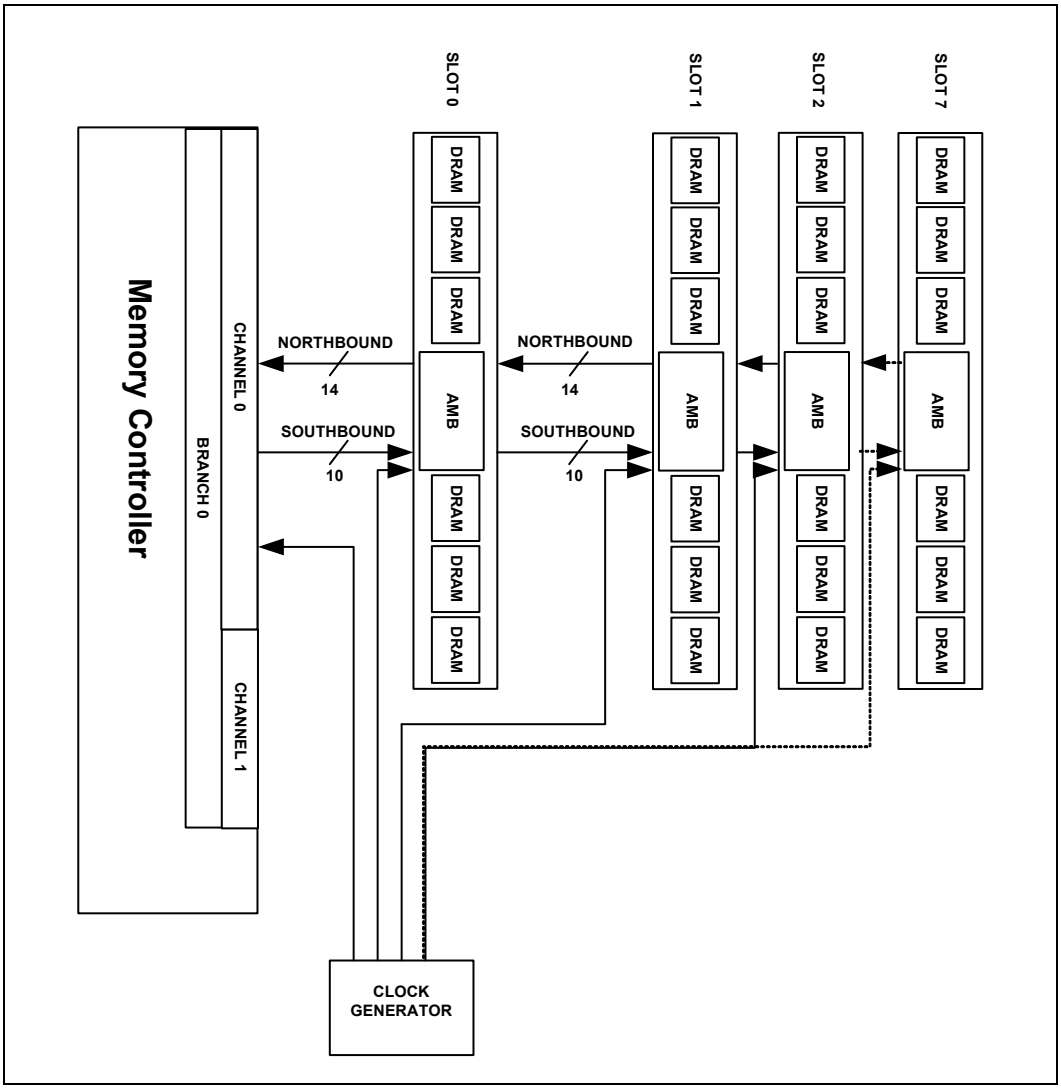
interface is the outgoing interface and is between the AMB and its succeeding device. The point to point FB-DIMM links are terminated by the last AMB in a chain. The outgoing interface of the last AMB requires no external termination.

There are three major components of the FB-DIMM channel interface:

- 14 Differential Northbound Signal pairs
- 10 Differential Southbound Signal pairs
- 1 Differential Clock Signal pair

Figure 7-2 depicts a single FB-DIMM channel with these three signal groups.

Figure 7-2. FB-DIMM Channel Schematic



An FB-DIMM channel consists of 14 unidirectional differential signal pairs referred to as the Northbound path, 10 unidirectional differential signal pairs referred to as the Southbound path, and a differential reference clock.

Note: The northbound signal pairs are enumerated from 0 to 13.

The southbound path is used to convey DIMM commands and write data to the addressed DIMMs. The northbound path returns read data and status from the addressed DIMM.

The northbound and southbound paths are used to convey FB-DIMM frames that are synchronized to the reference clock. Each frame consists of 12 data transfers. Southbound frames contain a payload of 8 bytes per frame per channel. Northbound frames contain a payload of 16 bytes per frame per channel.

7.3.2 FB-DIMM Memory Configuration Mechanism

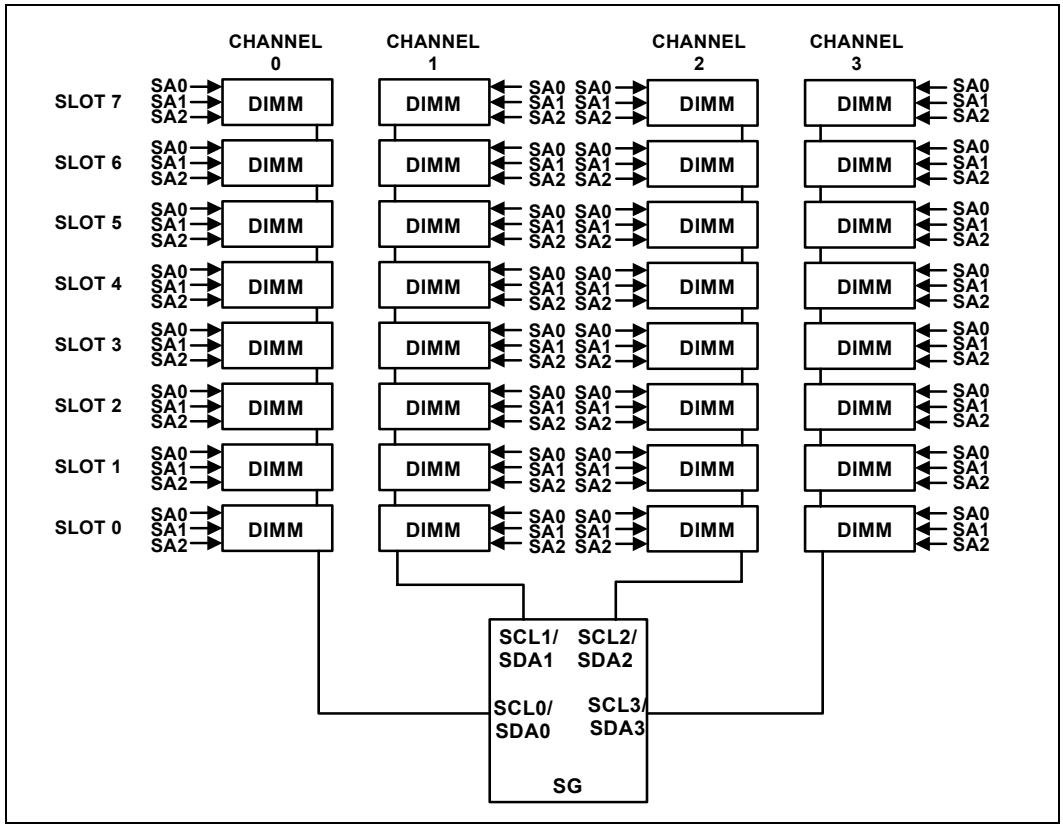
Before any cycles to the memory interface can be supported, the MCH DRAM registers must be initialized. The MCH must be configured for operation with the installed memory types. Detection of memory type and size is accomplished via the 4 Serial Presents Detect (System Management Bus) interfaces on the MCH (SMBus 1, 2, 3 and 4). The SMBus interfaces are two-wire buses used to extract the DRAM type and size information from the Serial Presence Detect port on the DIMMs.

FB-DIMMs contain a 6-pin Serial Presence Detect interface, which includes SCL (serial clock), SDA (serial data), and SA[3:0] (serial address). Devices on the SMBus bus have a 7-bit address. For the DIMMs, the upper three bits are fixed at 101. The lower four bits are strapped via the SA[3:0] pins. SCL and SDA are connected to the respective SPDxSMBDATA, SPDxSMBCLK pins on the MCH, see Figure 7-3.

The Intel® 7300 Chipset MCH integrates a 100 KHz SPD controller to access the DIMM SPD EEPROM's. There are four SPD ports. SPD0SMBDATA, and SPD0SMBCLK are defined for channel 0; SPD1SMBDATA, and SPD1SMBCLK are defined for channel 1; SPD2SMBDATA, and SPD2SMBCLK are defined for channel 2; and SPD3SMBDATA, and SPD3SMBCLK are defined for channel 3. There can be a maximum of eight SPD EEPROM's associated with each SPD bus. Therefore, the SPD interface is wired as indicated in Figure 7-3.

For details on the SPD protocol and messaging, refer to Section 7.12.5, "FB-DIMM SPD Interface, SM Buses 1, 2, 3 and 4" on page 443

Figure 7-3. Connection of DIMM Serial I/O Signals



Board layout must map chip selects to SPD Slave Addresses as shown in Table 7-5. The slave address is written to the **SPDCMD** configuration register (see Section 4.8.26.2).

Table 7-5. SPD Addressing

SPD Bus	FBD Channel	SLOT	Slave Address
0	0	0	0
		1	1
		2	2
		3	3
		4	4
		5	5
		6	6
		7	7
1	1	0	0
		1	1
		2	2
		3	3
		4	4
		5	5
		6	6
		7	7
2	2	0	0
		1	1
		2	2
		3	3
		4	4
		5	5
		6	6
		7	7
3	3	0	0
		1	1
		2	2
		3	3
		4	4
		5	5
		6	6
		7	7

7.3.3 DDR2 Protocol

7.3.3.1 Posted CAS

Posted CAS timing is used.

7.3.3.2 Refresh

Regardless of the number of DIMMs installed, each rank will get a minimum of one refresh every eight periods defined by the DRT.TREF configuration register field. The refreshes cycle through all 16 DIMM ranks.

The DIMM enters self-refresh mode during an FB-DIMM fast reset.

7.3.3.3 Access Size

All memory accesses are 64B.

7.3.3.4 Transfer Mode

Each DIMM is programmed to use a burst-length of 32 bytes (4 transfers) across the channel. The Mode Register of each DIMM must be programmed for a burst length of 4, and interleave mode.

7.3.3.5 Invalid and Unsupported DDR Transactions

The memory controller prevents cycle combinations leading to data interruption or early termination. The memory controller prevents combinations of DDR commands that create bus contention (i.e. where multiple ranks would be required to drive data simultaneously on a DIMM). The memory controller does not interrupt writes for reads. A precharge command is provided, but early read or write termination due to precharge is not supported.

7.3.4 FB-DIMM Memory Operating Modes

The MCH supports two major modes of operation, mirrored and non-mirrored. There is also a single DIMM, single channel mode of operation referred to as single-channel mode. The MCH operates in lock step in all modes except in single-channel mode.

7.3.4.1 Non-Mirrored Mode Operation

When operating in non-mirrored mode the MCH operates the two branches independently. In non-mirrored mode the full MCH address space of 256 GB is available.

7.3.4.2 Single Channel Mode Operation

When only one of the four available memory channels is populated and used, the memory mode is called as Single Channel Mode Operation. The MCH has very limited support for single channel mode operation. Only 1 DIMM in slot 0, channel 0 (Branch 0) can be populated in Single Channel mode.

7.3.4.3 Mirrored Mode Operation

Memory Mirroring MCH is a user-selectable feature. Mirrored mode provides for complete recovery from a DIMM device failure. The mirroring feature is fundamentally a way for hardware to maintain two copies of all data in the memory subsystem, such that a hardware failure or uncorrectable error is no longer fatal to the system. When an uncorrectable error is encountered during normal operation, hardware simply retrieves the “mirror” copy of the corrupted data, and no system failure will occur unless both primary and mirror copies of the same data are corrupt simultaneously (statistically very unlikely).

When operating in mirrored mode FB-DIMM Branch 0 (Channels 0/1) and Branch 1 (Channels 2/3) contain replicate copies of data (are mirrored images). Since Branch 1 contains a replicate copy of Branch 0’s data, the maximum addressable memory is reduced to 128 GB.

When operating in mirrored mode both channels on a branch are operated in lock-step.

Mirrored mode must be selected at configuration time by enabling mirrored operation.

Note: Memory cannot be mirrored when using the DIMM sparing feature.

The general flows for mirroring are as follows:

- The same Write (in the non-failed case) is issued to both images in the same cycle (which is complete when both branches acknowledge).
- Different Reads (in the non-failed case) can be issued to each image in the same cycle.
- Corrected data will be forwarded to the requester.
- Uncorrectable errors will be retried from the other image. If the other image is off-line, uncorrectable errors will be retried from the same image.
- To recover from a failed DIMM:
  - DIMM failure is detected
  - The defective branch is shut down (also known as off-lining the defective branch, or as working in degraded mode)
  - The system is gracefully shut down by operator and the defective DIMM is replaced
  - The system is repowered up
  - Normal processing is restored

7.3.5 Memory Population Rules

DIMM population rules depend on the operating mode of the memory controller. When operating in non-mirrored mode the minimum memory upgrade increment is two **identical** DIMMs per branch (DIMMs must be identical with respect to device size, speed, number of ranks, and organization). When operating in mirrored mode the minimum upgrade increment is four **identical** DIMMs.

7.3.5.1 Non-Mirrored Mode Memory Upgrades

The minimum memory upgrade increment is two DIMMs per branch. The DIMMs must cover the same slot position on both channels. DIMMs that cover a slot position must be identical with respect to size, speed, and organization. DIMMs that cover adjacent slot positions need not be identical.

Within a branch, memory DIMMs must be populated in slot order. Slot 0 is populated first, slot 1 second, slot 2 third, and slot 3 last. Slot 0 is closest to the MCH.

Section 7-4 depicts the minimum two DIMM configuration. The populated DIMMs are depicted in gray (Slot 0 of Branch 0 populated).

Figure 7-4. Minimum Two DIMM Configuration

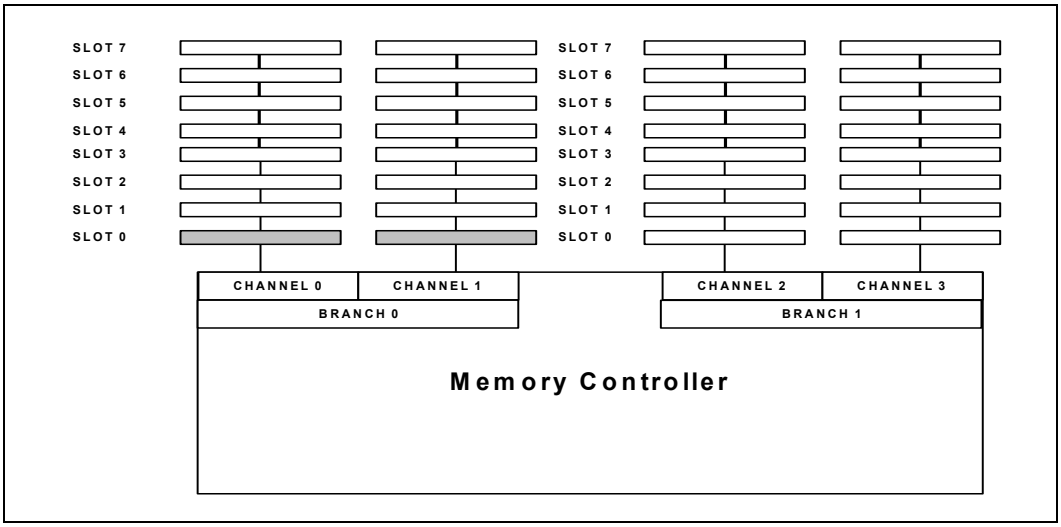
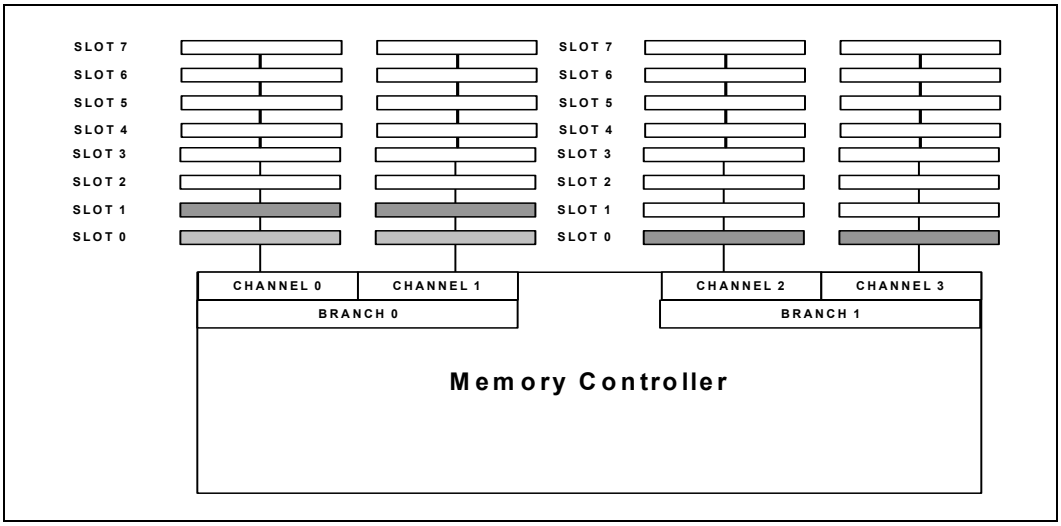


Figure 7-5 depicts the next two positions where DIMMs may be added. These positions are depicted in dark gray. The two upgrade positions are Branch 0, Slot 1 and Branch 1, Slot 0. Of these Branch 1, Slot 0 is the preferred upgrade because it allows both branches to operate independently and simultaneously. FB-DIMM memory bandwidth is doubled when both branches operate in parallel.

While it is possible to completely populate one branch before populating the second branch, it is not desirable to do so from a performance standpoint. In general memory upgrades should be balanced with respect to both branches to optimize FB-DIMM performance.

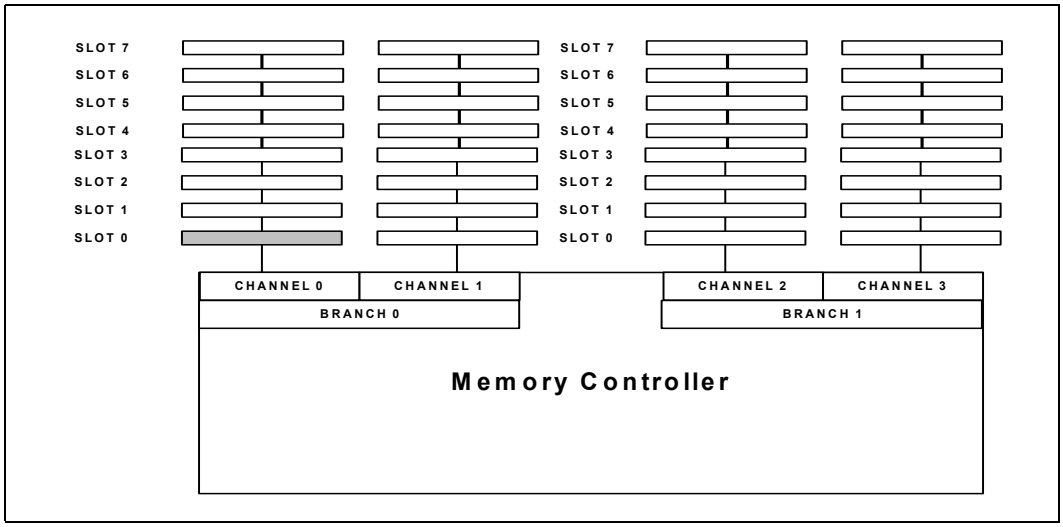
Figure 7-5. Next Two DIMM Upgrade Positions



7.3.5.2 Single Channel Mode Memory Upgrades

Figure 7-6 depicts a special single DIMM non-mirrored operation mode. This mode requires that the DIMM be placed in Branch 0, Channel 0, Slot 0. When upgrading from this mode the normal two DIMM memory upgrade rules are followed.

Figure 7-6. Single DIMM Operation Mode



7.3.5.3 Mirrored Mode Memory Upgrades

When operating in mirrored mode both branches operate in lock step. In mirrored mode Branch 1 contains a replicate copy of the data in Branch 0. For this reason the minimum memory upgrade increment, for mirrored mode, is four DIMMs across all branches. The DIMMs must cover the same slot position on both branches. DIMMs that cover a slot position must be identical with respect to size, speed, and organization. DIMMs within a slot position must match each other, but aren't required to match adjacent slot positions.

Figure 7-7 shows the minimum memory configuration required to operate in mirrored mode.

Figure 7-7. Minimum Mirrored Mode Memory Configuration

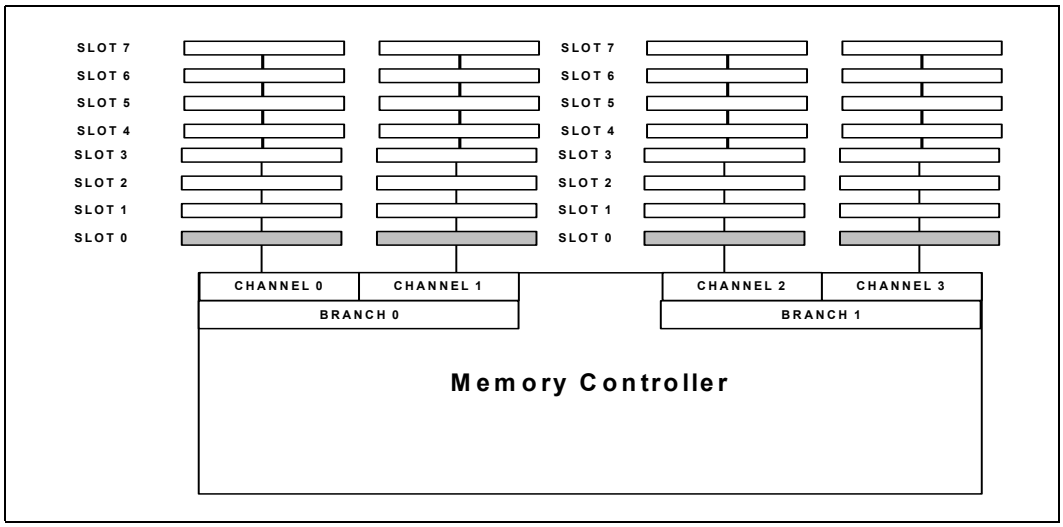
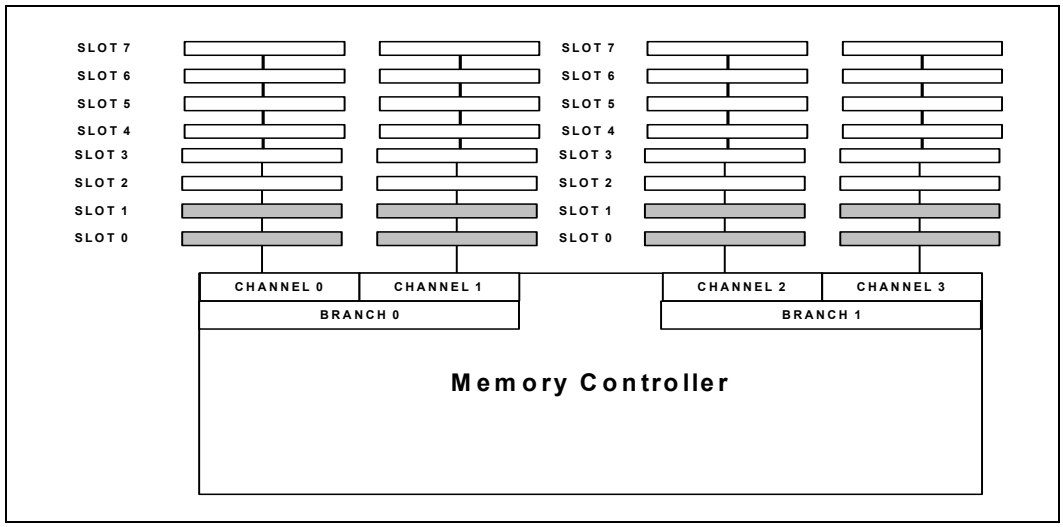


Figure 7-8 shows the positions of the next four DIMM upgrade. Like non-mirrored mode upgrade DIMMs must be added in slot order, starting from the slot closest to the MCH. DIMMs in a slot position must be identical with respect to size, and organization. Speed should be matched but is not required. The MCH will adjust to the lowest speed DIMM. DIMMs in adjacent slots need not be identical.

Figure 7-8. Mirrored Mode Next Upgrade



7.3.6 ECC Code

Lock-Step Mode (Mirrored and Non-mirrored Modes):

When branches operate in lock-step mode (mirrored and non-mirrored mode), Intel® 7300 Chipset supports the 18 device DRAM failure correction code option for FBD. As applied by the MCH , this code has the following properties:

- 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 MCH 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 two-wire faults on the DIMM's. This includes any pair of single bit errors.
- Detection of all permutations of 2 x4 DRAM failures.

Single Channel Mode:

When the branch operates in single-channel mode, the MCH supports an 8-byte-over-32-byte SECDED+ code. It is the same ECC code used in lockstep-channel mode, but the SDDC properties degrade to SECDED+ because the number of devices over which the codeword is distributed is halved. As applied by the MCH, this code has the following properties:

- Detection of any 2-bits in error within 8 bytes of data
- Correction of any single bit in error within 8 bytes of data

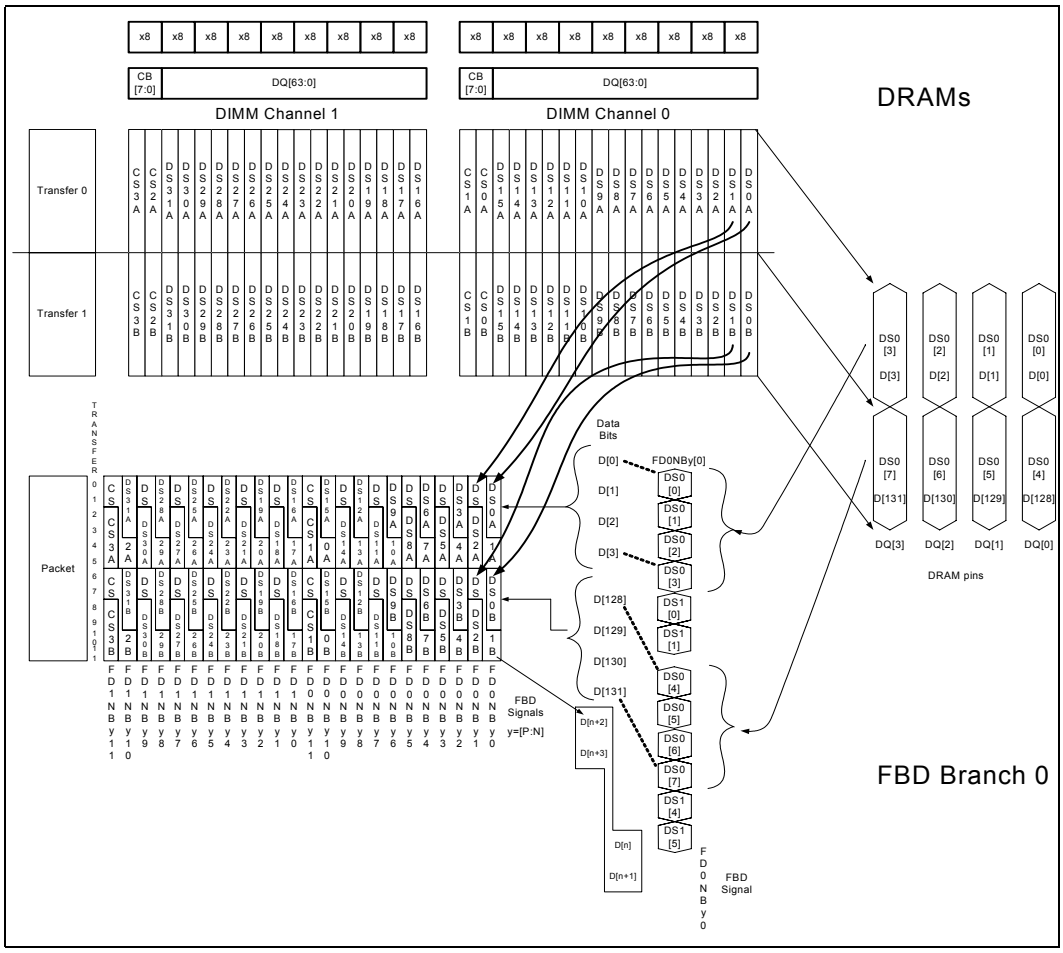
7.3.6.1 Inbound ECC Code Layout for Lockstep Branches

The code is systematic: i.e., the data is separated from the check-bits rather than all being encoded together. It consists of 32 eight-bit data symbols (DS31-DS0) and four eight-bit Check-bit Symbols (CS3-CS0). The code corrects any two adjacent symbols in error. 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 7-9 illustrates the ECC code layout for branch 0. The figure shows how the symbols are mapped on the FBD branch and to DRAM bits by the DIMM for a transfer in which the critical 16B is in the lower half of the code-word (A[4]=0). If the upper portion of the code-word were transferred first, bits[7:4] of each symbol would be transferred first on the DRAM interface and in the first six transfers on the FBD channel. The layout for branch 1 is the same.

The bits of Data Symbol 0 (DS0) are traced from DRAM to FBD Northbound. The same mapping of symbols to data and code bits applies to Southbound data. The lower nibble (DS0A) consists of DS0[3:0] 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] 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. The same DS0 is shown expanded on the Northbound FBD interface where it occupies the FD0NB[P:N][0] signal. DS0 and DS1 cover all transfers on FD0NB[P:N][0] (even though FD0NB[P:N][0] does not cover all of DS1).

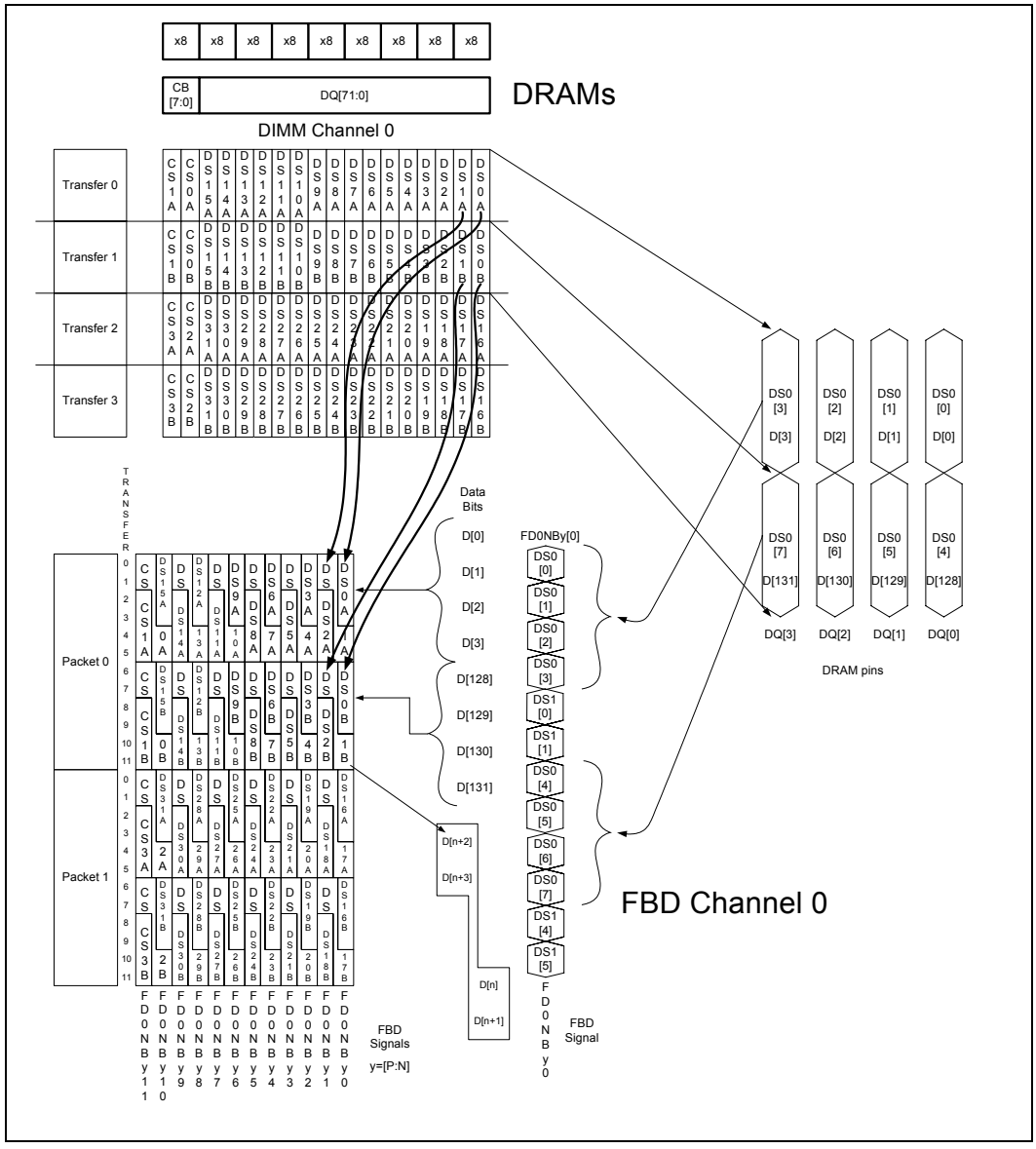
Figure 7-9. Code Layout for Lockstep Branches



7.3.6.2 ECC Code Layout for a Single-Channel Branch

The ninth byte of each burst on each DIMM contains the ECC bits for 8 bytes of data. These nine bytes comprise a code word. There are eight code words in a cache line.

Figure 7-10. Code Layout for the Single-Channel Branch



7.3.7 Single Device Data Correction (SDDC) Support

The Intel® 7300 Chipset chipset employs a single device data correction (SDDC) algorithm for the memory subsystem that will recover from a x4/x8 component failure. The chip disable is a 32-byte two-phase code. SDDC is also supported for x4 devices. In addition the MCH supports demand and patrol scrubbing.

A scrub corrects a correctable error in memory or poisons an uncorrectable 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.

The attributes of the SDDC code are as follows:

- Two Phase Code over 32 bytes of data.
- 100% Correction for all single x4 or x8 component failures.
- 100% Detection of all double x4 component failures.
- Detection Characteristics for x8 double device errors are provided in the Table 7-6

Table 7-6. x8 Double Device Detection Characteristics

Overall coverage - 99.986%	Device plus single - 99.99999%
Double bit errors - 100%	Device plus wire - 99.99998%
Double wire faults - 100%	Device plus equal/phase - 99.9998%
Wire plus single bit - 100%	Equal/phase plus equal/phase - 100%

To increase the detection coverage of a (x8 device failure + SBE), i.e. to avoid silent data corruption in the event of a particle induced error while correcting for a failed device, the MCH provides the following features:

- Each rank will have an encoded value of the "failed" x8 component or pair of x4 components.
- If for any given rank, the MCH detects a correctable error with a weight >1 and the "corrected" symbol does not match the "failed" component, the MCH will assume that the error is multi-bit uncorrectable error and signal a "fatal error".

7.3.8 x8 Correction

7.3.8.1 Normal

This correction mode is in effect when the MC.SCRBALGO configuration bit is cleared. An erroneous read will be logged. If the ECC was correctable, it is corrected (scrubbed) in memory (Demand scrub needs to be enabled for scrubbing. See Section 7.3.9, "Demand Scrubbing"). A conflicting read or write request pending issue will be held until the scrub is either completed or aborted because it was uncorrectable.

7.3.8.2 Enhanced

This correction mode is in effect when the MC.SCRBALGO configuration bit is set and software has initialized the MC.BADRAMTH to a non-zero value.

- Maintain 4-bit saturating counters per rank in the BADCNT configuration registers.
  Floor at zero. Saturate at the value of the MC.BADRAMTH configuration register field.
  Increment on correctable errors on both symbols of a x8 device and Northbound CRC OK.
  Decrement upon completion of the number of patrol scrub cycles through the entire memory specified by the MC.BADRAMTH configuration register field. A sufficient resolution of this period is three patrol scrub cycles through all memory.

- Maintain five-bit bad-device marks per rank in the BADRAM(A/B) configuration registers.

  Upon incrementing BADCNT to saturation, then mark the bad devices in the BADRAM(A/B) configuration registers.
- A correctable ECC in a symbol other than that marked in the BADRAM(A/B) configuration registers is an aliased uncorrectable read.

An erroneous read will be logged. If the read was correctable, it is corrected (scrubbed) in memory. (Demand scrub needs to be enabled for scrubbing. See Section 7.3.9, "Demand Scrubbing"). A conflicting read or write request remains pending until the scrub succeeds or is dropped. A failed scrub is replayed once, resulting in success or a drop.

7.3.9 Demand Scrubbing

A demand scrub corrects a correctable error in memory or poisons an uncorrectable error in memory. To enable this function, the MC.DEMSEN configuration bit must be set. Correctable read data will be corrected to the requestor and scrubbed in memory. This adds an extra cycle of latency to accomplish the correction. Error logs include RAS/CAS/BANK/RANK.

Demand scrubbing is not available in mirrored mode (when MC.MIRROR is set). If a correctable error is encountered, the data is corrected and sent to the requestor at the cost of one extra cycle of latency. The probability of soft errors due to alpha rays affecting multiple x4/x8 devices is low. However, patrol scrubbing when enabled in the MCH for the Mirrored mode will clean up all correctable errors in the memory running in the background and runs twice as fast. There is no incurred RAS benefit by enforcing demand scrubbing in mirrored mode with the exception of the error logging. Demand scrubbing does not help in failed ECC case (uncorrectable errors). i.e. If the data read is uncorrectable from the bad branch, then the golden data needs to be retrieved from the other mirrored branch (copy) at the cost of additional FB-DIMM reset, link training and DDR protocol rules. The failed branch is offlined and needs to be replaced for mirroring to continue.

Either Demand or Patrol scrub can in process.

7.3.10 Patrol Scrubbing

To enable this function, the MC.SCRBEN configuration bit must be set. The scrub unit starts at DIMM Rank 0 / Address 0 upon reset. Every 16 k core cycles the unit will scrub one cache line and then increment the address one cache line provided that back pressure or other internal dependencies (queueing, conflicts, etc.) do not prolong the issuing of these transactions to FB-DIMM. Using this method, roughly 64 GBytes of memory behind the MCH can be completely scrubbed every day (estimate). Error logs include RAS/CAS/BANK/RANK. Patrol scrub writes hit both branches in mirrored mode (when MC.MIRROR is set). Normally, one branch is scrubbed in entirety before proceeding to the other branch. In the instance of a fail-down to non-redundant operation that off-lines the branch that was being scrubbed, the scrub pointer merely migrates to the other branch without being cleared. In this unique instance, the scrub cycles for that branch is incomplete.

7.3.11 Data Poisoning in Memory

Data Poisoning in memory is defined as all zeroes in the code word (32B) except for the 3 least significant bytes being 0xFF00FF. The Intel® 7300 Chipset MCH poisons a memory location based on the events described in Table 7-7

Table 7-7. Memory Poisoning Table

Event	Correctable Error	UnCorrectable Error
Normal Memory Read	Correct Data to be given to requester. the MCH logs M17 error. (Correctable Non-Mirrored demand data ECC Error) Correct Data to be written back to memory. (Demand Scrub needs to be enabled)	Detects an Uncorrectable and logs a M9 error. (Non-aliased uncorrectable non-mirrored demand data ECC error) Re-Issue Read to Memory. If error persistent 1. Poison the response to requester and log. 2. Leave data untouched in memory location.
Patrol Scrub	Correct Data to be written back to memory and log M20 error. (Correctable patrolled data ECC error)	1. Log and Signal M12 Error (Non-Aliased uncorrectable patrol data ECC error). 2: Leave data untouched in memory location.
DIMM Spare Copy	Correct Data to be written back to memory and log M19 error. (Correctable spare copy data ECC error)	If error persistent 1. Log and Signal M11 Error (Non Aliased uncorrectable spare copy data ECC error). 2: Poison Location in DIMM Spare
Mirror Copy	Correct Data to be written to new memory and log M19 error. (Correctable spare copy data ECC error)	Re-use Read to memory and signal a M11 error. (Non-aliased uncorrectable spare copy data ECC error). If error persistent 1. Poison the new memory image.

7.3.12 Memory Mirroring

Refer to Section 7.3.4.3, “Mirrored Mode Operation” on page 361

7.3.13 DIMM Sparing

7.3.13.1 Normal (Automatic) DIMM Sparing

At configuration time, a DIMM rank is set aside to replace a defective DIMM rank. It is important to understand that in the context of this specification, “DIMM rank” does not refer to the physical rank on a single FBDIMM. Rather, “DIMM rank” refers to a set of DRAM devices that supply a cacheline of data. Commonly, this “DIMM rank” will be a pair of FBDIMMS being accessed in lockstep.

Counters are used to implement a “leaky bucket” algorithm, described here. When any of the counters associated with a failing DIMM rank reaches a pre-determined threshold, the SPCPS.LBTHR configuration bit will issue an interrupt and initiate a spare copy. While the copy engine is line-atomically reading locations from the failing DIMM rank and writing them to the spare, system reads will be serviced from the failing DIMM rank, and system writes will be written to both the failing DIMM rank and the spare DIMM rank.

At the completion of the copy, the failing DIMM rank is disabled and the “spared” DIMM rank will be used in its place. The MCH will change the rank numbers in the DMIRs from the failing rank to the spare rank. DMIR.LIMIT’s are not updated.

This mechanism requires no software support once it has been enabled by designating the spare rank through the SPCPC.SPRANK configuration register field and enabling sparing by setting the SPCPC.SPAREN configuration bit. Hardware will detect the

threshold-initiated fail-over, accomplish the copy, and off-line the "failed" DIMM rank once the copy has completed. This is accomplished autonomously by the memory control subsystem. The SPCPS.SFO configuration bit is set and an interrupt is issued indicating that a sparing event has completed.

Sparing cannot be invoked while operating a mirrored memory configuration. Sparing to a smaller DIMM is not supported.

Note: DIMM sparing is not supported in the Single Channel Mode when MCA.SCHDIMM is set.

7.3.13.2 Manual DIMM Sparing

Sparing can also be started manually, by setting the SPCPC.FORCE bit.

7.3.14 FB-DIMM Power Management

Channels which are not connected to a DIMM or connected to a failed DIMM can be held in a suspended state by setting FBDHPC.NEXTSTATE to Reset. This corresponds to the Disable state in the FBD specification. This will hold the FBD channel in reset which disables drivers and receivers.

7.3.15 FBD Throttling

The Intel® 7300 Chipset Memory controller implements an adaptive throttling methodology along with electrical throttling to limit the number of memory requests to the FB-DIMMs. The methodology is comprised of the following:

1. Activation throttling: Consists of closed/open loop throttling of activates on the FB-DIMM.
   a. Open Loop Thermal Throttling (OLTT) to limit requests when the number of activates crosses an event threshold events in large time window
   b. Static Closed Loop Thermal Throttling (S-CLTT) when the temperature of the FB-DIMMs increases beyond a certain threshold.
2. Electrical Throttling to prevent silent data corruption by limiting the number of activates per rank in a small sliding window

Thermal throttling window: A window consisting of 1,344 cycles, during which throttling is applied. The various throttling thresholds defined in Section 7.3.15.3 are for a thermal throttling window

Global throttling window (GTW): The Global throttling window is defined as an integral multiple of 1344 clocks. Under normal operating conditions, the Global Throttling Window duration is set to 16384*1344 bus clocks which gives a global throttling window of 82.58ms (266 Mhz bus clock). However, for purposes of validation and debug, the global throttling window can be reduced to a smaller duration viz. 4*1344 cycles[1] by setting the GTW_MODE register bit defined in Section 4.8.18.1, "MC: Memory Control Settings" on page 217

7.3.15.1 Open Loop Thermal Throttling (OLTT)

The MCH works in OLTT mode if the THRTCTRL.THRMHUNT bit is not set. See Section 4.8.18.7, "THRTCTRL: Thermal Throttling Control Register"

1. If MC.GTW_MODE=1, the MCH will use the 4*1344 cycle duration for the global throttling window.

In the OLTT scheme, the number of activates per DIMM pair per branch is counted for a Global Throttling window. If the number exceeds the number indicated by the GBLACT.GBLACTLM[1] field defined in Section 4.8.18.2, "GBLACT: Global Activation Throttle Register" on page 219 then the THRTSTS[1:0].GBLTHRT bit is set for the respective branch at the end of the current Global Throttling window, and the branch is throttled to the throttle settings in THRTMID register (Section 4.8.18.5, "THRTMID: Thermal Throttle Mid Register").

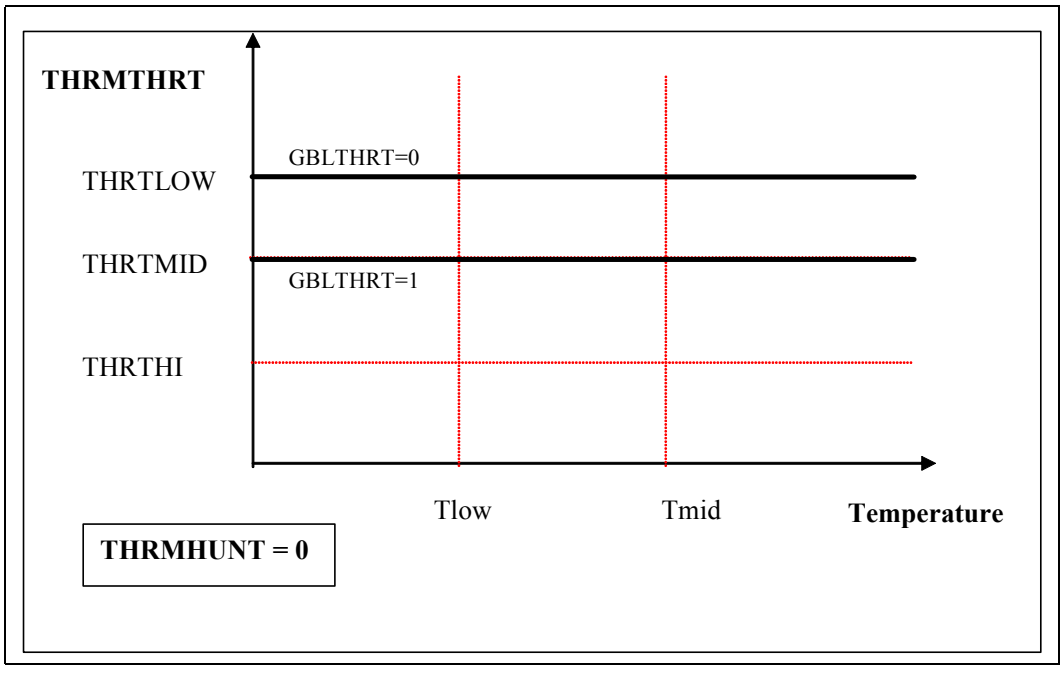
Throttling will remain active until 16 global throttling windows in a row have gone by without any DIMM exceeding the GBLACT. At the end of the 16 global throttling windows, if no DIMM pair activates exceed the GBLACT.GBLACTLM value, then the MCH indicates the end of the period by clearing the THRTSTS[1:0].GBLTHRT register field. If part way through the count of 16 global throttling windows, the GBLACT.GBLACTLM is again exceeded within one Global Throttle Window, the counter gets reset and it will once again count 16 Global throttle windows throttling at the THRTMID level.

Note: If MC.GTW_MODE is set (validation mode), then 2 global throttling windows will be used a limit instead of 16 global throttling windows as described above.

Using the global throttling window prevents short peaks in bandwidth from causing throttling when there has not been sufficient DRAM activity over a long period of time to warrant throttling. It is in effect a low pass filter on throttling.

In OLTT, the temperature of the DIMMs has no impact on throttling as shown in Figure 7-11

Figure 7-11. Open Loop Thermal Throttling (THRTCTRL.THRMHUNT = 0)



1. The GBLACT is an 8-bit register, with a granularity of either 16 or 65536 activates depending on the MC.GTW_MODE bit.

7.3.15.2 Static Closed Loop Thermal Throttling (S-CLTT)

To enable S-CLTT mode the THRTCTRL.THRMHUNT bit must be set.

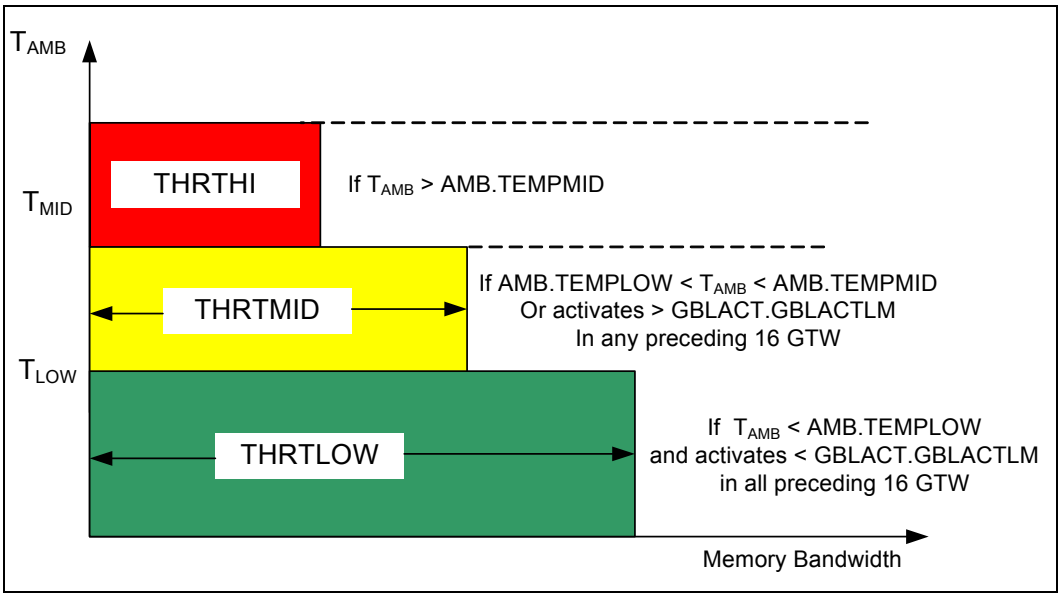
In the S-CLTT scheme, the temperature of the DIMMs is read by the temperature sensor located in the AMB and is fed back to the Intel® 7300 Chipset MCH. The throttling logic in the MCH uses this information to limit the number of activates to any DIMM within a throttling window.

Every 42 frames the host controller is required to send a sync packet, which returns a status packet from the AMBs. The temperature related information is returned in the status packet which the MCH uses for throttling adjustment. The AMB component has two temperature threshold points, T_{low} (programmed into the TEMPLO register) and T_{mid} (programmed into the TEMPMID register), and the current temperature of the AMB with respect to these thresholds are returned in the status packet.

The level of throttling applied in S-CLTT depends on the temperature information returned in the status packet and the throttling thresholds programmed into the threshold registers described in Section 7.3.15.3. See Figure 7-12 for a graphical representation of S-CLTT.

Note: Even in S-CLTT mode, OLTT will get turned on if the number of activates for any DIMM pair in any of the preceding 8 Global throttling windows is greater than the value in the GBLACT.GBLACTLM field (even when temperature returned s less than T_{low}). THRTMID level of throttling is applied in this case, and the MCH will set THRTSTS[1:0].GBLTHRT.

Figure 7-12. Static Closed Loop Thermal Throttling (THRTCTRL.THRMHUNT = 1)



7.3.15.3 Throttling Thresholds for Closed/Open Loop Thermal Throttling.

The current throttling parameters for each branch are stored in the THRMTHRT field defined in Section 4.8.18.3, "THRTSTS[1:0]: Throttling Status Register" . Three levels of throttling limits are defined, and are for a Thermal throttling window.

- THRTLOW: A base throttling level for both OLTT And S-CLTT. It is applied when the temperature is in the low range (below T_{low}) and the THRTSTS.GBLTHRT bit is not set by the MCH (number of activates for any DIMM pair in any of the preceding 8 Global throttling windows is less than the value in the GBLACT.GBLACTLM field). See Section 4.8.18.4, "THRTLOW: Thermal Throttle Low Register"

- THRTMID: Mid throttling level for S-CLTT and highest throttling level for OLTT. This is applied when the temperature is in the middle range (above T_{low} but below T_{mid}) or the THRSTS.GBLTHRT bit is set by the MCH (number of activates for any DIMM pair in any of the preceding 8 Global throttling windows is greater than the value in the GBLACT.GBLACTLM field). See Section 4.8.18.5, "THRTMID: Thermal Throttle Mid Register"
- THRTHI: The highest level of throttling for CLTT. This is applied when the temperature is above T_{mid}. See Section 4.8.18.6, "THRTHI: Thermal Throttle High Register"

7.3.15.4 Closed/Open Loop Throttling Policy

There are separate counters associated with each of the 8 lockstep DIMM pairs in a given branch. When any of the counters reaches its throttling limit (as specified by the THRTSTS[1:0].THRMTHRT register field, see Section 4.8.18.3, "THRTSTS[1:0]: Throttling Status Register"), the entire branch is throttled until the end of the throttle window. No new DRAM commands are issued to any of the DIMMs on the branch until the end of the throttle window. If an activate has been issued to a bank, the follow on read or write may be issued, including an additional page hit access if applicable, allowing the page to close.

7.3.15.5 Electrical Throttling

Electrical throttling is a mechanism that limits the number of activates within a very short time interval that would otherwise cause silent data corruption on the DIMMs. Electrical throttling is enabled by setting the MTR.ETHROTTLE bit defined in Section 4.8.23.7, "MTR[1:0][7:0]: Memory Technology Registers" on page 250. These bits occur on a per DIMM pair basis per branch as to whether electrical throttling should be used. It is assumed that both ranks within a DIMM would be the same technology, and therefore does need not separate enable bits.

The MC.ETHROT configuration register field is used by the Intel® 7300 Chipset to limit the number of activations per sliding electrical throttle window

The per rank electrical throttling for FB-DIMM is set at 4 activates per 37.5 ns window (JEDEC consensus) and is summarized in Table 7-8 on page 374 for various DIMM technologies.For DDR2, the number of activates per tRC (RAS cycle time) is around 4*tRC/tFAW=4*60/37.5=6.4. This tFAW limit helps limit cost in three ways:

1. size of DRAM on die voltage regulator
2. amount of DIMM decoupling
3. Size and slew rate of Vdd Voltage Regulator (in systems with one DIMM per channel.

Table 7-8. Electrical Throttle Window as a Function of DIMM Technology and Core: FBD Gear Ratios

DIMM Modes	Intel® 7300 Chipset Core:FBD clock ratio	Electrical Throttle Window[a] (the MCH clocks per rank per DIMM pair per branch as a function of core to FBD clock ratio)
DDR533/667	1:1	10
Conservative (safe mode)	All	20

Notes:

a. Maximum 4 activates per rank is allowed within the window.

7.3.16 FB-DIMM IBIST Support

FB-DIMM IBIST support is covered in Section 10.2.2.

7.4 Interrupts

The Intel® 7300 Chipset chipset supports both the XAPIC and traditional 8259 methods of interrupt delivery. I/O interrupts and inter processor interrupts (IPIs) appear as write or interrupt transactions in the system and are delivered to the target processor via the processor bus. This chipset does not support the three-wire sideband bus (the APIC bus) that is used by Pentium® and Pentium® Pro processors.

XAPIC interrupts that are generated from I/O will need to go through an I/O(x)APIC device unless they support Message Signalled Interrupts (MSI). In this document, I/O(x)APIC is an interrupt controller that is found in the Intel® 631xESB/632xESB I/O Controller Hub component of the chipset.

The legacy 8259 functionality is embedded in the Intel® 631xESB/632xESB I/O Controller Hub component. The MCH will support inband 8259 interrupt messages from PCI Express devices for boot. The chipset also supports the processor generated "interrupt acknowledge" (for legacy 8259 interrupts), and "end-of-interrupt" transactions (XAPIC).

Routing and delivery of interrupt messages and special transactions are described in this section.

7.4.1 XAPIC Interrupt Message Delivery

The XAPIC interrupt architectures deliver interrupts to the target processor core via interrupt messages presented on the front side bus. This section describes how messages are routed and delivered in a Intel® 7300 Chipset system, this description includes interrupt redirection.

Interrupts can originate from I/O(x)APIC devices or processors in the system. Interrupts generated by I/O(x)APIC devices occur in the form of writes with a specific address encoding. Interrupts generated by the processor appear on the processor bus as transactions with a similar address encoding, and a specific encoding on the REQa/REQb signals (REQa=01001, REQb=11100).

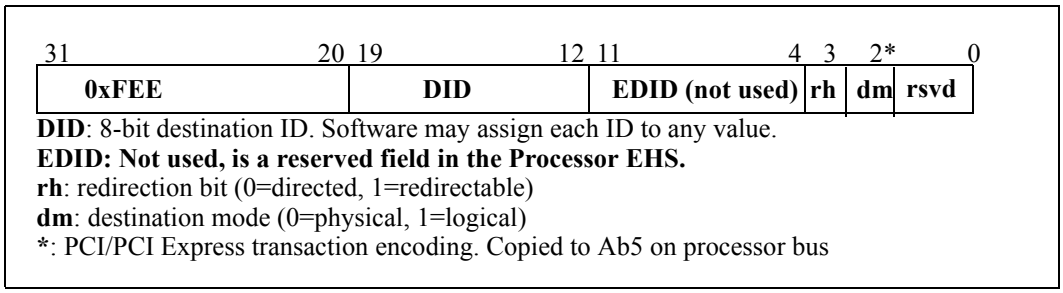
The XAPIC architecture provides for lowest priority delivery, through interrupt redirection by the chipset. If the redirectable hint bit is set in the XAPIC message, the chipset may redirect the interrupt to another processor. Note that redirection of interrupts can be to any processor on either Processor Bus ID and can be applied to both I/O interrupts and IPIs. The redirection can be performed in logical and physical destination modes. For more details on the interrupt redirection algorithm, see Section 7.4.1.4, "Interrupt Redirection" .

7.4.1.1 XAPIC Interrupt Message Format

Interrupt messages have an address of 0x000_FEEz_zzzY. The 16-bit "zzzz" field (destination field) determines the target to which the interrupt is being sent. The Y field is mapped to A3 (redirectable interrupt) and A2 (destination mode). Figure 5-18 shows the address definition in IA-32 systems (XAPIC). For each interrupt there is only one data transfer. The data associated with the interrupt message specifies the interrupt vector, destination mode, delivery status, and trigger mode. The transaction type on the processor bus is a request type of, interrupt transaction. The transaction type on

the PCI Express and ESI buses is a write. The address definition of Figure 5-18 applies to both the PCI Express bus and processor bus. Note that the current assumption is that we can't make any conclusions about which FSB an interrupt ID is associated with. At power-up, there is an association for certain types of interrupts, but the current assumption is that the OS can reprogram the interrupt ID's. Therefore, for directed interrupts, the MCH will ensure that each interrupt is seen on both FSBs.

Figure 7-13. XAPIC Address Encoding



DID: 8-bit destination ID. Software may assign each ID to any value.
EDID: Not used, is a reserved field in the Processor EHS.
rh: redirection bit (0=directed, 1=redirectable)
dm: destination mode (0=physical, 1=logical)
*: PCI/PCI Express transaction encoding. Copied to Ab5 on processor bus

The data fields of an interrupt transaction are defined by the processor and XAPIC specifications. It is included here for reference.

Table 7-9. XAPIC Data Encoding

D[63:16]	D[15]	D[14]	D[13:11]	D[10:8]	D[7:0]
x	Trigger Mode	Delivery Status	x	Delivery Mode	Vector

7.4.1.2 XAPIC Destination Modes

The destination mode refers to how the processor interprets the destination field of the interrupt message. There are two types of destination modes; physical destination mode, and logical destination mode. The destination mode is selected by A[2] in PCI Express and Ab[5] on the processor bus.

Physical Destination Mode (XAPIC)

In physical mode, the APIC ID is 8 bits, supporting up to 255 agents. Each processor has a Local APIC ID Register where the lower 5 bits are initialized by hardware (Cluster ID=ID[4:3], Bus Agent ID=ID[2:1], thread ID=ID[0]). The upper 3 bits default to 0's at system reset. By default, the MCH will drive A[12:11] to `00/`01/`10/`11 for FSB0/1/2/3, though default values can be modified through the POC.APICID1 field (See Section 4.8.6.3, "POC: Power-On Configuration").

The MCH will not rely on the cluster ID or any other fields in the APIC ID to route interrupts. The MCH will ensure the interrupt is seen on all busses and the processor with the matching APIC ID will claim the interrupt.

Physical destination mode interrupts can be directed, broadcast, or redirected. An XAPIC message with a destination field of all 1's denotes a broadcast to all.

In a directed physical mode message the agent claims the interrupt if the upper 8 bits of the destination field (DID field) matches the Local APIC ID of the processor or the interrupt is a broadcast interrupt.

Redirected interrupts are redirected and converted to a directed interrupt by the chipset as described in Section 7.4.1.6.

Logical Destination Mode (XAPIC)

In logical destination mode, destinations are specified using an 8 bit logical ID field. Each processor contains a register called the Logical Destination Register (LDR) that holds this 8-bit logical ID. Interpretation of the LDR is determined by the contents of the processor's Destination Format Register (DFR). The DFR establishes if the processor is in flat or cluster mode. Logical destination mode interrupts can be directed (fixed delivery), redirectable (lowest priority delivery), or broadcast. The LDR is initialized to flat mode (0) at reset and is programmed by firmware (and can be changed by system software). How the MCH determines if the system is in flat or cluster mode is described in Section 7.4.1.5.

Interpretation of the destination field of the XAPIC message and the logical ID of the processor is different whether the processor is in flat or cluster mode. The 2 models are flat and cluster.

Flat Model - The 8-bit logical ID is compared to the 8-bit destination field of the incoming interrupt message. If there is a bit-wise match, then the local XAPIC is selected as a destination of the interrupt. Each bit position in the destination field corresponds to an individual Local XAPIC Unit. The flat model supports up to 8 agents in the system. An XAPIC message where the DID (destination field) is all 1's is a broadcast interrupt.

Cluster Model - The 8-bit logical ID is divided into two fields, where the upper four bits contain a cluster ID, and the lower 4-bits identifies the agent within the cluster using the same encoding scheme as the flat model. Up to 16 logical clusters (4 agents each) can be supported. An XAPIC message where the lower 4-bits of the DID is all 1's is a broadcast interrupt within a cluster. Interrupt redirection can only be applied to processors within the cluster.

7.4.1.3 XAPIC Interrupt Routing

Interrupt messages that originate from I/O(x)APIC devices or from processing nodes must be routed and delivered to the target agents in the system. In general XAPIC messages are delivered to all four processor busses because there is no reliable way to determine the destination processor of the message from the destination field. Interrupts originating from I/O can be generated from a PCI agent using MSI interrupts, or by an interrupt controller on a bridge chip such as the Intel® 631xESB/632xESB I/O Controller Hub. Table 7-10 shows the routing rules used for routing XAPIC messages in a Intel® 7300 Chipset based platform. This table is valid for both broadcast and non-broadcast interrupts.

Table 7-10. Intel® 7300 Chipset XAPIC Interrupt Message Routing and Delivery

Source	Type	Routing
I/O	physical or logical directed	Deliver to all processor busses as an interrupt transaction.
Processor	physical or logical directed	Deliver to other processor bus as an interrupt transaction.
Any Source	logical, redirectable physical, redirectable	Redirection (see Section 7.4.1.4, "Interrupt Redirection") is performed by the MCH and is delivered to all FSBs

7.4.1.4 Interrupt Redirection

The XAPIC architecture provides for lowest priority delivery through interrupt redirection by the MCH If the redirectable “hint bit” is set in the XAPIC message, the chipset may redirect the interrupt to another agent. Redirection of interrupts can be applied to both I/O interrupts and IPIs.

7.4.1.5 LAPIC State Registers

To accomplish redirection, the Intel® 7300 Chipset implements a set of 32 LAPIC State registers, one for each logical processor (a thread is considered a logical processor). The MCH is built to support up to 4 Octal core processors. Each register contains the following fields:

1. APIC enable bit (TPR Enable)
2. logical APIC ID (LOGID)
3. processor physical APIC ID (PHYSID)

Register 0 alone contains a bit (Cluster Mode) that tracks the cluster vs. flat mode of the system globally.

The LAPIC State registers are modified by a front side bus xTPR_Update transaction. All the fields in an LAPIC State register are updated on the corresponding xTPR_Update transaction as-is, except the Cluster Mode bit in register 0 which is described below. In addition, the LAPIC State registers can be modified by software.

Changing Cluster Mode

The Cluster Mode bit in LAPIC register 0 defaults to 0 indicating flat mode. On each xTPR_Update transaction if the following criteria are met, then the Cluster Mode bit in the transaction is used to update the Cluster Mode bit in register 0.

1. The APIC enabled bit being set in the message AND
2. The Logical APICID field is non-zero AND
3. REDIRCTRL.Check_APICID is set (bit 15).

Otherwise the Cluster Mode bit in register 0 remains unchanged. The MCH redirection logic for logical mode interrupts uses this Cluster Mode bit in register 0 to appropriately redirect logical interrupts.

If all conditions above are met, the MCH samples the flat/cluster bit in the message and updates the global bit in LAPIC_STATE[0] entry. The global bit determines if the MCH handles logical mode interrupts as flat mode interrupts or cluster mode interrupts.

Note: INTRSTS bit 1 will be set if the MCH receives an XTPR message with APIC enable bit set and the Logical APICID field is all 0s.

For more details on the MCH LAPIC State registers refer to Section 7.4.1.5, “LAPIC State Registers” and Section 4.8.5, “Interrupt Redirection Registers”

The XTPR special cycle must guarantee that the LAPIC State register is updated for interrupt redirection in a consistent manner. For reproducibility, there needs to be an internal serialization point after which subsequent interrupts will be redirected based on the updated LAPIC State value.

7.4.1.6 Redirection Algorithm

1. If A[3] =1, then this is a redirection (also known as "lowest priority") interrupt request. Proceed to the next step.
2. FLAT: If Destination Mode = 1 (A2 for I/O, Ab5 for IPI's) and the MCH .LAPIC_STATE[0].CLUSTER_Mode is disabled (0), then this is Flat-Logical Destination Mode.
3. CLUSTER: If Destination Mode = 1 (A2 for I/O, Ab5# for IPI's) and the MCH .LAPIC_STATE[0].CLUSTER is enabled (1), then this is Clustered-Logical Destination Mode.
4. PHYSICAL: If Destination Mode = 0 (A2 for I/O, Ab5 for IPI's), then this is Physical Destination Mode. All enabled xTPR's are included in the arbitration pool.

Redirection is performed if an interrupt redirection hint bit (A[3]) is set. This is the algorithm used in determining the processor that the interrupt will be redirected to.

Physical Mode Interrupt (IPIs and I/O interrupts)

- If redirection hint bit is clear, just broadcast the interrupt to all FSB's (NOT including source FSB in case of IPIs)
- If redirect hint bit is set and the APICID indicated in the interrupt is enabled, simply clear the hint bit and broadcast to all FSB's (including source FSB in case of IPIs)

Logical Flat Mode Interrupt (both IPI's and I/O interrupts)

- If redirection hint bit is clear, just broadcast the interrupt to all FSB's (NOT including source FSB in case of IPIs)
- If redirection hint bit is set, pick one "enabled" agent from the list of N (N <=8) agents indicated in the interrupt using an LRU mechanism, modify the interrupt message so that only the winning agent's bit is set in the interrupt message and then broadcast the interrupt to all FSB's. (including source FSB in case of IPIs)
  - 'Enabled' agent is one for which the corresponding bit in the interrupt message is set AND the APIC enable bit in the corresponding LAPIC State table is set

Logical Cluster Mode Interrupt (both IPI's and I/O interrupts)

- If redirection hint bit is clear, just broadcast the interrupt to all FSB's (NOT including source FSB in case of IPIs)
- If redirection hint bit is set, pick one "enabled" agent from the list of N (N <=4) agents in the specified "cluster" in the interrupt, using an LRU mechanism, modify the interrupt message such that the winning agent's Cluster ID and its Logical-ProcessorID-within-the-cluster is indicated and then broadcast the interrupt to all FSB's. (including source FSB in case of IPIs)
  - Enabled' agent is one for which the corresponding bit in the interrupt message is set AND the APIC enable bit in the corresponding LAPIC State table is set
  - The LRU mechanism for cluster interrupts needs 15 4-way LRU trackers (compared to the 1 8-way LRU tracker for flat mode interrupts)

7.4.1.7 EOI

For Intel® Xeon® processor based platforms using XAPIC, the EOI is a specially encoded processor bus transaction with the interrupt vector attached. Since the EOI is not directed, the MCH will broadcast the EOI transaction to all I/O(x)APIC's. The MCH PEXCTRL.DIS_APIC_EOI bit per PCI Express port can be used to determine whether an EOI needs to be sent to a specific PCI Express port. EOI usage is further described in Section 7.4.2.2.

Note: The MCH will translate the EOI on the FSB into an EOI TLP message type on the PCI Express/ESI ports.

7.4.2 I/O Interrupts

Intel® 631xESB/632xESB I/O Controller Hub receives I/O interrupts with either dedicated interrupt pins or with writes to the integrated interrupt redirection table. The I/OxAPIC controllers integrated within the Intel® 631xESB/632xESB I/O Controller Hub turns these interrupts into writes destined for the processor bus with a specific address.

Interrupts triggered from an I/O device can be triggered with either a dedicated interrupt pin or through an inbound write message from the PCI Express bus (MSI). Note that if the interrupt is triggered by a dedicated pin, the I/OxAPIC controllers in the Intel® 631xESB/632xESB I/O Controller Hub turns this into an inbound write. On the processor bus, the interrupt is converted to an interrupt request. Other than a special interrupt encoding, the processor bus interrupt follows the same format as discussed in Section 7.4.1.1, “XAPIC Interrupt Message Format” . Therefore, to all components other than the Intel® 631xESB/632xESB I/O Controller Hub, and the processor, an interrupt is an inbound write following the format mentioned in Section 7.4.1.1. The MCH will not write combine or cache the APIC address space.

I/O(x)APIC’s can be configured through two mechanisms. The traditional mechanism is the hard coded FEC0_0000 to FECF_FFFF range is used to communicate with the IOAPIC controllers in the Intel® 631xESB/632xESB I/O Controller Hub.

The second method is to use the standard MMIO range to communicate to the PXH (in the Intel® 631xESB/632xESB I/O Controller Hub). To accomplish this, the PXH.MBAR and/or must be programmed within the PCI Express device MMIO region.

7.4.2.1 Ordering

Handling interrupts as inbound writes has inherent advantages. First, there is no need for the additional APIC bus resulting in extra pins and board routing concerns. Second, with an out-of-band APIC bus, there are ordering concerns. Any interrupt needs to be ordered correctly and all prior inbound writes must get flushed ahead of the interrupt. The *PCI Local Bus Specification*, Revision 2.2 attempts to address this by requiring all interrupt routines to first read the PCI interrupt register. Since PCI read completions are required to push all writes ahead of it, then all writes prior to the interrupt are guaranteed to be flushed. However, this assumes that all drivers perform this read.

7.4.2.2 Hardware IRQ IOxAPIC Interrupts

Dedicated pin interrupts may be edge or level triggered. They are routed to IRQ pins on IOxAPIC device such as the PXH, or Intel® 631xESB/632xESB I/O Controller Hub. The IOxAPIC device will convert the interrupt into either an XAPIC or 8259 interrupt.

For level-triggered interrupts, the I/OxAPIC will generate an interrupt message when any of the interrupt lines coming into it become asserted. The processor will handle the interrupt and eventually write to the initiating device that the interrupt is complete. The device will deassert the interrupt line to the I/OxAPIC. After the interrupt has been serviced, the processor sends an EOI command to inform the I/OxAPIC that the interrupt has been serviced. Since the EOI is not directed, the MCH will broadcast the EOI transaction to all I/O(x)APIC’s. If the original I/O(x)APIC sees the interrupt is still asserted, it knows there’s another interrupt (shared interrupts) and will send another interrupt message.

For edge-triggered interrupts, the flow is the same except that there is no EOI message indicating that the interrupt is complete. Since the interrupt is issued whenever an edge is detected, EOIs are not necessary.

While not recommended, agents can share interrupts to better utilize each interrupt (implying level-triggered interrupts). Due to ordering constraints, agents can not use an interrupt controller that resides on a different PCI bus. Therefore either only agents on the same PCI bus can share interrupts, or the driver MUST follow the PCI requirement that interrupt routines must first read the PCI interrupt register

The MCH supports the INTA (interrupt acknowledge) special bus cycle for legacy 8259 support. These are routed to the compatibility Intel® 631xESB/632xESB I/O Controller Hub in the system. The INTA will return data that provides the interrupt vector.

7.4.2.3 Message Signalled Interrupts

A second mechanism for devices to send interrupts is to issue the Message Signalled Interrupt (MSI) introduced in the *PCI Local Bus Specification*, Revision 2.2 . This appears as a 1 DWORD write on the PCI/PCI-X/PCI Express bus.

With PCI devices, there are 2 types of MSI's. One type is where a PCI device issues the inbound write to the interrupt range. The other type of MSI is where a PCI device issues an inbound write to the upstream APIC controller (for example, in the PXH) where the APIC controller converts it into an inbound write to the interrupt range. The second type of MSI can be used in the event the OS doesn't support MSI's, but the BIOS does. In either way, the interrupt will appear as an inbound write to the MCH over the PCI Express ports.

MSI is expected to be supported by the operating systems when the MCH is available. The platform will also feature a backup interrupt mechanism in the event that there is a short period of time when MSI is not available. This is described in the next section.

7.4.2.4 Non-MSI Interrupts - "Fake MSI"

For interrupts coming through the PXH, and Intel® 631xESB/632xESB I/O Controller Hub components, their APIC controller will convert interrupts into inbound writes, so inbound interrupts will appear in the same format as an MSI.

For interrupts that are not coming through an APIC controller, it is still required that the interrupt appear as an MSI-like interrupt. If the OS does not yet support MSI, the PCI Express device can be programmed by the BIOS to issue inbound MSI interrupts to an IOxAPIC in the system. The safest IOxAPIC to choose would be the Intel® 631xESB/632xESB I/O Controller Hub since it is always present in a system. Although the MCH supports the PCI Express "Assert_Int" and "Deassert_Int" packets for boot, the performance is not optimal and is not recommended for run time interrupts.

In this method, PCI Express devices are programmed to enable MSI functionality, and given a write path directly to the pin assertion register in a selected IOxAPIC already present in the platform. The IOxAPIC will generate an interrupt message in response, thus providing equivalent functionality to a virtual (edge-triggered) wire between the PCI Express endpoint and the I/OxAPIC.

All PCI Express devices are strictly required to support MSI. When MSI is enabled, PCI Express devices generate a memory transaction with an address equal to the I/OxAPIC_MEM_BAR + 20 and a 32-bit data equal to the interrupt vector number corresponding to the device. This information is stored in the device's MSI address and data registers, and would be initialized by the system firmware (BIOS) prior to booting

a non-MSI aware operating system. (With the theory that an MSI aware O/S would then over-write the registers to provide interrupt message delivery directly from the endpoint to the CPU complex.)

The PCI Express memory write transaction propagates to the MCH and is redirected down the appropriate PCI Express port following the MCH IOAPIC address mapping definition. The IOAPIC memory space ranges are fixed and cannot be relocated by the OS. The assert message is indistinguishable from a memory write transaction, and is forwarded to the destination I/OxAPIC, which will then create an upstream APIC interrupt message in the form of an inbound memory write. The write nature of the message "pushes" all applicable pre-interrupt traffic through to the MCH core, and the MCH guarantees that the subsequent APIC message cannot pass any posted data already within the MCH.

7.4.3 Interprocessor Interrupts (IPIs)

- Previous IA-32 processors use IPIs after reset to select the boot strap processor (BSP). Recent XPF processors including Dual-Core Intel® Xeon® Processor 7200 Series and Quad-Core Intel® Xeon® Processor 7300 Series do not use IPIs to select the BSP. A hardware arbitration mechanism is used instead.
- IA-32 processors use Startup IPIs (SIPIs) to wake up sleeping application processors (non boot strap processors) that are in "Wait for SIPI state". These are broadcast interrupts.
- Interrupts transactions are claimed with TRDY# and No-Data Response.
- For directed XAPIC (A[3] = 0) interrupts, the MCH completes the interrupt normally and forwards the interrupt to the other buses.
- For redirectable XAPIC interrupts, the MCH will generate an interrupt message to all four processor buses MCH with A[3] (redirectable hint bit) set to 0. This message will contain a processor ID based on the redirection algorithm.
- For directed XAPIC broadcast interrupts (Destination ID = 0xFF), the MCH will forward the broadcast interrupt to the other processor buses.
- Interrupts are not deferred.
- Since XAPIC directed interrupts (A[3] = 0) cannot be retried, they must be accepted. If the MCH cannot accept the interrupt, then it must assert BPRI# until resources are available.

7.4.3.1 IPI Ordering

In a system, there are ordering requirements between IPIs and other previous coherent and non-coherent accesses. The way the ordering is maintained is that it is expected that the chipset will defer the previous ordered access. The chipset will not complete the transaction until the write is "posted" or the read data is delivered. Since the processor will not issue an ordered IPI until the previous transaction has been completed, ordering is automatically maintained.

An example where the ordering must be maintained is if a processor writes data to memory and issues an IPI to indicate the data has been written, subsequent reads to the data (after the IPI) must be the updated values. (Producer consumer). For this example, assuming cacheable memory, the chipset defers the BIL/BRIL (read for ownership). Only after all other processor caches have been invalidated, and the deferred reply is returned (where the cache will be written) will the subsequent IPI be issued.

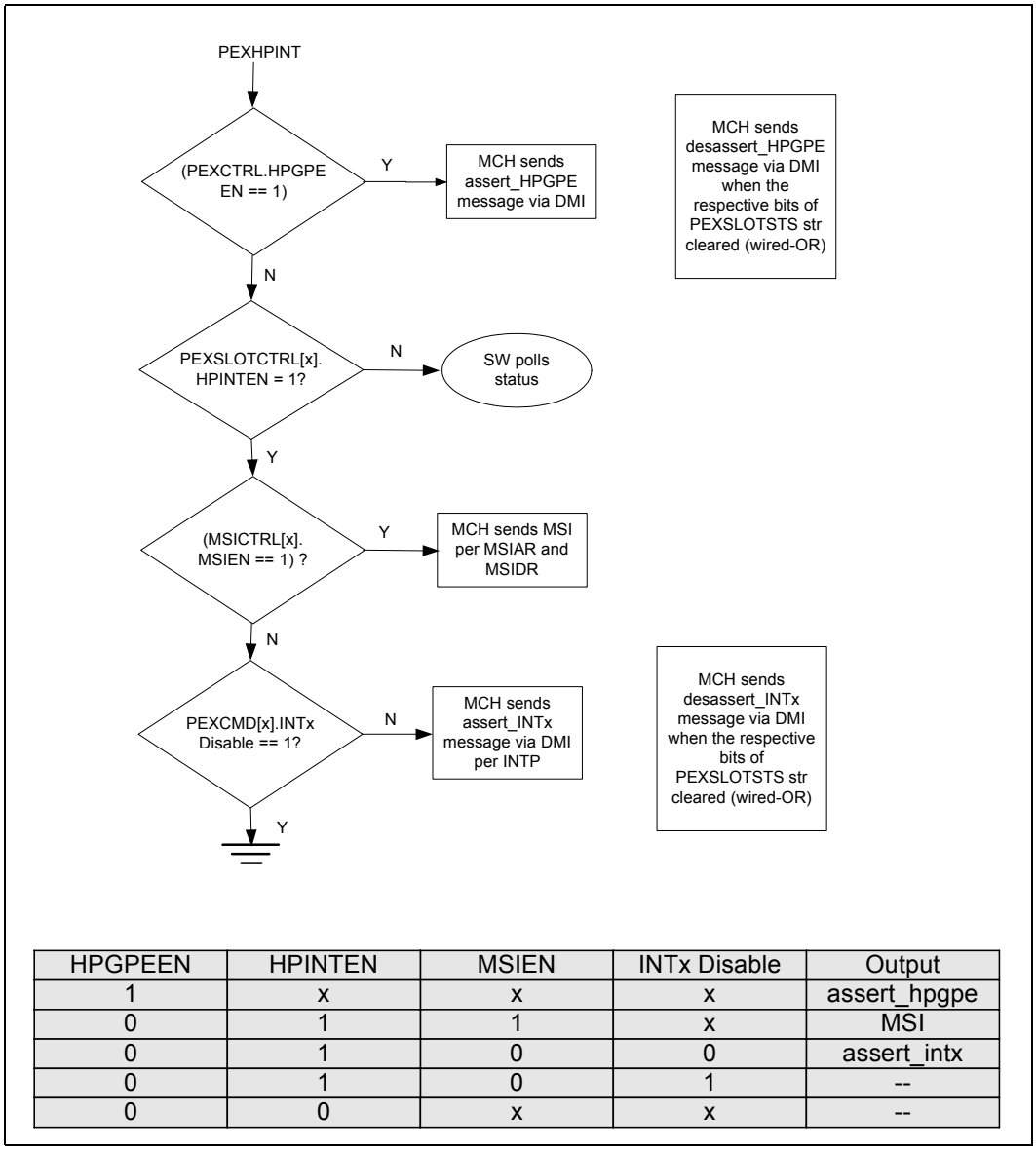
There are no ordering requirements between IPIs. There are no ordering requirements between IPIs and subsequent request. The IPIs are claimed on the FSB (front side bus) and are not deferred. Therefore, software must not rely on the ordered delivery between the IPI and subsequent transactions. If ordering is needed, it must protect any subsequent coherent and non-coherent accesses from the effects of a previous IPI using synchronization primitives. Also, software must not rely on ordered delivery of an IPI with respect to other IPI from the same processor to any target processor.

7.4.4 Chipset Generated Interrupts

The Intel® 7300 Chipset MCH can trigger interrupts for chipset errors and for PCI Express errors and PCI Express hot plug. For these events, the chipset can be programmed to assert pins that the system can route to an APIC controller. The interrupts generated by the chipset are still being defined. The following is a preliminary list of interrupts that can be generated.

1. Chipset error - The MCH asserts appropriate ERR pin, depending on severity. This can be routed by the system to generate an interrupt at an interrupt controller. (the MCH pins ERR[2:0], MCERR).
2. PCI Express error - The MCH asserts appropriate ERR pin, depending on severity. This can be routed by the system to generate an interrupt.
   a. The MCH can receive error indications from the PCI Express ports. These are in the form of inbound ERR_COR/UNC/FATAL messages. The MCH will assert the appropriate ERR signal just like any internal MCH error.
   b. The MCH can be programmed to generate MSIs or legacy INTx for PCI Express RAS errors using the Advanced Error Capability.
3. PCI Express Hotplug - The MCH send Assert_HPGPE (Deassert_HPGPE) or generates an MSI or a legacy interrupt on behalf of a PCI Express Hot-Plug event.
   a. The MCH generated Hot-Plug event such as PresDet change, Attn button, MRL sensor changed, power fault, etc. Each of these events have a corresponding bit in the PCI Express Hot-Plug registers (Attention Button, Power Indicator, Power Controller, Presence Detect, MRL Sensor, Port Capabilities/Slot registers). This will generate an interrupt via the assert_HPGPE, intx, or an MSI. Refer to Figure 7-14 for the Hotplug interrupt flow priority.
   b. PCI Express Hot-Plug event from downstream. This could be either an MSI or a GPE message.
   — MSI: Handled like a normal MSI interrupt (see Figure 7.4.2.3)
   — GPE message: Upon receipt of a Assert_GPE message from PCI Express, the MCH will send assert_GPE signal to the ESI port. To generate an SCI (ACPI), this signal will be routed to the Intel® 631xESB/632xESB I/O Controller Hub appropriate GPIO pin to match the GPE0_EN register settings. When the Hot-Plug event has been serviced, the MCH will receive a Deassert_GPE message. At this point the MCH can deassert_GPE message to ESI. There needs to be a tracking bit per PCI Express port to keep track of Assert/Deassert_GPE pairs. These tracking bits should be OR'd together to determine whether to send the assert_GPE/Deassert_GPE message. When the MCH receives a matching deassert_GPE message for that port, it will clear the corresponding tracking bit. When all the tracking bits are cleared, the MCH will send a Deassert_GPE message to the ESI port.
   — Sideband signals: Some systems may choose to connect the interrupt via sideband signals directly to the Intel® 631xESB/632xESB I/O Controller Hub. No action is required from the MCH.

Figure 7-14. PCI Express Hot-Plug Interrupt Flow



HPGPEEN	HPINTEN	MSIEN	INTx Disable	Output
1	x	x	x	assert_hpgpe
0	1	1	x	MSI
0	1	0	0	assert_intx
0	1	0	1	--
0	0	x	x	--

4. PCI Hot-Plug - Chipset will receive an Assert/Deassert GPE message from the PCI Express port when a PCI Hot-Plug event is happening. Assert/Deassert GPE messages should be treated the same as Assert/Deassert GPE messages for PCI Express Hot-Plug. (Keep track of Assert/Deassert GPE messages from each port and send Assert_GPE, Deassert_GPE message to ESI appropriately)
5. PCI Express Power management - PCI Express sends a PME message. Chipset sends Assert_PMEGPE to ESI port when a power management event is detected.
   a. MSI: For PM_PME message, the MCH will generate an MSI using the normal MSI interrupt flow (Refer to Section 4.8.13.13, "PEXRTSTS[7:0]: PCI Express Root Status Register" on page 164.)

b. Upon receipt of the PME message, the MCH will set the PEXRTSTS.PMESTATUS bit corresponding to that port and send Assert_PMEGPE to ESI port to generate the interrupt. (Assert_PMEGPE should be sent if one or more of the PMESTATUS bits are set and enabled.) To generate an SCI (ACPI), this message will be used by the Intel® 631xESB/632xESB I/O Controller Hub to drive appropriate pin. When software has completed servicing the power management event, it will clear the PEXRTSTS.PMESTATUS bit (by writing 1), at which point the MCH can send Deassert_PMEGPE to ESI port.

The following table summarizes the different types of chipset generated interrupts that were discussed. Although the interrupt and SW mechanism is flexible and can be changed depending on how the system is hooked up, for reference this table also describes what SW mechanism is expected to be used.

Table 7-11. Chipset Generated Interrupts

Source	Signalling mechanism	Intel® 7300 Chipset MCH signal method	Expected SW mechanism
Chipset Error	The MCH registers	ERR[2:0], MCERR, Intel® 631xESB/632xESB I/O Controller Hub Reset	Any
PCI Express Error	PCI Express ERR_COR/UNC/FATAL message	ERR[2:0], MCERR, Intel® 631xESB/632xESB I/O Controller Hub Reset	Any
PCI Express HP (PresDet chg, Attn button, etc.)	The MCH registers For card-these registers are set via the VPP/SM bus interface. For module- these registers are set by inband Hot-Plug messages.	MSI or Assert_intx, Deassert_intx, or Assert_HPGPE, Deassert_HPGPE	SCI->ACPI or MSI
PCI Express HP from downstream device	MSI	MSI interrupt[a] (processor bus[b])	MSI
PCI Express HP from downstream device (non-native, Intel part)	PCI Express Assert/Deassert GPE	Assert_GPE, Deassert_GPE to ESI	SCI->ACPI
PCI Express HP from downstream device (non-native, non-Intel part)	Sideband signals directly to Intel® 631xESB/632xESB I/O Controller Hub	N/A	SCI->ACPI
Downstream PCI Hot-Plug	PCI Express Assert/Deassert GPE	Assert_GPE, Deassert_GPE to ESI	SCI->ACPI
Power Management Event (PME)	PCI Express PM_PME message	Assert_PMEGPE, Deassert_PMEGPE to ESI	SCI->ACPI
Intel® QuickData Technology DMA completion/Error Event	Intel® QuickData Technology Configuration registers (ATTN_STATUS etc.)	MSI, Assert_intx, Deassert_intx to ESI	MSI, SCI->ACPI

Notes:

a. Technically, this is not chipset generated, but is included for completeness
b. See 7.4.1.2 for more details on the processor bus interrupt cycle.

7.4.5 Generation of MSIs

The MCH generates MSIs on behalf of PCI Express Hot-Plug events if the MCH MSICTRL.MSIEN is set. Refer to Figure 7-14. The MCH will interpret PCI Express Hot-Plug events and generate an MSI interrupt based on the MSIAR and MSIDR registers. When the MCH detects any PCI Express Hot-Plug event, it will generate an interrupt transaction to both processor buses. The address will be the value in the MSIAR. The data value will be the value in MSIDR.

Internal to the MCH, the MSI can be considered an inbound write to address MSIAR with data value of MSIDR, and can be handled the same as other inbound writes that are MSIs or APIC interrupts.

Note: MSIs can also generated by the MCH internally for PM_PME messages, RAS errors and Intel® QuickData Technology completions/errors.

7.4.5.1 MSI Ordering in the Intel® 7300 Chipset MCH

Ordering issues on internally generated MSIs could manifest in the MCH if software/device drivers rely on certain usage models e.g. interrupt rebalancing, Hot-Plug to flush them. The producer-consumer violation may happen, if a root port has posted an MSI write internally in the MCH and the software wants to "flush" all MSI writes from the root port i.e. guarantee that all the MSI writes pending in the MCH from the root port have been delivered to the local APIC in the processor. To accomplish this flush operation, OS can perform a configuration read to, say, the VendorID/DeviceID register of the root port and the expectation is that the completion for this read will flush all the previously issued memory writes. The reason the OS wants to flush is for cases where an interrupt source (like a root port) is being retargeted to a different processor and OS needs to flush any MSI that is already pending in the fabric that is still targeting the old processor.

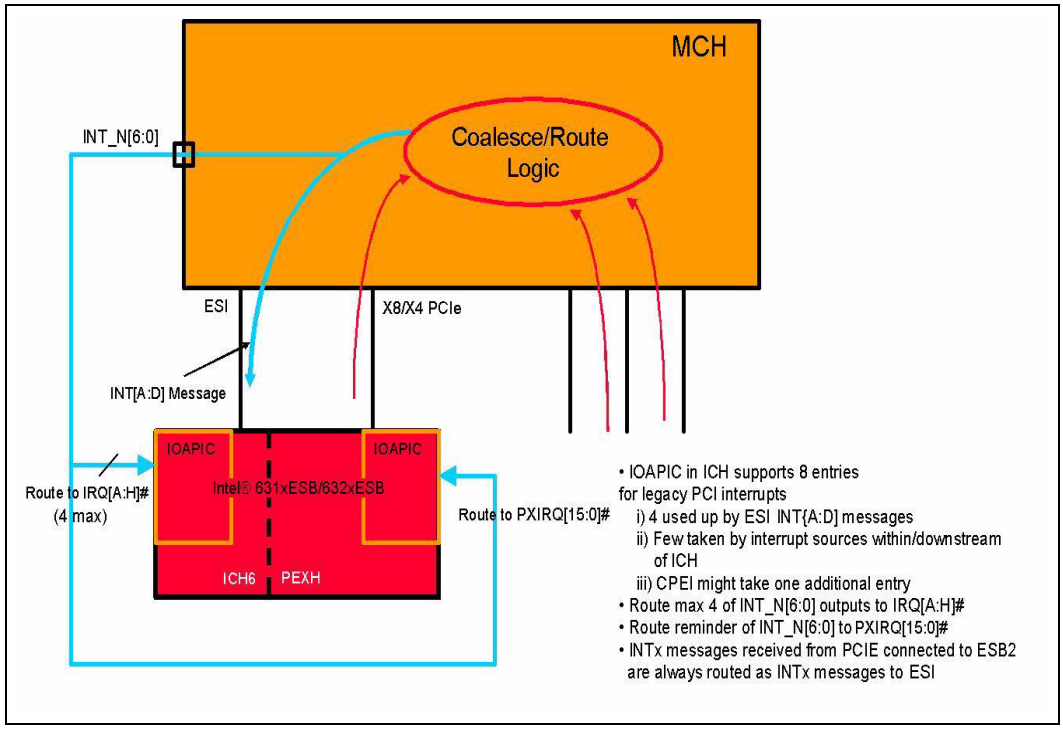
Note: Internal MSIs cannot be continuously generated since the corresponding status register field needs to be cleared by software through configuration access before a new MSI can be asserted.

7.4.6 Legacy PCI-style INTx Message Handling

There are several sources of legacy INTx interrupts in Intel® 7300 Chipset. Some are internal sources as described in Section 7.4.4 and others are external PCIe devices. The MCH consolidates the interrupts received from the various sources and routes them to the Intel® 631xESB/632xESB I/O Controller Hub via inband ESI messages and/or to the interrupt pins as shown in Figure 7-15. Interrupts inputs in to the Intel® 631xESB/632xESB I/O Controller Hub have an internal path to both the integrated Intel® 631xESB/632xESB I/O Controller Hub's I/OxAPIC and the 8259 controller. Interrupts routed to the 7 interrupt output pins INT_N[6:0] of MCH, are routed on the motherboard to the interrupt input pins in the Intel® 631xESB/632xESB I/O Controller Hub. Note that there are couple of groups of interrupt pins in Intel® 631xESB/632xESB I/O Controller Hub that route to interrupt logic in Intel® 631xESB/632xESB I/O Controller Hub/ICH and Intel® 631xESB/632xESB I/O Controller Hub/PEXH. The 7 interrupt output pins from MCH can be routed to either or both of the groups.

The interrupt consolidation logic in Intel® 7300 Chipset takes in 30 different interrupt inputs - 4 * 7 PCIe ports + Intel® QuickData Technology DMA + Root port INTx - and consolidates/routes them to 11 possible outputs - 4 ESI messages + 7 interrupt pins. Each of the 30 interrupt inputs can be routed to any of the 11 interrupt outputs (see Section 4.8.7.10, "INTxROUTECTL: Legacy PCI INTx Interrupt Route Control Register" , register chapter for details), adhering to one platform restriction noted below. All interrupts routed to a given interrupt output (message or pin) are wire-OR'ed internally and the consolidated interrupt signal is routed to the output per the normal rules for handling PCI Express INTx interrupt messages. Platform restriction is that all interrupts from the PCIe root port connected to the Intel® 631xESB/632xESB I/O Controller Hub must always be routed to the ICH (otherwise we would end up in interrupt loops). The easiest way to guarantee this is for bios to always program the INTx interrupts from that root port to inband ESI messages.

Figure 7-15. INTx Interrupt Routing in the Platform



ICH could trigger an 8259 interrupt based on any of its interrupt inputs (inband ESI INTx message and/or IRQ input pins). The 8259 interrupt is routed to the processor on the board and the target processor for the interrupt uses the interrupt acknowledge transaction to obtain the interrupt vector from the 8259 controller. The MCH forwards the interrupt acknowledge to the Intel® 631xESB/632xESB I/O Controller Hub where the active 8259 controller resides. Typically, interrupt inputs into ICH are setup to trigger an 8259 interrupt during boot and once the OS is up and running the 8259 in ICH is turned off and the interrupts are instead serviced via the I/OxAPIC in ICH.

As mentioned earlier, the MCH will have a mechanism to track inband INTx interrupts from each PCI Express port, Intel® QuickData Technology DMA and internally root port generated INTA interrupt and assert virtual interrupt signals to the Intel® 631xESB/632xESB I/O Controller Hub/ICH through the inband "Assert_(Deassert)_Intx" messages on ESI and also assert the interrupt output pins. This is done by a tracking bit per interrupt (A, B, C, D) in each PCI Express port, one tracking bit Intel® QuickData Technology DMA's INTA message and one tracking bit for the internally generated INTA message from across all the root ports (for hotplug, AER, PM). Note that when a PCIe link goes down, any INTx signal associated with that port, that was asserted when the link goes down, is deasserted (whether or not this results in an interrupt output deasserting is dependent on the status of all the interrupt inputs sharing that output).

7.4.7 SMI

SMI (System Management Interrupt) interrupts are initiated by the SMI# signal in the platform. On accepting a System Management Interrupt, the processor saves the current state and enters SMM mode.

Note that the MCH core components do not interact with the LINT[1:0] and SMI signals. They are present on the Intel® 631xESB/632xESB I/O Controller Hub and the processor. The MCH interrupt signals described in Section 7.4.4, "Chipset Generated Interrupts" can be routed to the Intel® 631xESB/632xESB I/O Controller Hub to generate an SMI interrupt. Similarly SCI interrupts can be generated by routing the MCH interrupt signals to the appropriate Intel® 631xESB/632xESB I/O Controller Hub pin.

7.4.8 Interrupt Error Handling

Software must configure the system so that each interrupt has a valid recipient. In the event that an interrupt doesn't have a valid recipient, since the MCH will not necessarily know that the interrupt is targeted for a non-existing processor, will deliver the interrupt to the processor buses following the interrupt routing rules described in this chapter. If the interrupt targets a non-existing processor, it may be ignored but the transaction should still complete.

Any error in the data part of an interrupt message, interrupt acknowledge, or EOI will be treated the same way as data error with any other transaction – single bit errors will be corrected by ECC, double bit error will be treated and logged as uncorrectable.

7.5 PCI Express General Purpose Ports

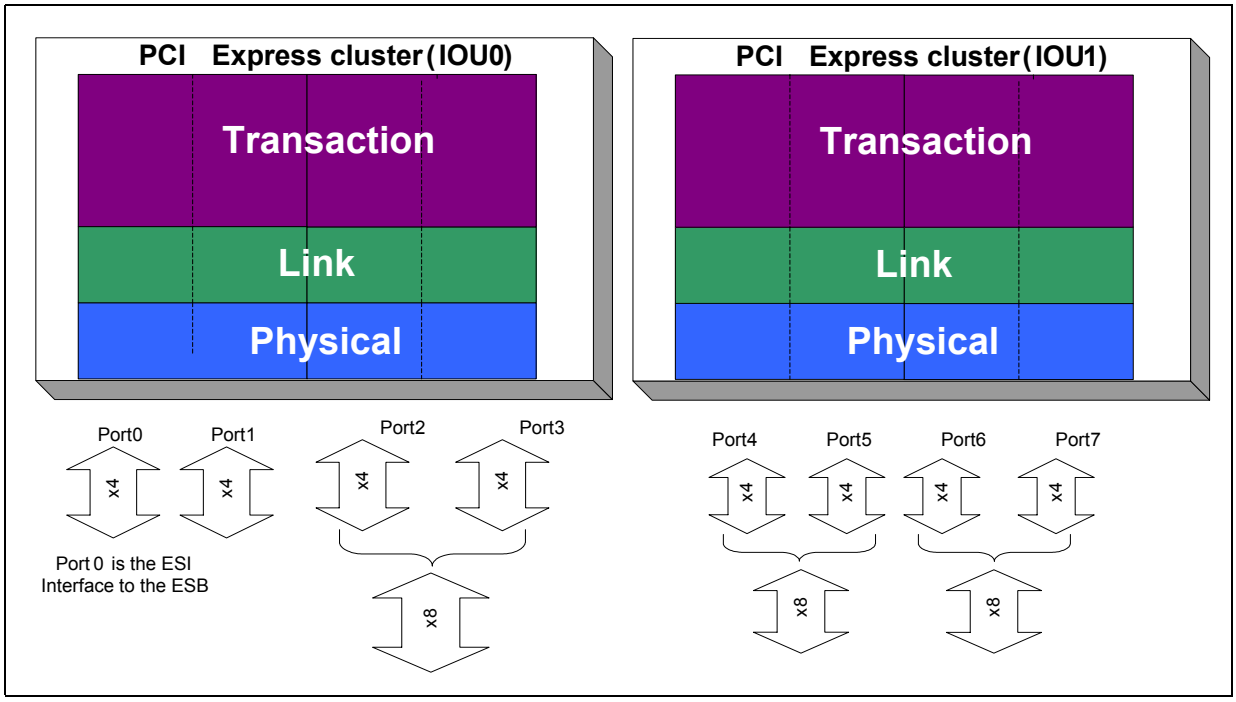
The Intel® 7300 Chipset MCH utilizes general purpose PCI Express high speed ports to achieve superior I/O performance. The MCH PCI Express ports are compliant with the *PCI Express Base Specification, Rev 1.0a*.

The I/O interface consists of seven x4 PCI Express links in addition to a x4 ESI link that are partitioned into two logical I/O Unit (IOU) clusters in the MCH.

- IOU0 - Port 0 (ESI), Ports 1-3
- IOU1 - Ports 4-7

The MCH supports only a finite combinations of ports that permit formation of larger links. Ports 2-3, 4-5 & 6-7 can combine to form three x8 respectively (see Figure 7-16 and Section 7.5.2.2, "Configuration of PCI Express Ports and Widths"). Port 1 is a standalone x4 port and cannot be combined with any other port. Port 0 (ESI) is a standalone port dedicated for an ESI connection to the Enterprise South Bridge (Intel® 631xESB/632xESB I/O Controller Hub). Port 0 cannot be combined with any other PCI Express ports. For details on the ESI Port 0, refer to Section 7.6, "Enterprise South Bridge Interface (ESI)"

Figure 7-16. Intel® 7300 Chipset PCI Express Ports



7.5.1 PCI Express Port Support Summary

The following table describes the options and limitations supported by the MCH PCI Express ports.

Table 7-12. Options and Limitations (Sheet 1 of 2)

Parameter	Support
Number of supported ports	The MCH will support seven x4 standard PCI Express ports and an additional x4 ESI port for Intel® 631xESB/632xESB I/O Controller Hub. (Total: 7+ 1 = 8ports)
Max payload	256B
Hot-Plug	Serial port to support pins
Virtual Channels	MCH only supports VC0
Stream/Port priority	MCH does not support Intel® QuickData Technology Stream/port priority.
Isochrony	MCH does not support isochrony
ECRC	MCH does not support ECRC
Ordering	MCH only supports strict PCI ordering
No Snoop	MCH will not snoop processor caches for transactions with the No Snoop attribute
Power Management	The MCH cannot be powered down, but will forward messages, generate PME_Turn_Off and collect PME_TO_Acks. It will provide the PM Capabilities structure. The MCH does not support Active State Power Management nor the L0s state.

Table 7-12. Options and Limitations (Sheet 2 of 2)

Parameter	Support
No Cable Support & no repeaters	Retry buffers are sized to meet the platform requirements for an integrated MP chassis and which do not require cable or repeater support. Only a 20” inch FR4 internal trace connector is assumed.
Poisoning	MCH will poison data that it cannot correct
Surprise Link Down	The MCH implements the required error reporting and status registers (PCI-SIG DCN #20)
Link Up Signaling	The MCH implements the Link Up status signalling (DCN #20)

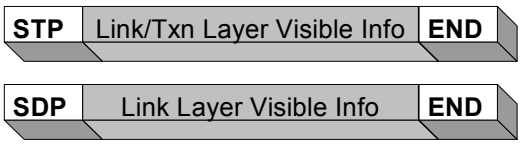
7.5.2 PCI Express Port Physical Layer Characteristics

The PCI Express physical layer implements high-speed differential serial signalling using the following techniques:

- Differential signalling (1.6V peak-to-peak)
- 2.5 GHz data rate (up to 2 GB/s/direction peak bandwidth for a x8 port)
- 8b/10b encoding for embedded clocking and packet framing
- Unidirectional data path in each direction supporting full duplex operation
- Random idle packets and spread-spectrum clocking for reduced EMI
- Loop-back mode for testability
- Lane reversal
- Polarity Inversion

Figure 7-17 illustrates the scope of the physical layer on a PCI Express packet. There are two types of packets: Link layer packets and Transaction Layer Packets. The physical layer is responsible for framing these packets with STP/END symbols (Transaction Layer Packets) and SDP/END symbols (Data Link Layer packets). The grayed out segment is not decoded by the Physical layer.

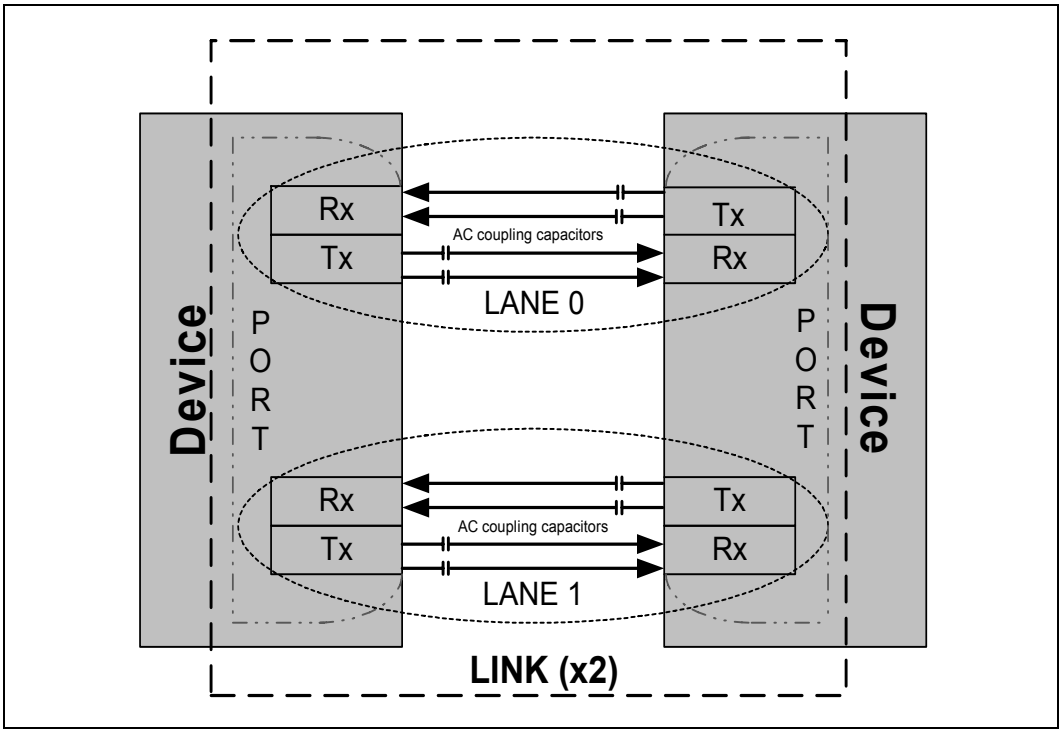
Figure 7-17. PCI Express Packet Visibility By Physical Layer



7.5.2.1 Intel® 7300 Chipset MCH PCI Express Physical Port Overview

A PCI Express port is defined as a collection of bit lanes. Each bit lane consists of two differential pairs in each direction (transmit and receive) as depicted in Figure 7-18.

Figure 7-18. x2 PCI Express Bit Lane



The raw bit-rate per PCI Express bit lane is 2.5 Gbit/s. This results in a real bandwidth per bit lane of 250 MB/s given the 8/10 encoding used to transmit data across this interface. The result is a maximum theoretical realized bandwidth on a x4 PCI Express port of 1 GB/s in each direction.

Each of the MCH PCI Express port are organized as four bi-directional bit lanes, and are referred to as a x4 port.

7.5.2.2 Configuration of PCI Express Ports and Widths

The Intel® 7300 Chipset provides a BIOS controlled mechanism to determine the number of PCI Express ports and their widths. The MCH does not provide external strapping pins to manually configure IOU0 or IOU1 port groups nor does the MCH support automatic port negotiation. For PCI Express ports 1-7, link training is controlled by BIOS exclusively and will not automatically link train after a platform reset. BIOS is required to program the DEVPRES register and the PXPLWTCTRL register properly before initiating PCI Express link training. Once training has been initiated on PCI Express ports 1-7 (by setting bits PXPLWTCTRL.ILNKTRN1 and PXPLWTCTRL.ILNKTRN), the number of active ports and their maximum capable port widths are fixed. More details on PCI Express port configuration and initiating training can be found in Section 4.8.13.15, "PXPLWTCTRL: PCI Express Link Width and Training Control Register" and Section , "" .

7.5.2.3 PCI Express Link Training

To establish a connection between PCI Express endpoints, they both participate in a sequence of steps known as training. For PCI Express ports 1-7, link training is controlled by BIOS exclusively and will not automatically link train after a platform reset. See Section 7.5.2.2 and Section 4.8.13.15 for more details on initiating link training.

PCI Express Link training will establish the operational width of the link, the ordering of lanes within the link (lane reversal), lane polarity (Section 7.5.2.4) as well as adjust skews of the various lanes within a link so that the data sample points can correctly take a data sample off of the link. The MCH port will negotiate to train at the highest common width, and will step down in its supported link widths in order to succeed in training. The ultimate result may be that the link has trained as a x1 link. Although the bandwidth of this link size is substantially lower than a x8 link or x4 link, it will allow communication between the MCH port and the device at the other end of the link. Software will then be able to interrogate the device at the other end of the link to determine why it failed to train at a higher width.

Note: The MCH does not support x2 PCIe cards or x2 PCIe end-point devices

7.5.2.4 Polarity Inversion

The *PCI Express Base Specification*, Revision 1.0a defines a feature called polarity inversion. Polarity inversion allows the board designer to connect the D+ and D- lines incorrectly between devices. The MCH supports polarity inversion on the general purpose PCIe ports (not on the ESI port).

7.5.2.5 PCI Express Lane Reversal

The *PCI Express Base Specification*, Revision 1.0a defines an optional feature called lane inversion. The MCH supports lane reversal on the general purpose PCIe ports (not on the ESI port).

7.5.2.6 PCI Express Port Degraded Mode Operation

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

The Intel® 7300 Chipset supported degraded modes are shown below. The MCH supports degraded mode only on outer lanes as shown in Table 7-13, “Supported Degraded Modes”. The MCH would try a higher width degraded mode before trying any lower width degraded modes.

Table 7-13. Supported Degraded Modes

Original Link Width	Degraded Mode Link Width	Specific Lane Numbers
x8	x4	0-3, 3-0, 4-7, 7-4
	x2	0-1, 1-0
	x1	0, 7
x4	x2	0-1, 1-0
	x1	0, 3

Note: x2 cards are not supported.
Note: Degraded mode is not supported on Port0 (ESI)

7.5.2.7 8b/10b Encoder/Decoder and Framing

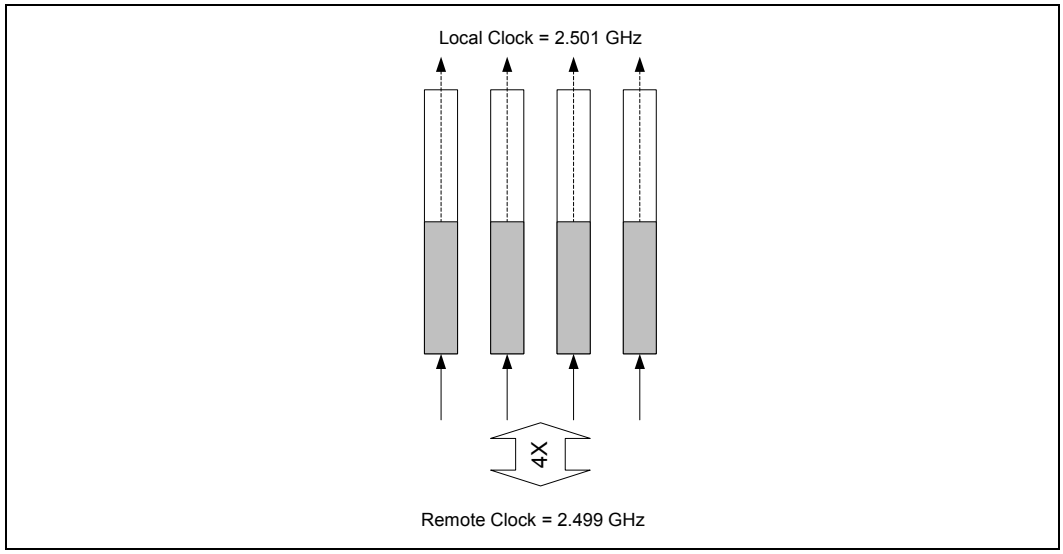
As a transmitter, the physical layer is responsible for encoding each byte into a 10 bit data symbol before transmission across the link. Packet framing is accomplished by the physical layer by adding special framing symbols (STP, SDP, END). PCI Express implements the standard Ethernet and InfiniBand* 8b/10b encoding mechanism.

As a receiver, the 10 bit symbols are decoded to differentiate between STP, SDP, END (used for packet framing), COMMA, SKIP, (described in Section 7.5.2.8 and Section 7.5.2.9), and data symbols. The Physical layer strips off non-data symbols before forwarding to the Data Link layer.

7.5.2.8 Elastic Buffers

Every PCI Express port implements an independent elastic buffer for each PCI Express lane. The elastic buffers are required since the MCH and PCI Express endpoints could be clocked from different sources. Clocks from different sources will never be exactly the same. The outputs of the elastic buffers feed into the deskew buffer.

Figure 7-19. PCI Express Elastic Buffer (x4 Example)



The physical layer periodically transmits special symbols called COMMA and SKIP. These are used to compensate for the variability in clock jitter. They are sent periodically as specified in the *PCI Express Base Specification*, Revision 1.0a.

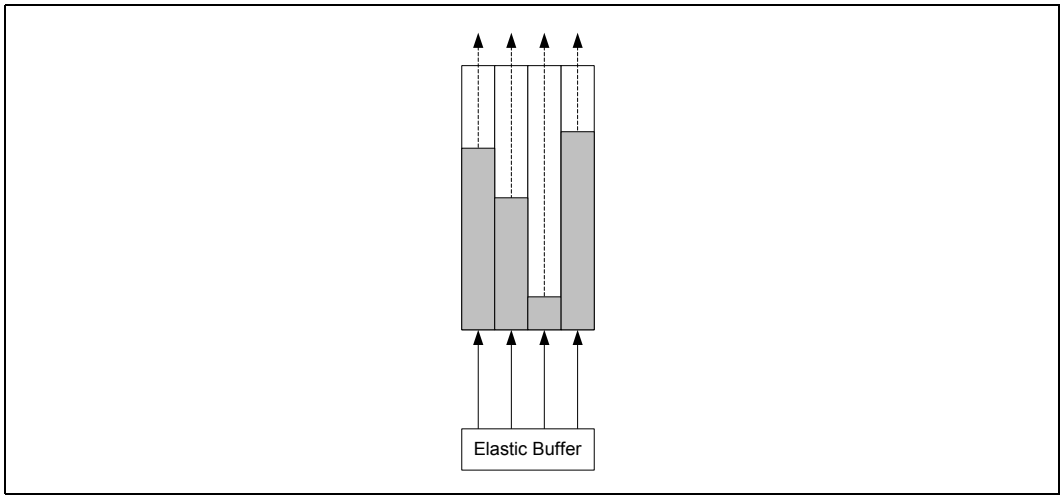
The MCH does not detect Elastic Buffer overflow or underflows. Clocks running outside of the jitter budget or too infrequent COMMA/SKIP updates are violations of the *PCI Express Base Specification*, Revision 1.0a.

The elastic buffer is eight symbols deep. This accounts for three clocks of synchronization delay, the longest possible TLP allowed by the MCH (256B), a 600ppm difference between transmitter and receiver clocks, and worst case skip ordered sequence interval of 1538, framing overheads, and a few symbols of margin.

7.5.2.9 Deskew Buffer

Every PCI Express port implements a deskew buffer. The deskew buffer compensates for the different arrival times for each of the symbols that make up a character. The outputs of the deskew buffer is the data path fed into the Link layer.

Figure 7-20. PCI Express Deskew Buffer (4X Example)



At reset, the delay of each lane in the deskew buffer is adjusted so that the symbols on each lane are aligned. The receiver must compensate for the allowable skew between lanes within a multi-lane link before delivering the data and control to the data link layer.

The deskew buffer is eight symbols deep to compensate for up to 20 ns of skew between lanes. Buffer overflow or underflow should never occur by construction and is considered an error since either the board or interconnect was not designed per specification or the COMMA/SKIP updates are too infrequent.

The MCH does not detect Deskew Buffer overflow or underflows. Clocks running outside of the jitter budget or too infrequent COMMA/SKIP updates are violations of the *PCI Express Base Specification*, Revision 1.0a.

7.5.3 Link Layer

The Data Link Layer of the PCI Express protocol is primarily responsible for data integrity. This is accomplished with the following elements:

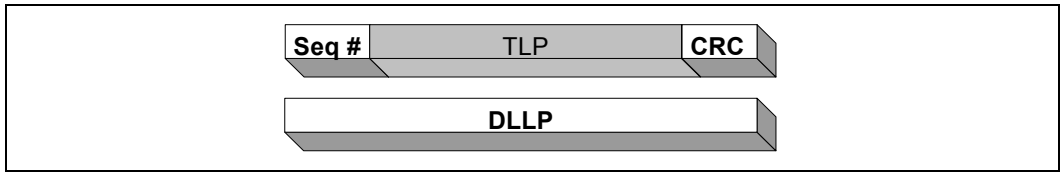
- Sequence number assignment for each packet
- ACK/NAK protocol to ensure successful transmission of every packet
- CRC protection of packets
- Time-out mechanism to detect “lost” packets

- Credit exchange

Figure 7-21 illustrates the scope of the link layer on a PCI Express packet. There are two types of packets: data link layer packets (DLLP) and Transaction Layer Packets (TLP). Data Link layer packets are sent between the Link layers of each PCI Express device and do not proceed to the Transaction Layer.

For Transaction layer packets (TLP), the link layer is responsible for prepending sequence numbers and appending 32-bit CRC. The grayed out segment is not decoded by the Data Link layer.

Figure 7-21. PCI Express Packet Visibility By Link Layer



7.5.3.1 Data Link Layer Packets (DLLP)

Refer to *PCI Express Base Specification*, Revision 1.0a for an explicit definition of all the fields in a Data Link Layer packet.

DLLPs are used to ACK or NAK packets as they are sent from the transmitter to the receiver. They are sent by the receivers of the packet to indicate to the transmitter that a packet was successfully received (ACK) or not (NAK). DLLPs are also used to exchange credit information between the transmitter and receiver.

DLLPs are protected with 16b CRC. If the CRC of a received DLLP indicates an error, the DLLP is dropped. This is safe because the PCI Express protocol supports dropping these packets and the next DLLP allows the transmitter to process successfully.

7.5.3.2 ACK/NAK

The Data Link layer is responsible for ensuring that packets are successfully transmitted between PCI Express agents. PCI Express implements an ACK/NAK protocol to accomplish this. Every packet is decoded by the physical layer and forwarded to the link layer. The CRC code appended to the packet is then checked. If this comparison fails, the packet is "retried".

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. Typically, as each packet is successfully received by the Data Link layer, the receiver issues an ACK. However, the PCI Express protocol allows that ACKs can be combined.

7.5.3.3 Link Level Retry

The *PCI Express Base Specification*, Revision 1.0a lists all the conditions where a packet gets negative acknowledged. 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 1.0a) for incoming packets and comparing the calculated CRC with the received CRC. If they do not match, then the packet is retried by negative acknowledging the packet with a NAK DLLP and specifying the sequence number of the last good packet. Subsequent packets are dropped until the reattempted packet is observed again.

When the transmitter receives the NAK, it is responsible for retransmitting the packet. Furthermore, any packets sent after the last good packet will also be resent since the receiver has dropped any packets after the corrupt packet.

The transmitter keeps track of packets that have been sent but not acknowledged through the use of a retry buffer. Transactions are added to the buffer as they are on the PCI Express port. Transactions are removed from the buffer after they have been acknowledged by the receiver.

7.5.3.4 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. Normally, lost packets are detectable with non-linearly incrementing sequence numbers. A time-out mechanism exists to detect (and bound) cases where the *last* packet sent (over a long period of time) was corrupted. A replay timer bounds the time a retry buffer entry waits for an ACK or NAK. Refer to the *PCI Express Base Specification*, Revision 1.0a for details on this mechanism for the discussion on Retry Management and the recommended timer values.

7.5.4 Flow Control

The PCI Express mechanism for flow control is credit based and only applies to TLPs. DLLP packets do not consume any credits. Through initial hardware negotiation and subsequent updates, a PCI Express transmitter is aware of the credit capabilities of the interfacing device. A PCI Express requester will never issue a transaction when there are not enough advertised credits in the other component to support that transaction. If there are not enough credits, the requester will hold off that transaction until enough credits free up to support the transaction. If the ordering rules and available credits allow other subsequent transactions to proceed, the MCH will allow those transactions.

For example, assume that there are no Non-Posted Request Header Credits (NPRH) credits remaining and a memory write is the next transaction in the queue. PCI Express ordering rules allow posted writes to pass reads. Therefore, the MCH will issue the memory write. Subsequent memory reads from the source device must wait until enough NPRH credits free up.

Note: Flow control is orthogonal with packet ACKs.

The PCI Express flow control credit types are described in Table 7-14. The *PCI Express Base Specification*, Revision 1.0a defines which TLPs are covered by each flow control type.

Table 7-14. PCI Express Credit Mapping for Inbound Transactions[a]

Flow Control Type	Definition	Initial MCH Advertisement
Inbound Posted Request Header Credits (IPRH)	Tracks the number of inbound posted requests the agent is capable of supporting. Each credit accounts for one posted request.	14 (x4) 28(8x)
Inbound Posted Request Data Credits (IPRD)	Tracks the number of inbound posted data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data.	54 (x4) 108(8X)
Inbound Non-Posted Request Header Credits (INPRH)	Tracks the number of non-posted requests the agent is capable of supporting. Each credit accounts for one non-posted request.	14 (x4) 28(8X)

Table 7-14. PCI Express Credit Mapping for Inbound Transactions[a]

Flow Control Type	Definition	Initial MCH Advertisement
Inbound Non-Posted Request Data Credits (INPRD)	Tracks the number of non-posted data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data.	2 (x4) 4 (8X)
Completion Header Credits (CPH) (outbound request completions received at the MCH)	Tracks the number of completion headers the agent is capable of supporting.[b]	0 (Infinite) (4) [x4] (8) [x8]
Completion Data Credits (CPD) (outbound request completions (data) received at the MCH)	Tracks the number of completion data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data.	0 (Infinite) (8) [x4] (16) [x8]

Notes:

a. This also applies to the ESI Port
b. Root complexes and end points are permitted to advertise an infinite number of credits for completions. Though The MCH implements finite queue structures as indicated in bracket for the completions on the inbound side, by construction, it will never overflow since for each outbound request, the MCH allocates sufficient space on the inbound side. i.e guarantee by construction

Table 7-15. PCI Express Credit Mapping for Outbound Transactions

Flow Control Type	Definition	Initial MCH Advertisement
Outbound Posted Request Header Credits (OPRH)	Tracks the number of outbound posted requests the agent is capable of supporting. Each credit accounts for one posted request.	4(x4) 8(8x)
Outbound Posted Request Data Credits (OPRD)	Tracks the number of outbound posted data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data.	8(x4) 16(8X)
Outbound Non-Posted Request Header Credits (ONPRH)	Tracks the number of non-posted requests the agent is capable of supporting. Each credit accounts for one non-posted request.	16(x4) 32(8X)
Outbound Non-Posted Request Data Credits (ONPRD)	Tracks the number of non-posted data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data.	16(x4) 32(8X)
Completion Header Credits (CPLH) (inbound request completions from MCH)	Tracks the number of completion headers the agent is capable of supporting.	2(x4) 4(x8)
Completion Data Credits (CPLD) (inbound request completions (data) from the MCH)	Tracks the number of completion data the agent is capable of supporting. Each credit accounts for up to 16 bytes of data.	8(x4) 16(x8)

The credit advertisements for the MCH are shown in Table 7-14 and Table 7-15. Every PCI Express device tracks the above six credit types (inbound) for both itself and the interfacing device. The rules governing flow control are described in the *PCI Express Base Specification*, Revision 1.0a.

7.5.4.1 Credit Update Mechanism, Flow Control Protocol (FCP)

After reset, credit information is initialized with the values indicated in Table 7-14 by following the flow control initialization protocol defined in the *PCI Express Base Specification*, Revision 1.0a. Since the MCH supports only VC0, only this channel is initialized.

7.5.4.1.1 Credit Update Mechanism (MCH as Transmitter)

Credit updates are transmitted between devices with Flow Control Packets (FCP). The transmit-side of a port uses FCP updates seen by its receive side to determine the number of outstanding requests the target is capable of receiving. A data credit corresponds to 1-16 bytes of data. A header credit corresponds to one header.

For each credit type, the PCI Express cluster implements a pool of credits representing the number of credits available in the receiver. Conceptually, each time a request is issued, the appropriate header credit pool is decremented by one and the data credit pool of available credits is decremented based on the amount of data transmitted (if any). If there are insufficient credits available, the next transaction will not be issued.

FCPs are sent back to the transmitter to recycle credits and the transmitter reloads the pool counter with the FCP payload. The FCP packets includes all credit pool information. PCI Express implements an "absolute mechanism" for returning credits. In other words, the FCP specifies the number of credits remaining in the receiver which solves the risk of losing credits due to transmission errors on DLLPs that are returning credits.

7.5.4.1.2 Credit Update Mechanism (MCH as Receiver)

As a receiver, the MCH is responsible for updating the transmitter with flow control credits as the packets are accepted by the Transaction Layer and processed by the interface logic in the MCH.

7.5.4.2 PCI Express Link Training

PCI-Express v1.0a protocol consists of three layers: the transaction layer, the data link layer, and the physical layer. Initialization of the link is handled mainly by the physical layer and is referred to as link initialization and training. Link training can be initiated by several mechanisms, generally from one of the multiple forms of reset. Link training consists of multiple state transitions, referred to as the ltssm, during which each link partner is responsible to transmit specific types of training ordered sets, and to expect to receive specific responses. The final state of link training is referred to as L0, at which time the physical layer is ready to send and receive packets from the higher layers. In cases of link training where the data link is down, the entry to L0 should trigger the data link to begin its initialization sequence. For more detail on Pci-Express link training states, trainings ordered set, and types of link reset which will initiate link training, refer to Sec 4.2.4-6 of the PCI Express Base Specification, Revision 1.0a. Information on the data link layer and its initialization sequence can be found in Sec. 3.2 of the same specification.

7.5.4.3 PCI-Express Surprise Link Down and Link Up Support

The Intel® 7300 Chipset implements the surprise link down error and link up signalling register fields as part of PCI-SIG DCN #20. This is an addendum to the PCI Express Base Specification, Revision 1.0a and it provides better debug and error/status reporting. Refer to Section 4.8.13.6, "PEXLNKCAP[7:0]: PCI Express Link Capabilities Register" on page 152 and Section 4.8.13.8, "PEXLNKSTS[7:0]: PCI Express Link Status Register" on page 155.

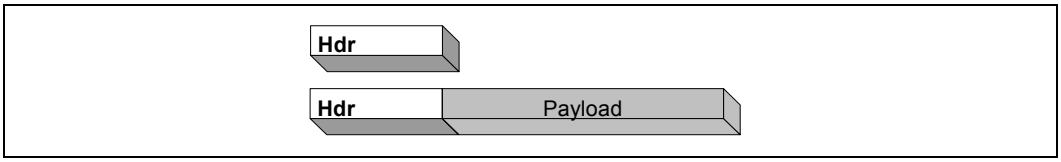
7.5.5 Transaction Layer

The PCI Express Transaction Layer is responsible for sending read and write operations between components. This is the PCI Express layer which actually moves software visible data between components. The transaction layer provides the mechanisms for:

- Software configuration of components
- Communication between the processor bus and different I/O technologies
- Communication between the memory and different I/O technologies

Figure 7-15 illustrates the scope of the transaction layer on a PCI Express packet. Some transaction layer packets have only a header (e.g. read request). Some transaction layer packets have a header followed by data (e.g. write requests and read completions).

Figure 7-22. PCI Express Packet Visibility By Transaction Layer



7.5.6 Stream/Port Arbitration

Arbitration in the PCI Express (IOU) cluster(s) occurs at several levels to dispatch transactions to/from memory. The basic methodology is to guarantee fairness, prevent starvation and provide some degree of programmability for debug/fine-tuning.

- Level 0 (lowest) - In order to provide better serviceability for high-priority streams from Ports 2 and 3, the MCH employs a mixed stream/port arbitration within IOU0 cluster as follows:
  - The low-priority streams from Ports 2 and 3 are grouped along with Port 0 (ESI) and round-robin arbitration selects a request from these queues i.e low-priority (Port 2), low-priority (Port 3) and ESI (Port 0)
  - The high-priority streams from Ports 2 and 3 are grouped together and round-robin arbitration proceeds for these requests from high-priority (Port 2) and high-priority (Port 3)
- Level 1: Programmable, weighted round-robin algorithm that alternates between the high priority and low priority stream requests selected from Level 0 of IOU0 and classifies them as Posted/Non Posted.
- Level 1a: Port Arbitration: Ports 4, Port 5, Port 6 and Port 7
  - Select one of the requests from IOU1 in a round robin fashion and classify as Posted/Non-Posted
- Level 2: Simple round robin algorithm between the Posted, Non-posted queues, and completions from Level 1 (IOU0) and Level 1a (IOU1) respectively.
- Level 3 (highest): Programmable, weighted round robin algorithm that connects the IOU0, IOU1 and SDMA engine with the CDB to share the allocated 8.53/10.6 GB/s bandwidth in each direction.

7.5.6.1 Level 3 - IOU0, IOU1, DMA Arbitration

The Level 3 arbitration shares available bandwidth in the CDB port by a programmable, weighted, round robin scheme that scans for requests from the two IOUs (IOU0 & IOU1) and the DMA engine optimizing bandwidth and avoiding starvation. The CDB port has a bandwidth of 8.53 GB/s (10.6 GB/s) and is proportionately divided based on the arbitration values defined for DMA, IOU0, IOU1 in the PEXCTRL2[7:2:0], PCI Express Control Register 2.

7.5.7 Supported PCI Express Transactions

Table 7-16 lists all the transactions supported by the MCH which are expected to be received from the PCI Express interface. Similarly, Table 7-18 lists all the transactions to be expected by an attached PCI Express component. Refer to the *PCI Express Base Specification*, Revision 1.0a for the specific protocol requirements of this interface.

Table 7-16. Incoming PCI Express Requests

PCI Express Transaction	Address Space or Message	Intel® 7300 Chipset MCH Response
Inbound Write Requests	Memory	Forward to Main Memory or PCI Express or ESI port depending on address.
Inbound Read Requests	Memory	Forward to Main Memory, or PCI Express or ESI
Inbound Message	ASSERT_INTA	Inband interrupt assertion/deassertion emulating PCI interrupts.
	DEASSERT_INTA	
	ASSERT_INTB	Inband interrupt assertion/deassertion emulating PCI interrupts.
	DEASSERT_INTB	
	ASSERT_INTC	Inband interrupt assertion/deassertion emulating PCI interrupts.
	DEASSERT_INTC	
	ASSERT_INTD	Inband interrupt assertion/deassertion emulating PCI interrupts.
	DEASSERT_INTD	
	ERR_COR	Propagate as an interrupt to system.
	ERR_UNC	Propagate as an interrupt to system.
	ERR_FATAL	Propagate as an interrupt to system.
	PM_PME	Propagate as a general purpose event to the system via the PME_OUT pin.
	PM_TO_ACK	Terminate the PME_Turn_OFF message issued from the originating PCI Express port
	PM_ENTER_L1, PM_ENTER_L23 (DLLP)	These messages are issued by downstream components that indicate their entry into L1 or L2/L3 states. The MCH must block subsequent TLP issue and wait for all pending TLPs to Ack. Then, send PM_REQUEST_ACK.
	ASSERT_GPE	Send the received "Assert_GPE" message at the PCI Express port to the ESI port as a virtual wire using a wired-OR approach. NOTE: This is an Intel vendor-specific message.
	DEASSERT_GPE	Send the received "Deassert_GPE" message at the PCI Express port to the ESI port as a virtual wire using a wired-OR approach. NOTE: This is an Intel vendor-specific message.
	FENCE_MSG	This message is used to provide ordering between different streams in the given PCI Express port
	IQD Query	This message is sent by the TOE device to request information on the Intel® QuickData Technology BAR and per port offset for configuration.
	Others	Set IO2 error (unsupported request), drop transaction (master abort) and return credit.

Table 7-17. Incoming PCI Express Completions

PCI Express Transaction	Address Space or Message	Intel® 7300 Chipset MCH Response
Completions for Outbound Writes	I/O or Configuration[a]	Forward to the processor bus, PCI Express or ESI from which the request originated.
Completions for Outbound Reads	Memory, I/O or Configuration	Forward to the processor bus, PCI Express or ESI from which the request originated.

Notes:

a. Outbound Memory writes are posted and have no completions

Table 7-18. Outgoing PCI Express Requests

PCI Express Transaction	Address Space or Message	Reason for Issue
Outbound Write Requests	Memory	Processor bus or peer memory-mapped I/O write targeting PCI Express device.
	I/O	Processor legacy I/O write targeting PCI Express device.
	Configuration	Processor or peer configuration write targeting PCI Express device.
Outbound Read Requests	Memory	Processor or peer memory-mapped I/O read targeting PCI Express device.
	I/O	Processor or peer I/O read targeting PCI Express device.
	Configuration	Processor or peer configuration read targeting PCI Express device.
Outbound Messages	EOI (Intel-specific)	End-of-interrupt cycle received on processor bus, the MCH broadcasts this message to all active PCI Express ports. Devices supporting edge triggered interrupts will ignore this cycle.
	Lock/Unlock	When a locked read or write transaction was previously issued to a PCI bridge, "Unlock" releases the PCI lock.
	PM_TURN_OFF	PEXGCTRL.PME_TURN_OFF bit was set. This message is broadcast to all enabled PCI Express ports.
	PM_REQUEST_ACK (DLLP)	Received PM_ENTER_L1 and PM_ENTER_L23. This message is continuously issued until link is idle.
	IQD Query Response	This message is sent back by the MCH in response to the IQD query and contains pertinent information on IQD version, BAR offset etc.

Table 7-19. Outgoing PCI Express Completions

PCI Express Transaction	Address Space or Message	Reason for Issue
Inbound Read Completions	Memory	Response for an inbound read to main memory or a peer I/O device.
	I/O	Response for an inbound read to a peer I/O device.

7.5.7.1 Unsupported Messages

If the MCH decodes any vendor message (which is not defined in Table 7-16), the MCH will take the following actions as specified in the Vendor defined message section of the *PCI Express Base Specification*, Revision 1.0a.

- Vendor Type 0 - Unsupported Request
- Vendor Type 1 - Drop request.

7.5.7.2 32/64 Bit Addressing

For inbound and outbound writes and reads, the MCH supports 64-bit address format. If an outbound transaction's address is a 32-bit address, the MCH will issue the transaction with a 32-bit addressing format on PCI Express. Only when the address requires more than 32 bits will the MCH initiate transactions with 64-bit addressing format. It is the responsibility of the software to ensure that the relevant bits are programmed for 64 bits based on the OS limits. (e.g 36 bits for MCH)

7.5.8 Transaction Descriptor

The *PCI Express Base Specification*, Revision 1.0a defines a field in the header called the Transaction Descriptor. This descriptor comprises three sub-fields:

- Transaction ID
- Attributes
- Virtual Channel ID

7.5.8.1 Transaction ID

The Transaction ID uniquely identifies every transaction in the system. The Transaction ID comprises four sub-fields described in Table 7-20.

Table 7-20. PCI Express Transaction ID Handling

Field	Definition	MCH as Requester	MCH as Completer	Peer-to-peer Transaction
Bus Number	Specifies the bus number that the requester resides on.	The MCH sets this field to 0.	The MCH preserves this field from the request and copies it to the completion.	For peer-to-peer posted requests, the MCH preserves these fields from the source PCI Express port to the destination port. For peer-to-peer non-posted requests, the MCH preserves Requestor ID Bus, Device, and Function, but reassigns the Tag.
Device Number	Specifies the device number of the requester.	The MCH fills this field in with the content of the DID register for this port.		
Function Number	Specifies the function number of the requester.	The MCH sets this field to 0.		
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.	The MCH fills this field in with a value such that every pending request carries a unique Tag.		

7.5.8.2 Attributes

PCI Express supports two attribute hints described in Table 7-21.

Table 7-21. PCI Express Attribute Handling

Attribute	Definition	MCH as Requester	MCH as Completer	Peer-to-peer Transaction
Relaxed Ordering	Allows the system to relax some of the standard PCI Express ordering rules.	For outbound transactions, this bit is not applicable and set to zero.	The MCH ignores this field on inbound transactions. The MCH makes no proactive attempts to reorder differently based on the value in this field.	For requests and completions, preserve this field from the source PCI Express port to the destination port.
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 bus bandwidth.		If this attribute is set for inbound transactions, the MCH will not snoop the transaction on the processor buses.	

7.5.8.3 Traffic Class

The MCH does not support any PCI Express virtual channels other than the default channel (channel 0). Therefore, the MCH effectively ignores the traffic class field for inbound requests. For inbound completions, the MCH will copy the TC value received into the completion. For peer-to-peer completions, the TC value of the request is preserved.

For outbound requests, the MCH sets this ID to zero.

7.5.9 Transaction Behavior

This section describes the specifics of how PCI Express transactions flow through the MCH. This section covers both generic PCI Express transactions and ESI transactions.

7.5.9.1 Inbound Transactions

Inbound requests should be serviced to maximize PCI Express bandwidth for the given link without stalling. The MCH will accept the transactions listed in Table 7-20. This section describes handling that is specific to the MCH for transactions that target the MCH or main memory. Ordering rules for inbound transactions are described in Section 7.5.9.4.

7.5.9.1.1 Inbound Memory Reads

Read Completion Policy

For inbound read requests, the Intel® 7300 Chipset is allowed to split completions along an Read Completion boundary (RCB) of 64B (See Section 4.8.13.7, "PEXLNKCTRL[7:0]: PCI Express Link Control Register" . For the MCH, the maximum size of a read completion is specified with Max_Payload_Size field as 256B[1] (See Section 4.8.27.18, "PEXDEVCTRL: Device Control Register"). If the PCI Express interface is idle, the MCH will return a completion for that read starting at the initial address up to the next cacheline boundary.

If a PCI Express interface is busy sending an outbound packet, the MCH will opportunistically combine subsequent inbound read completions up to Max_Payload_Size or until the initial request length is satisfied. Note that completion combining helps increase bus efficiency due to reduced header overhead on the PCI Express port.

7.5.9.1.2 Inbound Read/Write Streaming

The MCH IOU cluster implements extensive pipe-lining and non-speculative prefetching to maximize throughput and utilization of the various PCI Express interconnects. The IOU uses two phases to handle a transaction. A transaction cycle starts with a "prefetch" followed by a "fetch" cycle and terminates with a fetch completion.

1. Prefetch Phase: The IOU launches multiple cache lines as non-speculative prefetches for one or more enqueued requests. These prefetch commands are routed to the Coherency Engine (CE)/ Data Manager (DM) for decoding and then sent to memory/MIB (for snoops) if required.
2. Fetch Phase: Request data from cache lines in the CE/DM for sending to destination port. A Fetch completion is sent to terminate the cycle and remove buffer entries. In some cases, a fetch request can be issued without a prefetch for optimization (e.g. an initial read to memory when completion buffers are empty).

1. The read completion combining is 256B for (x8) and 128B for x4 modes respectively.

In the prefetch phase, the streaming logic in the IOU breaks inbound transactions (in the order received) into one or more cache lines and pipelines them to the CE. It should send enough requests up to the total number of outstanding cache lines that the MCH can handle to maximize bandwidth on the PCI Express port. When acknowledgement for the individual cache lines in the prefetch phase is returned, the fetch phase begins immediately and internal commands are pipelined to the CE/DM to obtain the data as fast as possible. Finally the data is packaged into appropriate TLPs (e.g. read completions) and returned to the PCI Express interface based on the request arrival order and the appropriate completion combining conditions prevailing for that port.

7.5.9.1.3 Zero-Length Reads/Writes

The *PCI Express Base Specification*, Revision 1.0a describes that a zero-length memory read must be supported and may be used by devices as a queue flushing mechanism. With the PCI Express ordering rules, a device can issue such a read to push ahead all previously issued writes.

When the MCH receives a zero-length read, data is not actually read from anywhere. Rather, the read is completed by the MCH after all writes previously posted on that inbound port are considered to be globally visible by the system. At that point, the MCH will return one DW of non zero values to the requesting PCI Express port.

The *PCI Express Base Specification*, Revision 1.0a also does not preclude the arrival of zero length writes on any of the PCI Express ports. For coherence compatibility and general software usage expectation, it is required to perform an RFO (Request For Ownership) for the cache line involved in the zero length write and then commit the unmodified cache line to memory. Similarly on the outbound path, the MCH will forward zero length reads and writes to the respective destinations.

7.5.9.1.4 Inbound Write Transactions

Inbound coherent write transactions actually comprise two operations: request for ownership, and the cache line write (mark to modified state). The PCI Express unit will enqueue each inbound write as a single atomic instruction. As the PCI Express unit enqueues the write, it will also bypass all queues by sending a request-for-ownership command (RFO) directly to the processor buses requesting for line ownership. The RFO commands are allowed to be issued in any order.

If the MCH owns the line after all inbound ordering rules have been met, the write command proceeds and the line is modified. If the line is not owned by the MCH when after all inbound ordering rules have been met, the write is temporarily stalled until ownership is acquired and requests will continue to fill the inbound queue. When the request for ownership completes, the write command is forwarded where the line is marked in the modified state.

For write transactions with the Don’t Snoop attribute, the PCI Express unit (following all inbound ordering rules) will simply issue a write command to memory without snooping the processor buses. Note that ordering is required between normal and “Don’t Snoop” transactions.Inbound

7.5.9.1.5 Inbound Write Combining

The MCH performs write combining opportunistically in the on-chip coherent data buffer.

7.5.9.1.6 Interrupt Handling

A PCI Express device represents interrupts with either MSI or inbound interrupt messages (ASSERT_INTx/DEASSERT_INTx).

Each PCI Express port of the MCH is responsible for tracking assert/deassert messages for each of the four interrupts (INTA, INTB, INTC, INTD) and representing them with four output "virtual (ASSERT_INTA, ASSERT_INTB, ASSERT_INTC, and ASSERT_INTD) messages to the Intel® 631xESB/632xESB I/O Controller Hub.

The Intel® 7300 Chipset has an interrupt swizzling logic to rebalance and distribute inbound legacy interrupts for performance and load balancing considerations in the OEM systems. To this end, the MCH implements the "barber-pole" mechanism (round robin) strided interrupt remapping scheme. Software/BIOS configures the appropriate swizzling required for each of the ports during boot time. Incoming interrupts are then routed based on the look-up table for the respective port and the IOU handles the assert/deassert_intx messages using the wired-OR approach.

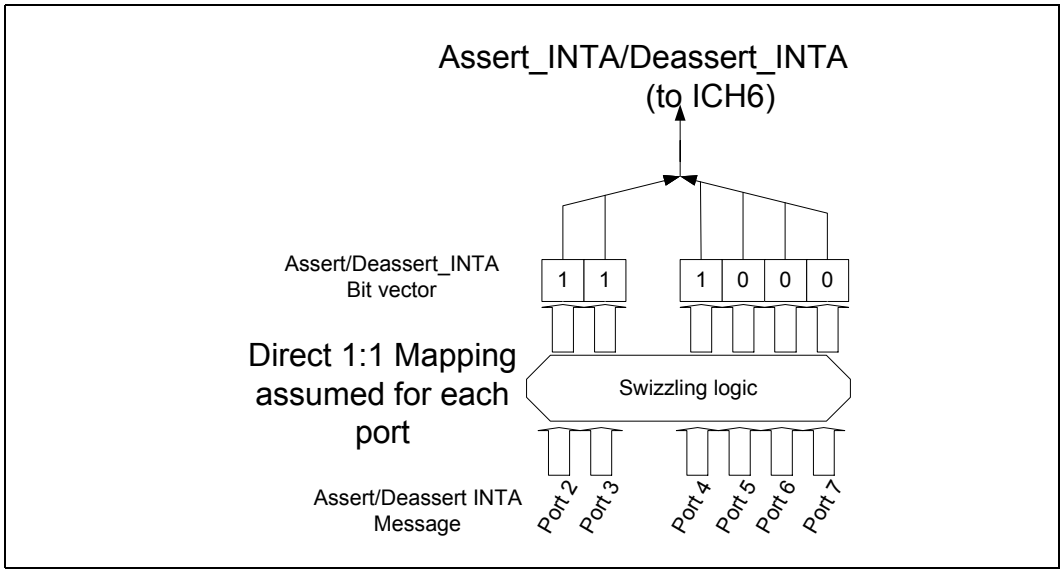
Figure 7-23 illustrates how the PCI Express interrupts are routed to the ICH. The example shown represents Interrupt A associated with an assert/deassert state and this logic is replicated for the four interrupts. None of the bits depicted are software visible.

Note: If the port associated with one of these set bits goes to the "DL_down" status, the MCH will reset the state bit for that port so that the MCH can synthetically generate the Deassert_Intx message through the wired-or mechanism to terminate the interrupt.

For more details on the MCH interrupt handling, refer to Section 7.4, "Interrupts" .

When MSI interrupts are used, the MCH treats these writes as any other inbound write. The difference is that MSI writes are detected as a write to addresses in the range FEE0_0000 - FEEF_FFFF. If the write falls within this range, the MCH issues the write to both processor buses where it will be claimed by the targeted CPU.

Figure 7-23. Legacy Interrupt Routing (INTA example)



7.5.9.1.7 Error Messages

PCI Express reports many error conditions through explicit error messages: ERR_COR, ERR_UNC, ERR_FATAL.

7.5.9.1.8 Inbound Vendor-Specific Messages

ASSERT_GPE and DEASSERT_GPE form a virtual wire which is sent by a PCI Express device. The Intel® 631xESB/632xESB I/O Controller Hub component issues these messages as a response to its integrated PCI Express Hot-Plug controllers.

7.5.9.2 Outbound Transactions

The MCH will generate the outbound transactions listed in Table 7-18. This section describes handling that is specific to the MCH for transactions that target a PCI Express interface. Ordering rules for outbound transactions are described in Table 7.5.10.2.

7.5.9.2.1 Outbound Non-Posted Transactions

Non-posted transactions that the MCH supports includes memory reads, I/O reads and writes, and configuration reads and writes. When a non-posted transaction is issued from the MCH, the PCI Express device will respond with a completion.

Each PCI Express interface supports up to four outstanding non-posted transactions comprising transactions issued by the processors or a peer PCI Express device.

Stalled Non-Posted Requests

Non-posted requests are non-blocking transactions. In other words, while a non-posted request is pending, subsequent transactions are required to bypass the transactions which are waiting for a completion.

7.5.9.2.2 Outbound Posted Transactions

Once a posted request (memory mapped I/O write) is issued from the PCI Express's transaction layer, the request is considered to be complete and the transaction is removed from the outbound queue. For posted requests, the acknowledge has already been sent to the initiating interface (processor bus or alternate PCI Express inbound queue) when the write was enqueued in the outbound PCI Express unit so proper ordering is guaranteed.

7.5.9.2.3 Outbound Vendor-Specific Messages

The MCH supports only vendor-specific EOI messages outbound.

EOI Message

EOI messages will be broadcast to all the PCI Express interfaces and require posted transaction ordering. This ensures that the appropriate interrupt controller receives the end-of-interrupt. Depending on outbound traffic patterns, the EOIs will often be delivered on the PCI Express ports at different times.

EOI is a message required for I/OAPICs which support XAPIC. Since EOI is Intel-specific, this PCI Express message can only be forwarded to Intel devices that support an integrated I/OAPIC supporting level-sensitive interrupts (Intel® 631xESB/632xESB I/O Controller Hub).

7.5.9.2.4 Lock Support

For legacy PCI functionality, the MCH supports bus locks through an explicit sequence of events. The MCH can receive a locked transaction sequence (Read-Write or Read-Read-Write-Write) on a processor interface directed to a PCI Express-to-PCI bridge.

Note that native PCI Express devices are prohibited from supporting bus locks according to the *PCI Express Base Specification*, Revision 1.0a.

The PCI Express interface cluster must support the following capabilities:

- Block all transactions on the PCI Express ports (when the lock targets another port)
- Generate a locked read request to the target PCI Express port
- Unlock the locked PCI Express port

The MCH lock flow will not complete if a memory read is sent to a native PCI Express endpoint, which will return an unsupported request (UR) status response. Operation is not guaranteed, if a lock to a native PCI Express device is issued. Locks to PCI through PCI Express will still be supported.

7.5.9.3 Peer-to-Peer Transactions

Peer-to-peer support is defined as transactions which initiate on one I/O interface and target another without going through main memory. The MCH supports peer-to-peer transactions only for memory transactions. Any PCI Express interface can be the source or destination of a peer-to-peer transaction. Peer-to-peer transactions are not observed on any interface except the target and destination (they are not snooped by the processors).

Inbound coherent transactions and peer-to-peer transactions must maintain PCI Express ordering rules between each other. Peer-to-peer transactions follow inbound ordering rules until they reach the head of the inbound queue. Once the transaction reaches the head of the inbound queue, the MCH routes the transaction to the next available slot in the outbound queue where PCI ordering is maintained until the end of the transaction. The MCH does not support peer-to-peer transactions where the source and destination are on the same PCI Express interface.

7.5.9.4 PCI Express Hide Bit

To “hide” PCI Express ports from the OS, there is one PEXCTRL.DEVHIDE bit for each port. When firmware sets the write-once PEXCTRL.DEVHIDE bit, the PCI Express device in the MCH changes its response to a CCR read (Class Code Register, 0x0600h) (Refer to Section 4.8.2.4, “CCR: Class Code Register” on page 91) so that it appears to be a host bridge. The OS will not perform configuration reads and writes to a device with its hide bit set. See Section 4.8.10.31, “PEXCTRL2[7:0]: PCI Express Control Register 2” on page 137 for details.

7.5.10 Ordering Rules

This section describes the MCH ordering rules for transactions progressing through the PCI Express unit.

7.5.10.1 Inbound Transaction Ordering Rules

Inbound transactions originate from PCI Express and target main memory. In general, the PCI Express cluster holds inbound transactions in FIFO order. There are exceptions to this order under certain situations. For example, PCI Express requires that read

completions are allowed to pass read requests. This forces any read completions to bypass any reads which might be back pressured in the queue. The PCI Express ports have no ordering relationship to each other (aside from the peer-to-peer restrictions below).

Sequential non-posted requests are not required to be completed in the order they were requested. However, if a non-posted request requires multiple sub-completions (typically due to splitting a memory read into cache line requests), then those sub-completions must be delivered in order.

Inbound writes cannot be posted beyond the PCI Express domain and outbound writes may only be posted after the write is acknowledged by the destination PCI Express cluster. The posting of writes relies on the fact that the system maintains a certain ordering relationship. Since the MCH cannot post inbound writes beyond the PCI Express cluster, the MCH must wait for snoop responses before issuing subsequent, order-dependent transactions.

7.5.10.1.1 Inbound Ordering Requirements

In general, there are no ordering requirements between transactions issued on the different PCI- Express interfaces. The following rules apply to inbound transactions issued on the same interface.

The following rules must be ensured for inbound transactions:

Rule 1. Outbound non-posted read and write completions must be allowed to progress past stalled inbound non-posted requests.

Rule 2. Inbound posted write requests must be allowed to progress past stalled inbound non-posted requests.

Rule 3. Inbound posted write requests, inbound read requests, outbound non-posted read and write completions cannot pass enqueued inbound posted write requests.

The Producer - Consumer model prevents read requests, write requests, and non-posted read or write completions from passing write requests. Refer to *PCI Local Bus Specification*, Revision 2.3 for details on the Producer - Consumer ordering model.

Rule 4. Outbound non-posted read or write completions must push ahead *all* prior inbound posted write transactions from that PCI Express port.

Rule 5. To optimize performance, Inbound, coherent, posted writes will issue ownership requests (RFO) without waiting for prior ownership requests to complete.

Rule 6. Inbound messages follow the same ordering rules as inbound posted writes.

Inbound messages are listed in Table 7-16. Similarly to inbound posted writes, reads should push these commands ahead.

The above rules apply whether the transaction is coherent or non-coherent. Some regions of memory space are considered non-coherent (Don’t Snoop attribute is set). The MCH PCI Express cluster will order all transactions regardless of its destination.

7.5.10.2 Outbound Transaction Ordering Rules

Outbound transactions through the MCH are memory, I/O, or configuration read/write transactions originating on a processor interface destined for a PCI Express device. Multiple transactions destined for the same PCI Express port are ordered according to the ordering rules specified in *PCI Express Base Specification*, Revision 1.0a.

7.5.10.2.1 Outbound Ordering Requirements

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

Rule 1. Inbound non-posted completions must be allowed to progress past stalled outbound non-posted requests.

Rule 2. Outbound posted write requests must be allowed to progress past stalled outbound non-posted requests.

Rule 3. Outbound non-posted requests, outbound messages, outbound write requests, and inbound completions cannot pass enqueued outbound posted write requests.

The Producer - Consumer model prevents read requests, write requests, and read completions from passing write requests. Refer to *PCI Local Bus Specification*, Revision 2.3 for details on the Producer - Consumer ordering model.

Rule 4. Posted outbound messages must follow the same ordering rules as outbound posted writes.

Rule 5. If a non-posted inbound request requires multiple sub-completions, then those sub-completions must be delivered in linearly increasing address order.

7.5.10.3 MCH Ordering Implementation

The following table summarizes the rules enforced on transactions from a given PCI Express port by the MCH.

Table 7-22. MCH Ordering Implementation

Transaction	Will the transaction pass a stalled Posted Request?	Will the transaction pass a stalled Non-Posted Request?	Will the transaction pass a stalled completion?
Posted requests	never	always	always
Non-Posted Requests	never	Can happen in implementation; no architectural ordering requirement is imposed	always
Completions	never	always	never

7.5.10.4 Peer-to-Peer Ordering

All peer-to-peer memory write transactions are treated as non-coherent memory writes by the system. Peer-to-peer memory reads are treated as non-coherent reads.

On the MCH, any peer-to-peer transaction is ordered with other inbound transactions from the same PCI Express port. This provides a serialization point for proper ordering (e.g. cases where the flag and data are not in the same memory).

When the PCI Express interface receives a peer-to-peer memory write command, inbound ordering rules require that it must wait until all prior inbound writes are globally visible. The peer-to-peer write completes when the target PCI Express port receives the transaction in its ordered domain. The acknowledge must return to the source PCI Express unit quickly enough to allow the PCI Express device to post further peer-to-peer memory writes without any stalls on the interface.

Peer-to-peer memory write transactions are considered posted with regards to ordering. Peer to peer read transactions are non-posted. Peer-to-peer transactions must adhere to the ordering rules listed in Section 7.5.10.1.1 and Section 7.5.10.2.1.

7.5.10.5 Interrupt Ordering Rules

With MSI, SAPIC and Expiate, interrupts are simply inbound non-coherent writes to the processor. With legacy interrupts, the interrupts are ASSERT and DEASSERT messages (also following posted write ordering rules). This enforces that the interrupt will not be observed until all prior inbound writes are flushed to their destinations. However, the MCH does not guarantee that the interrupt will be observed by the processor before subsequent writes are visible to a processor.

7.5.10.5.1 EOI Ordering

When a processor receives an interrupt, it will process the interrupt routine. The processor will then proceed to clear the I/O card's interrupt register by writing to that I/O device. In addition, for level-triggered interrupts, the processor sends an End-of-Interrupt (EOI) special cycle (8 bit interrupt vector on D[7:0]# of the processor's data bus) to an IOAPIC controller south of MCH. This EOI cycle must be treated as an outbound write with regard to ordering rules. This ensures that the EOI will not pass any prior outbound writes. If the EOI passes the prior write to clear the register, then the IOAPIC controller could mistakenly signal a second interrupt since the register clear had not occurred yet.

7.5.11 Prefetching Policies

The MCH does not perform any speculative prefetching for PCI Express interface component reads. The PCI Express component south of the MCH is solely responsible for its own prefetch algorithms since those components are best suited to know what tradeoffs are appropriate.

The MCH does not perform any outbound read prefetching.

7.5.12 No Isochronous Support

The MCH does not support isochrony. Only the default virtual channel (channel 0) is supported on the PCI Express interfaces.

7.5.13 PCI Express Hot-Plug

PCI Express native Hot-Plug allows for higher availability and serviceability of a server. It gives the user the capability of adding, removing, or swapping out a PCI Express slot device without taking down the system. The user and system communicate through a combination of software and hardware utilizing notification through mechanical means and indicator lights.

Each MCH PCI Express port supports the optional Hot-Plug capability described in *PCI Express Base Specification*, Revision 1.0a. The PCI Express Hot-Plug model implies a Hot-Plug controller per port which is identified to software as a capability of the peer-to-peer bridge configuration space.

PCI Express Hot-Plug support requires that the MCH supports a set of Hot-Plug messages, listed in Table 7-16 and Table 7-20, to manage the states between the Hot-Plug controller and the device.

The PCI Express form factor has an impact on the level of support required from the MCH. For example, some of the Hot-Plug messages are required only if the LED indicators reside on the actual card and are accessed through the endpoint device. The MCH supports all of the Hot-Plug messages.

Refer to Section 7.12.6, "PCI Express Hot-Plug Support, VPP SMBus" on page 444 for further details.

7.5.14 Error Handling

The Intel® 7300 Chipset supports the optional Advanced Error Handling capability in the *PCI Express Base Specification*, Revision 1.0a. The MCH can receive error indications with two mechanisms:

- Error Messages

In-band messages are issued for errors that occur with posted transactions and event that are not synchronous to a transaction's completion.

- Completion Status

For non-posted transactions, errors are indicated through the completion status field on the PCI Express completion.

7.5.14.1 Unsupported Transactions and Unexpected Completions

If the MCH receives a packet that is not included in Table 7-16, then the MCH treats that packet as an unsupported transaction. If the transaction is non-posted (determined after fully decoding the header), then the completion is issued by the MCH with an Unsupported Request status. If the transaction is posted, it is dropped. The following errors are also treated as Unsupported Transactions

- Lock to non-legacy device
- Illegal configuration accesses

7.5.14.2 Error Actions:

- **Receiving an UnSupported Request**
  If Non Posted Request then set UR Status in response;
  Set PEXSTS.RMA, PEXDEVSTS.URD
  Set UNCERRSTS.IO2 and assert error signal per the PEXDEVCTRL and the PEX_ERR_DOCMD register.

- **Receiving a supported request type but cannot complete**
  If Non Posted Request then send Completer Abort in response.
  Set PEXSTS.RTA
  Set UNCERRSTS.IO7 and assert error signal per the PEXDEVCTRL and the PEX_ERR_DOCMD register.

7.5.14.3 ECRC

The *PCI Express Base Specification*, Revision 1.0a supports an optional end-to-end CRC mechanism (ECRC). The MCH does not support this feature.

7.5.14.4 EDB

The *PCI Express Base Specification*, Revision 1.0a supports a mechanism to signal a bad packet (End Bad or EDB). The MCH can receive a packet marked with as EDB and will correctly identify the packet as "bad", but the MCH will never generate a EDB packet.

7.5.14.5 Error Forwarding

PCI Express has a concept called Error Forwarding or Data Poisoning. This feature allows a PCI Express device to forward data errors across the interface without it being interpreted as an error originating on that interface.

When the MCH is the initiator of a transaction with data (write or read completion), the MCH poisons the transaction if the data contains an internal ECC error. Poisoning is accomplished by the MCH setting the EP bit in the PCI Express packet header.

When the MCH is the recipient of a transaction with data and the data is poisoned (refer to *PCI Express Base Specification*, Revision 1.0a for the various mechanisms) then the PCI Express unit poisons the data before forwarding internally to the destination interface.

7.5.14.6 Unconnected Ports

If a non-posted transaction targets a PCI Express interface that is not connected to any device, the MCH behaves as it would if it had received an Unsupported Request completion (master abort). The MCH should complete any reads initiated on the MCH (from the processor bus, or JTAG, SMBus) to an unconnected PCI Express port with all ones in the data. If the transaction is posted, the transaction is dropped and the signal is driven.

Note: Design must comprehend the difference between L2 state and an unconnected port. In order to exit L2, a configuration cycle will target the port and must not trigger the above mechanism.

7.5.15 PCI Express Power Management Support

The *PCI Express Base Specification*, Rev 1.0a requires that root complexes enable an end point to enter low power states. PCI Express power management happens in two phases:

- The PCI Express link transitions from L0 (active state) to L1 (low power)
- The PCI Express link transitions from L1 (low power) to L2 (end point power off) through L0. i.e. L0 --> L1 --> L0 --> L2 etc.

While it is not required that the MCH enter low power states, the end point also participates in a handshake protocol to wake up a sleeping system. The MCH must honor this protocol which is described further below.

Note: The MCH will not initiate a PME event but will forward PME messages to wake the system due to a Hot-Plug event on the PCI Express interface, for example.

7.5.15.1 L0s

The MCH does not support Active State PM transition (ASPM) into L0s state (i.e MCH does not power down link into a "standby" mode and will not transmit electrical_idle ordered sets) due to latency/performance issues that degrade performance while presenting minimal power savings. However, it must tolerate an end PCI Express device from placing the link into an L0s state. The lack of L0s support in the MCH is a deviant from the base *PCI Express Base Specification*, Rev 1.0a as it simplifies power control and onboard logic circuitry.

Note: The MCH has recently added a mechanism through the PEXDLLPCTRL.GBLFCUT timer to prevent entry into L0s[1] by injecting FC credit DLLPs at less than 7 µs interval.

7.5.15.2 L1 Entry

Refer to the *PCI Express Base Specification*, Rev 1.0a for details. This section describes at a high level what is required of the MCH.

The *PCI Express Base Specification*, Rev 1.0a lists two mechanisms for entering L1 state:

- Active State Power Management
- Software Initiated Power Management

Note: The MCH fully supports Software Initiated Power Management but does not initiate Active State Power Management. The L1 state is initiated with a configuration write to the end device being put into the low power state. The configuration write is executed by software writing to the "Power State" bits[1:0] of the PMCSR register of the PCI-PM capability structure in the end device such that the end point goes into the D1 state. In response to this configuration write, the end point will block subsequent TLPs and wait until all pending TLPs have Acked. When this condition is met, the end point will issue a PM_ENTER_L1 DLLP to the MCH to put the link in L1 mode. The MCH responds to this message by blocking outbound TLPs and waiting for all "in-flight" TLPs to get acknowledged. After all pending TLPs have been ACKed, the MCH issues a PM_REQUEST_ACK DLLP continuously until the link is observed to be electrically idle. The link is now in the L1 state.

7.5.15.3 L1 Exit

Exit from L1 brings the link back into the active (L0) state. When the MCH receives a configuration write to a link that is in L1, the link is trained again. This is done by configuration software setting the Retrain Link bit in the PEXLNKCAP register. When the link is retrained and is reactivated in the L0 state, the MCH will forward the configuration write to the end point to perform a similar link retraining.

7.5.15.4 L2/L3 Ready State Entry

The L2/L3 Ready state is entered when software wants to remove power from a device. If the end point continues to operate under Vaux power, its ending state will be L2. If the end point truly powers down, its ending state will be L3. The L2/L3 Ready state is initiated with a configuration write to PEXGCTRL.PME_Turn_Off (See Section 4.8.10.33, "PEXGCTRL: PCI Express Global Control Register"). When this bit is set, the MCH broadcasts a PM_TURNOFF message to all populated PCI Express ports. The MCH waits for a PM_TO_ACK message from each of the ports that were sent the PM_TURNOFF message.

After the endpoint sends a PM_TO_ACK, it will initiate L2/L3 Ready state entry with a PM_ENTER_L2 DLLP. From this point on, the link handshake follows the L1 Entry flow.

Once all of the PM_TO_ACK messages are received, the MCH sets the PEXGCTRL.PME_TO_Ack. Software can poll the register PEXGCTRL.PME_TO_Ack register field to determine when all the PM_TO_ACK messages have arrived, take appropriate action and reset the field.

7.5.15.5 Wake Sequence

The *PCI Express Base Specification*, Rev 1.0a allows an endpoint to wake up a system that is asleep (in L2). This sequence is split into two elements:

- Waking the system (powering it up)
- Alerting the system that a device has awakened

To power the system and devices, Intel® 7300 Chipset platforms require the side-band WAKE# signal. The inband tone mechanism is not supported. The WAKE# signal is driven by the endpoint directly to on-board logic which powers the device(s) up. The L2 state implies that power is off to the devices and therefore, exit from this state implies that power is reapplied and the links would simply retrain according to the platform reset requirements.

After the links are trained, the waking device issues a PM_PME message to the MCH. This message carries the DeviceID of the initiator which is logged in the PEXRTSTS register. The MCH uses the PMESTATUS bit in the PEXRTSTS register to reflect the PME state for each of the PCI Express ports. The MCH OR's all the bits together and sends the Assert_PMEGPE message to the Intel® 631xESB/632xESB I/O Controller Hub. When all the bits are clear, the Deassert_PMEGPE message is sent.

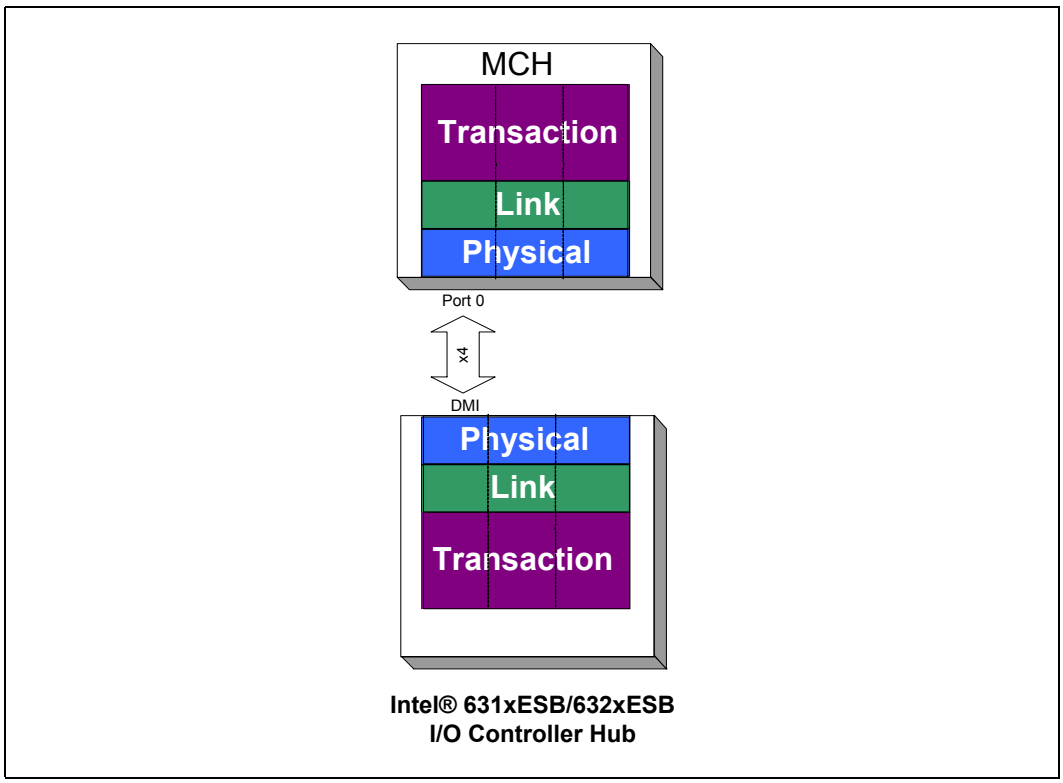
Note: The MCH can handle two outstanding PM_PME messages in its internal queues of the Power Management Controller per port. If the downstream device issues more than 2 PM_PME messages successively, it will be dropped.

Note: If the MCH receives a pending request to a port that is in L2 (for e.g. an outbound transaction), it will be a master abort and the respective error logged.

7.6 Enterprise South Bridge Interface (ESI)

The Enterprise South Bridge Interface (ESI) in the MCH is the chip-to-chip connection to the Intel® 631xESB/632xESB I/O Controller Hub see Figure 7-24. The ESI is an extension of the standard PCI Express specification with special commands/features added to enhance the PCI Express interface for enterprise applications. This high-speed interface integrates advanced priority-based servicing allowing for concurrent traffic and true isochronous transfer capabilities. Base functionality is completely transparent permitting current and legacy software to operate normally. For the purposes of this document, the Intel® 631xESB/632xESB I/O Controller Hub will be used as a reference point for the ESI discussion in the MCH.

Figure 7-24. MCH to Intel® 631xESB/632xESB I/O Controller Hub Enterprise South Bridge Interface



The ESI port in the MCH may be combined with two additional PCI Express ports to augment the available bandwidth to the Intel® 631xESB/632xESB I/O Controller Hub. When operating alone the available bi-directional bandwidth to the Intel® 631xESB/632xESB I/O Controller Hub is 2GB/s (1GB/s each direction). When the ESI is pared with 2 additional x4 PCI Express links the available bi-directional bandwidth to the Intel® 631xESB/632xESB I/O Controller Hub is increased to 6 GB/s.

Note: ESI Port does not support polarity inversion, lane reversal or lane degradation

7.6.1 Peer-to-Peer Support

Peer-to-peer support is defined as transactions which initiate on one I/O interface and target another without going through main memory. The MCH ESI supports peer-to-peer transactions for memory transactions only. The compatibility interface can be the destination of a peer-to-peer write or read except that peer-to-peer posted writes targeting LPC in Intel® 631xESB/632xESB I/O Controller Hub are not allowed (to prevent PHOLD deadlocks). The compatibility interface can be the source of a peer-to-peer read/write. Non-posted requests may prefetch into MMIO (with potential side effects). Peer-to-peer transactions are not observed on any interface except the target and destination (e.g. no processor bus snoops).

Inbound coherent transactions and peer-to-peer transactions must maintain ordering rules between each other. Peer-to-peer transactions follow inbound ordering rules until they reach the head of the inbound queue. Once the transaction reaches the head of the inbound queue, the MCH routes the transaction to the next available slot in the

outbound queue where PCI ordering is maintained until the end of the transaction. The MCH does not support peer-to-peer where the source and destination is the same PCI Express interface.

7.6.2 Power Management Support

The Intel® 631xESB/632xESB I/O Controller Hub provides a rich set of power management capabilities for the operating system. The MCH receives PM_PME messages on its standard PCI Express port and propagates it to the Intel® 631xESB/632xESB I/O Controller Hub over the ESI as an Assert_PMEGPE message. When software clears the PEXRTSTS.PME Status register bit, in the PEXRSTSTS[7:0] PCI Express Root Status Register, after it has completed the PME protocol, the MCH will generate a Deassert_PMEGPE message to the Intel® 631xESB/632xESB I/O Controller Hub. The MCH must also be able to generate the Assert_PMEGPE message when exiting S3 (after the reset). The PMGPE messages are also sent using a wired-OR approach.

Note: The MCH does not stay in the S3 state. It changes to S3 before transitioning into deeper Sleep states.

7.6.2.1 Rst_Warn and Rst_Warn_Ack

The Rst_Warn message is generated by the Intel® 631xESB/632xESB I/O Controller Hub as a warning to the MCH that it wants to assert PLTRST# before sending the reset. In the past, problems have been encountered due to the effects of an asynchronous reset on the system memory states. Since memory has no reset mechanism itself other than cycling the power, it can cause problems with the memory's internal states when clocks and control signals are asynchronously tri-stated or toggled, if operations resume following this reset without power cycling. To protect against this, the Intel® 631xESB/632xESB I/O Controller Hub will send a reset warning to the MCH. The Advanced Memory Buffer is supposed to handle putting the DIMMs into a non-lockup state in the event the link "goes down" in the middle of DDR2 protocol. The MCH is NOT required to place quiesce the DRAM's prior to reset.

The MCH completes the handshake by generating the Rst_Warn_Ack message to the ICH6 at the earliest.

7.6.2.2 STPCLK Propagation

The Intel® 631xESB/632xESB I/O Controller Hub has a sideband signal called STPCLK. This signal is used to place IA-32 CPUs into a low power mode. Traditionally, this signal has been routed directly from the I/O Controller Hub to the CPUs.

In future ESBx components, the plan is to rearchitect the mechanism for alerting the CPUs of a power management event. However, this chipset (using Intel® 631xESB/632xESB I/O Controller Hub) will require the same method used for past server chipsets (route STPCLK on the board as appropriate). The MCH will not provide any in-band mechanisms for STPCLK.

7.6.3 Special Interrupt Support

The Intel® 631xESB/632xESB I/O Controller Hub integrates an I/O APIC controller. This controller is capable of sending interrupts to the processors with an inbound write to a specific address range that the processors recognize as an interrupt. In general, the compatibility interface cluster treats these no differently from inbound writes to DRAM. However, there are a few notable differences listed below.

7.6.4 Inbound Interrupts

To the MCH, interrupts from the Intel® 631xESB/632xESB I/O Controller Hub are simply inbound non-coherent write commands routed to the processor buses. The MCH does not support the serial APIC bus.

7.6.5 Legacy Interrupt Messages

The ESI and PCI Express interfaces support two methods for handling interrupts: MSI and legacy interrupt messages. The interrupt messages are a mechanism for taking traditionally out-of -band interrupt signals and using in-band messages to communicate. Each PCI Express interface accepts up to four interrupts (A through D) and each interrupt has an assert/deassert message to emulate level-triggered behavior. The MCH effectively wire-ORs all the INTA messages together (INTBs are wire-ORed together, etc.).

When the MCH accepts these PCI Express interrupt messages, it aggregates and passes the corresponding "assert_intx" messages to the Intel® 631xESB/632xESB I/O Controller Hub's I/OAPIC with from the PCI Express ports (wired-OR output transitions from 0⇒1) mechanism. When the corresponding deassert_intx message is received at all the PCI Express ports (wired-OR output transitions from 1⇒0), the "deassert_intx" message is sent to ESI port.

7.6.6 End-of-Interrupt (EOI) Support

The EOI is a specially encoded processor bus transaction with the interrupt vector attached. Since the EOI is not directed, the MCH will broadcast the EOI transaction to all I/O(x)APICs. The MCH PEXCTRL.DIS_APIC_EOI bit per PCI Express port can be used to determine whether an EOI needs to be sent to a specific port.

7.6.7 Error Handling

Table 7-37 describes the errors detected on ESI through the standard PCI Express and Advanced error reporting mechanism.

7.6.7.1 Inbound Errors

In general, if an inbound read transaction results in a Master Abort (unsupported request), the compatibility interface cluster returns a Master Abort completion with data as all ones. Likewise, for a Target Abort condition, the ESI cluster returns a Target Abort completion with data as all ones. If a read request results in a Master or Target Abort, the MCH returns the requested number of data phases with all ones data.

Master aborted inbound writes are dropped by the MCH, the error is logged, and the data is dropped.

If the MCH receives an inbound unsupported Special Cycle message it is ignored and the error condition is logged. If the completion required bit is set, an Unsupported Special Cycle completion is returned.

7.6.7.2 Outbound Errors

It is possible that the compatibility interface cluster will receive an error response for an outbound request. This can include a Master or Target Abort for requests that required completions. The MCH might also receive an "Unsupported Special Cycle" completion.

7.7 Intel® QuickData Technology

The Intel® 7300 Chipset incorporates Intel® QuickData Tchnology which provides a high performance and scalable DMA engine for transferring data from memory using the "push model' for applications such as TCP/IP, RAID etc. that commonly occur in network/communication/storage segments.

7.7.1 Integrated DMA Overview

Direct Memory Access (DMA) is a scheme used by I/O devices to transfer large blocks of data from memory using the traditional "pull" model (reads). For such an access, system performance is dictated by several factors such as inbound memory access latency, number of descriptors, size of the transfers and the amount of pipelining in the Chipset. By designing a DMA engine that is closer to memory in the north bridge and adding extra features, data can be efficiently retrieved and sent downstream using the more efficient bandwidth optimized "push" model (writes) leading to higher overall system performance and minimizing the load on the Chipset. In addition, such a DMA engine's capability can be exploited for performing network protocol operations such as TCP/IP through host adapters commonly called as TOE (TCP/IP Off load Engines). The DMA engine in Intel® 7300 Chipset supports 4 independent channels. In addition, the DMA engine has built-in intelligence for features such as chain descriptors, stream priority initialization in the PCI_Express ports, interrupt/MSI generation, data (byte) alignment, automatic status updates to destination and error signaling. With these mechanisms, data can be moved within system memory or from system memory to the memory mapped I/O as desired providing a low-latency/high throughput path.

To use the DMA capability, a linked list of descriptors is built in the local system memory by the controlling process prior to starting the process. Each of these descriptors contain information on the source, destination addresses, length of data transfer (in bytes) and pointer to the next descriptor block. Using this generalized method, data can be copied from one local system memory location to another, or from local system memory to MMIO space.

The DMA engine optimizes block transfer through the linked-list based descriptor chain and supports scatter/gather operations along with error handling if any, during the DMA operation. The linked list of chain descriptors are issued to the DMA channel by initializing the chain descriptor address and ringing a doorbell. This allows the CPU or I/O subsystem to do a "fire and forget" type of block data transfer. The DMA engine in the chipset is responsible for executing the scatter/gather operation and initiates traffic on the system memory and/or the PCI-Express ports based on the programmed context to fetch and send data. It then signals status updates or generates an interrupt on successful completion.

7.7.2 Features

The following features are supported by the DMA engine in the Intel® 7300 Chipset:

- Four Independent DMA Channels
  - — Dedicated data transfer queue per channel
  - — Full register set for descriptor and transfer handling per channel
- Support data transfer between two system memory locations, or from system memory to MMIO.
- 64-bit addressing on the system memory interface and I/O subsystem of which the MCH will internally use 40-bits (1 TB) of addressing.
- Maximum transfer length of 4KB/1MB per DMA descriptor block.

- Intel® QuickData Technology Configuration space mapping for DMA and per port write combining
- Fully programmable MMIO space for channel-specific register sets to enable faster access for CPU and I/O device to descriptor region in main memory.
- Dynamically appending a descriptor (or chain of descriptors) to the end of current executing DMA descriptor chain (Chain mode DMA)
- Splicing in a descriptor (or chain of descriptors) at the current location in the DMA descriptor chain.
- Byte aligned data transfer from source to destination on the System Memory Interface and MMIO. Data Transfer (writes) on PCI-Express ports (for MMIO) will be up to maximum supported payload of the target[1] per TLP with appropriate byte enables set for handling byte offsets at the boundaries
- Both coherent and non-coherent memory transfer on a per descriptor basis with independent control of coherency for source and destination.
- Programmable mechanisms for signaling the completion of a descriptor by generating an MSI 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.

7.8 Trusted Platform Module (TPM)

7.8.1 TPM Overview

Establishing trust between two or more systems is an increasingly important requirement today for e-commerce, banking, and many other applications. The Trusted Platform Module (TPM) is an attempt to address this security concern in an open interoperable standard. The TPM is a self-contained integrated circuit device that is soldered directly on the motherboard. It is designed specifically to protect against software-based attacks that would compromise a system. TPM information cannot be compromised by potentially malicious host system software.

TPM security features are as follows:

- Protects key security operations and tasks in a protected environment
- Provides non-volatile storage for encryption keys and other critical security information
- Provides mechanism to securely attest for level of security of the host system
- Provides mechanism to determine if boot-up configuration parameters have changed or have been corrupted

For more details on the usage model or TPM specific information, please visit the Trusted Computing web page at http://www.trustedcomputinggroup.com and refer to "TCG PC Client Specific TPM Interface Specification, Version 1.2".

The Intel® 7300 Chipset supports TPM v 1.2 locality 0 access from CPU on the system. TPM locality 0 access is mapped to FED4_0xxx MMIO address range. The MCH does not support Intel® Trusted Execution (TXT). However, with TPM, MCH allows security to exist within the platform. The TPM is accessed through address space FED4_0xxxh. chipset will only open FED4_0xxxh - FED4_0FFFh to processors in the system. It is

1. The maximum payload size for the MCH is 256B for read completions and write TLPs on the PCI-Express port. However, the MCH limits the outbound MMIO writes to 64B maximum.

available at all times and is capable of being accessed by all processors. The MCH will convert these CPU accesses into corresponding LT_Read/Write messages on the ESI port and send it to Intel® 631xESB/632xESB I/O Controller Hub.

Refer to the *Enterprise Southbridge Interface (ESI) Specification* for details on the LT Read/Write cycle encoding (type/format). Only the LT Memory Read is non-posted and requires a completion. However, LT Memory Writes are posted. The completion for the LT Memory Read will be the same as for normal read completion cycles. If a zero length read/write is targeted to 0xFED4_0xxx space, the MCH has the ability to either complete the request internally or send the request south of the ESI port.

The MCH will decode TPM locality 1-4 addresses on the FSB and they will be routed to the ESI port as standard Memory transactions. The same rule also applies to Peer-to-Peer accesses for locality 1-4 accesses.

Locked accesses to 0xFED4_0xxx range are not supported and will be master aborted by the MCH. Since the MCH does not have an internal master abort mechanism, it will be passed as MMIO transactions to ESI. All peer-to-peer requests that attempt to access the TPM space will be allowed as normal memory transactions on the ESI port (i.e. it will not be converted LT-Read/ Write). Refer to Table 7-23. Such a peer-to-peer transaction will be master aborted by the Intel® 631xESB/632xESB I/O Controller Hub port to the originating port in the MCH.

Table 7-23. Decode Table in Intel® 7300 Chipset for TPM Locality

Address Range	Transactions on FSB	Transactions on ESI
FED4_0000 - FED4_0FFF (Locality 0)	Memory Rd/Write	LT_Read/Write (Memory Rd/Write for Peer-to-Peer)
FED4_1000 - FED4_1FFF (Locality 1)	Memory Rd/Write	Memory Rd/Write
FED4_2000 - FED4_2FFF (Locality 2)	Memory Rd/Write	Memory Rd/Write
FED4_3000 - FED4_3FFF (Locality 3)	Memory Rd/Write	Memory Rd/Write
FED4_4000 - FED4_4FFF (Locality 4)	Memory Rd/Write	Memory Rd/Write

Note: The MCH will forward zero length MMIO read/write requests that originate from the FSB targeting the 0xFED4_0xxx range (ring 0) access to an appropriate LT-Read/Write memory cycle of zero length on ESI. It is the responsibility of the Intel® 631xESB/632xESB I/O Controller Hub to convert these transactions appropriately or internally complete the transaction back to the MCH. In addition, the MCH as a component does not impose any transaction length1 or address alignment restrictions2 for these MMIO reads/writes when they are mapped to LT transactions on ESI. They are passed through as they appear on FSB but as LT transactions on the ESI. The MCH supports attribute aliasing (WC,UC) for these CPU accesses but it is the responsibility of the software to employ the correct attributes as the case may be and should be cognizant of the inherent ordering/ serialization issues in the CPU. These requests are deferred on the FSB by the MCH and it is entered into the ordering domain only after it place in IOU for dispatch to the ESI port. Also note that the ESI port in the MCH does not support Chipset Write coalescing (CSWC). (i.e does not combine smaller packets to form a larger TLP)

Note: It is recommended that software use the "UC" attribute for the 0xFED4_0xxx range.

7.9 Direct Cache Access (DCA)

7.9.1 DCA Introduction

Direct Cache Access (DCA) mechanism is a system level protocol in a multi-processor system to improve I/O network performance by either placing the data from the I/O devices directly into a CPU cache or by providing hints to the processor to perform a data prefetch and install it in its local caches. The basic goal is to minimize the likelihood of a cache miss for the application when a demand read is executed and provide higher system performance.

Since efficient delivery of I/O data to the host is a fundamental and increasing performance concern for Multi-gigabit Ethernet I/O cards, the DCA protocol provides a method to circumvent this problem and boost overall system performance through close interaction amongst the CPU, Chipset and I/O cards by passing relevant hints.

7.9.2 Features

The following features are supported by the DCA engine in Intel® 7300 Chipset:

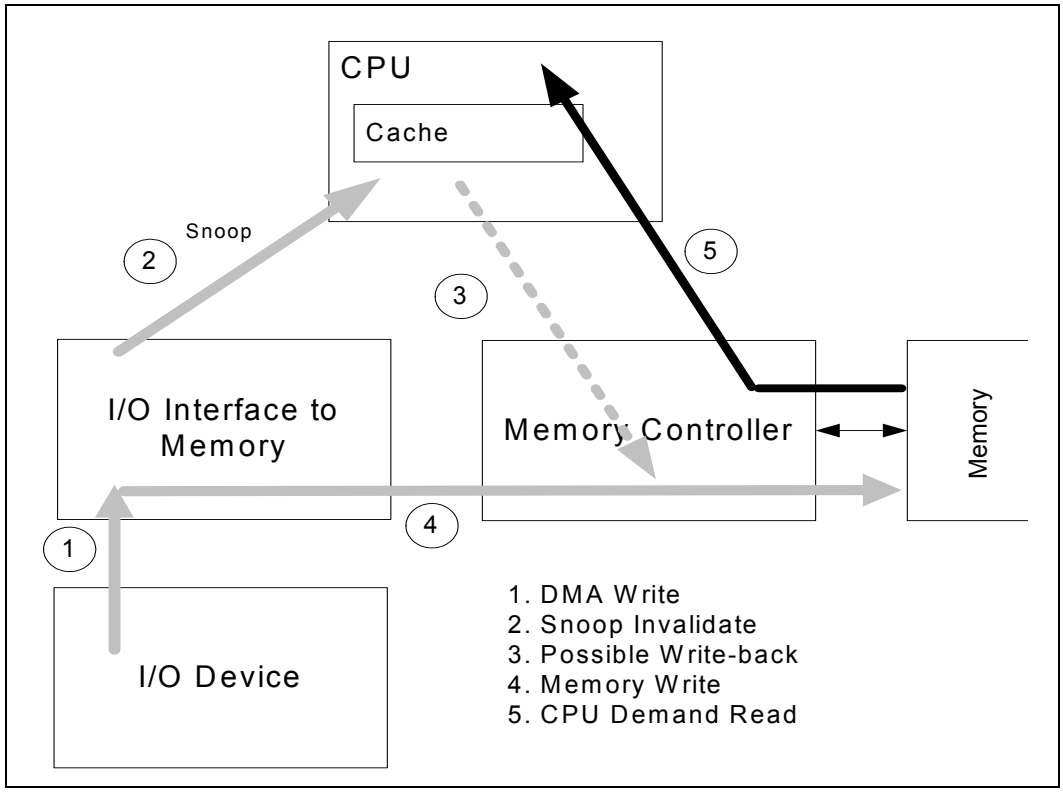
- DCA routing information through PCI-Express inbound write messages encoded in the Tag[4:0] field of the TLP initialized by the end device.
- FSB BIL-Hint Transaction generated on the target bus
- DCA with SF bypass mode
- DCA feature can be enabled/disabled for all PCI-Express ports.
- Distinguish DCA and non DCA traffic from end device through Hardware
- Authentication (DCA_REQID) using IQD MMIO registers in the MCH.

7.9.3 DCA Overview

DCA promotes efficient data transfer between agents that communicate using shared, cacheable memory. The concept of DCA is applicable to workloads where the source of data transfer has specific knowledge of where and how soon the data will be 'consumed'. The DCA implementation in the MCH uses a “hint to pull data” model where data from the I/O card is written to memory but a hint is passed concurrently by the Chipset to the CPU of interest such that the selected processor can execute a hardware prefetch and obtain the data prior to the demand read.

A prevalent usage model for DCA is network interfaces in the context of high server network bandwidth requirements and the availability of multi-gigabit network interfaces. Packetized data from the network is transferred to memory by a network controller using traditional Direct Memory Access (DMA) transfers. This data is read and processed by the CPU as it executes driver code or network packet processing code. DCA helps by reducing CPU read latency and memory bandwidth using a new protocol that permits data to be moved directly into the CPU's cache. Note that DCA focuses on cacheable memory based interaction between two agents. DCA is not applicable to other forms of communication such as interrupts or mechanisms that use un-cacheable memory.

Figure 7-25. Inbound Write Flow in a typical MCH without prefetch hint



7.9.4 Write Protocol without DCA

In Figure 7-25, "Inbound Write Flow in a typical MCH without prefetch hint" on page 423, the source of the data is an I/O device such as a network controller connected through a PCI-Express fabric to a generic MCH. The sequence is described as follows:

1. The I/O device sends an inbound Mem_Write transaction via the PCI_Express port connected to the MCH.

2. The MCH initiates relevant operations on the FSB to get ownership of the line through snoops (BIL).

3. If the cache line is in a Modified (M) state in any of the CPU's caches, then the data is evicted (write-back) from the CPU.

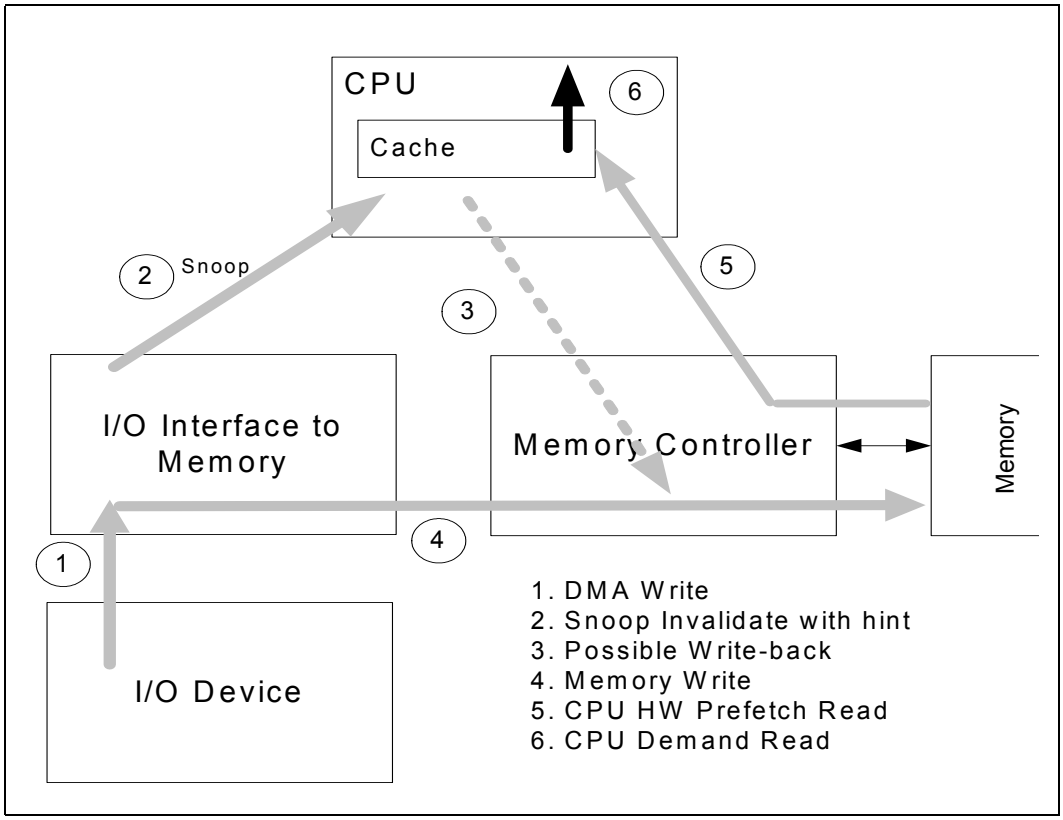
4. If the inbound write transaction is a full cache-line write, then the CPU write-back is overwritten by the incoming data in the MCH enroute to memory. If the inbound write transaction is a partial cache-line transaction, then it must be merged with the full cache-line evicted by the CPU (M). If the FSB snoop is clean, then incoming write data is written to memory.

5. The write update by the I/O device is followed shortly by a CPU demand read - a read that must be satisfied for program execution to make forward progress compared to a speculative prefetch. This demand read in the CPU due to an L2/L3 internal cache miss requires a long roundtrip latency for completion (traditional pull model) and hence affects system performance and increases CPU overhead.

7.9.5 Write Protocol with DCA

Figure 7-26, "DCA hint to pull Model" on page 424 illustrates how DCA can improve system efficiency. The snoop invalidate shown in Step 2 of Figure 7-26, is sent with hint information to the CPU. In Step 3, a HITM response may be obtained for the snoop and in Step 4, the write data is committed to memory. In Step 5, the CPU responds by issuing a hardware prefetch without interrupting any program running in its cores. The Hardware prefetch uses the same address in the snoop invalidate transaction to bring new data into the CPU's cache. At Step 6, when the demand read from the CPU arrives, it will find the data in the cache (hit). This minimizes the latency and improves throughput.

Figure 7-26. DCA hint to pull Model



7.10 Power Management

The MCH power management support includes:

- ACPI supported
- System States: S0, S1, S4, S5, C0, C1, C2

7.10.1 Supported ACPI States

The MCH supports the following ACPI States:

- Processor
  - C0: Full On.
  - C1: Auto Halt.
  - C2 Desktop: Stop Grant. Clock to processor still running. Clock stopped to processor core.
- System
  - G0/S0: Full On.
  - G1/S1: Stop Grant, Desktop S1, same as C2.
  - G1/S2: Not supported.
  - G1/S3: Not supported.
  - G1/S4: Suspend to Disk (STD). All power lost (except wake-up logic on Intel® 631xESB/632xESB I/O Controller Hub).
  - G2/S5: Soft off. Requires total system reboot.
  - G3: Mechanical Off. All power lost (except real time clock).

7.10.2 FB-DIMM Thermal Management

The MCH implements the following thermal management mechanisms. These mechanisms manage the read and write cycles of the system memory interface to implement thermal throttling.

7.10.2.1 Hardware-Based Thermal Management

The number of hex-words transferred over the DRAM interface are tracked per row. The tracking mechanism takes into account that the DRAM devices consume different levels of power based on cycle type (page hit/miss/empty). If the programmed threshold is exceeded during a monitoring window, the activity on the DRAM interface is reduced. This helps in lowering the power and temperature.

7.10.2.2 Software-Based Thermal Management

This is used when the external thermal sensor in the system interrupts the processor to engage a software routine for thermal management.

7.10.3 FB-DIMM Thermal Diode Overview

The FB-DIMM Advanced Memory Buffer (AMB) contains an internal thermal diode to measure AMB / DIMM temperature. Upon detecting a thermal over temperature condition the AMB initiates a thermal throttling event. For more information see the *Gold Bridge Component External Design Specification*.

7.11 Power Up and System Reset

The MCH is the root of the I/O subsystem tree, and is therefore responsible for general propagation of system reset throughout the platform. The MCH must also facilitate any specialized synchronization of reset mechanisms required by the various system components.

7.11.1 MCH Power Sequencing

General power sequencing requirements for the MCH are simple. In general higher voltages must come up before lower voltages. Figure 7-27 depicts the sequencing of the three main voltages powering the MCH.

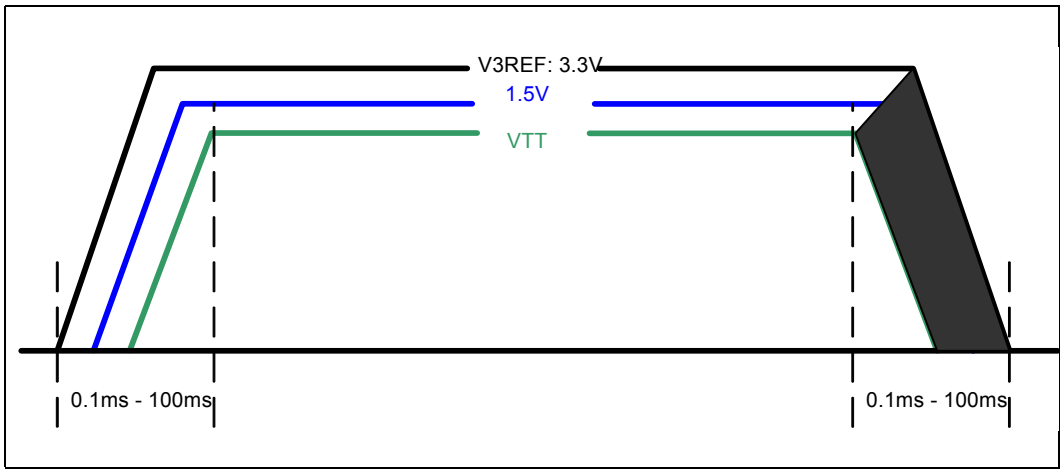
Power supplies must ramp UP in the following order (from first to last in a system environment):

1. The voltage on the V3REF (3.3V) rail must always be at least 0.7V higher than the voltage on the VCCORE (1.5V) rail.
2. The voltage on the VCCCORE (1.5V) rail must always be higher than the voltage on the VTT rail.

For a ramp UP or DOWN:

- Duration of the power ramp cannot be less than 0.1ms.
- Duration of the power ramp cannot exceed 100ms.

Figure 7-27. Intel® 7300 Chipset Power Sequencing

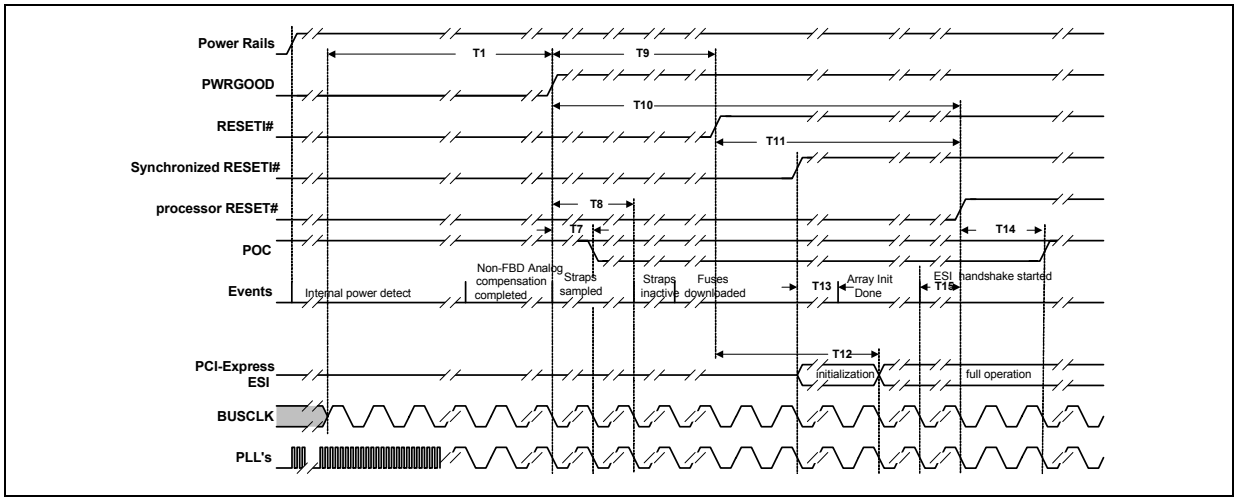


Note: Power-up -> 3.3V must ramp ahead and stay above 1.5V, which must ramp ahead and stay above 1.2V. 3.3V must always be at least 0.7V greater than 1.5V. Duration of the power ramp must be between 0.1 ms and 100 ms.

7.11.2 MCH Power On Sequence (Power Up/On Reset)

The power-up reset sequence is illustrated in Figure 7-28. The detailed timing requirements for the *Power Rails* in Figure 7-28 are shown in Figure 7-27.

Figure 7-28. Power-Up



7.11.3 MCH Reset Types

The MCH differentiates among five types of reset as defined in table Table 7-24.

Table 7-24. MCH Reset Classes

Type	Mechanism	Effect / Description
Power-Good	PWRGOOD Input Pin	Propagated throughout the system hierarchy. Resets all logic and state machines, and initializes all registers to their default states (sticky and non-sticky). Tri-states all MCH outputs, or drives them to "safe" levels.
Hard	RESET_N Input Pin, Configuration Write	Propagated throughout the system hierarchy. Resets all logic and state machines, and initializes all non-sticky registers to their default states. Tri-states all MCH outputs, or drives them to "safe" levels.
Processor-only	Configuration Write	Propagated to all processors via the FSBxRERSET_N pins on the FSB. The MCH does not undergo an internal reset.
Targeted	Configuration Write	Propagated down the targeted PCI Express port hierarchy. Treated as a "Hard" reset by all affected components, clearing all machine state and non-sticky configuration registers.
BINIT_N	Internal Error Handling Propagated via FSB **BINIT_N** pin	Propagated to all FSB attached components (the MCH and up to two processors). Clears the IOQ, and resets all FSB arbiters and state machines to their default states. Not recoverable.

7.11.3.1 Power-Good Mechanism

The initial boot of an Intel® 7300 Chipset based platform is facilitated by the Power-Good mechanism. The voltage sources from all platform power supplies are routed to a system component which tracks them as they ramp-up, asserting the platform "PwrGd" signal a fixed interval (nominally 2mS) after the last voltage reference has stabilized.

Both the MCH and the Intel® 631xESB/632xESB I/O Controller Hub receive the system PwrGd signal via dedicated pins as an asynchronous input, meaning that there is no assumed relationship between the assertion or deassertion of PwrGd and any system reference clock. When PwrGd is deasserted all platform subsystems are held in their reset state. This is accomplished by various mechanisms on each of the different interfaces. The MCH will hold itself in a power-on reset state when PwrGd is deasserted.

The Intel® 631xESB/632xESB I/O Controller Hub is expected to assert its PCIRST# output and maintain its assertion for 1mS after power is good. The PCIRST# output from Intel® 631xESB/632xESB I/O Controller Hub is expected to drive the RESET_N input pin on the MCH, which will in turn hold the processor complex in reset via assertion of the FSBxRESET# FSB signals.

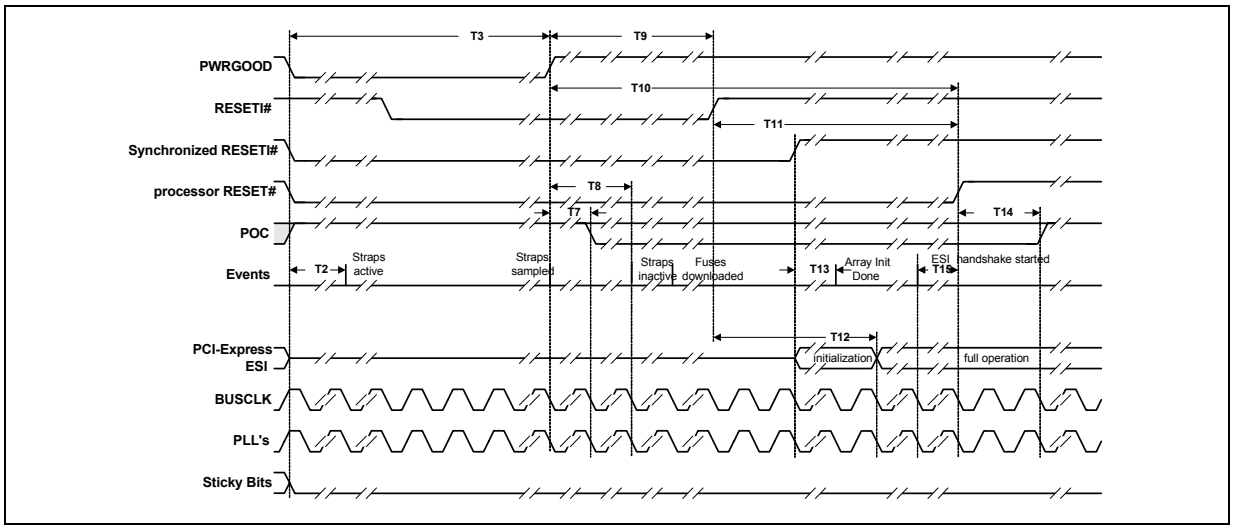
The PCI Express attached devices and any hierarchy of components underneath them are held in reset via implicit messaging across the PCI Express interface. The MCH is the root of the hierarchy, and will not engage in link training until power is good and the internal "hard" reset has deasserted.

A PwrGd reset will clear all internal state machines and logic, and initialize all registers to their default states, including "sticky" error status bits that are persistent through all other reset classes. To eliminate potential system reliability problems, all devices are also required to either tri-state their outputs or to drive them to "safe" levels during a power-on reset.

The only system information that will "survive" a PwrGd reset is either contained in battery-backed or non-volatile storage.

The PWRGOOD reset sequence is illustrated in Figure 7-29.

Figure 7-29. PWRGOOD Reset Sequence



7.11.3.2 Hard Reset Mechanism

Once the Intel® 7300 Chipset based platform has been booted and configured, a full system reset may still be required to recover from system error conditions related to various device or subsystem failures. The "hard" reset mechanism is provided to accomplish this recovery without clearing the "sticky" error status bits useful to track down the cause of system reboot.

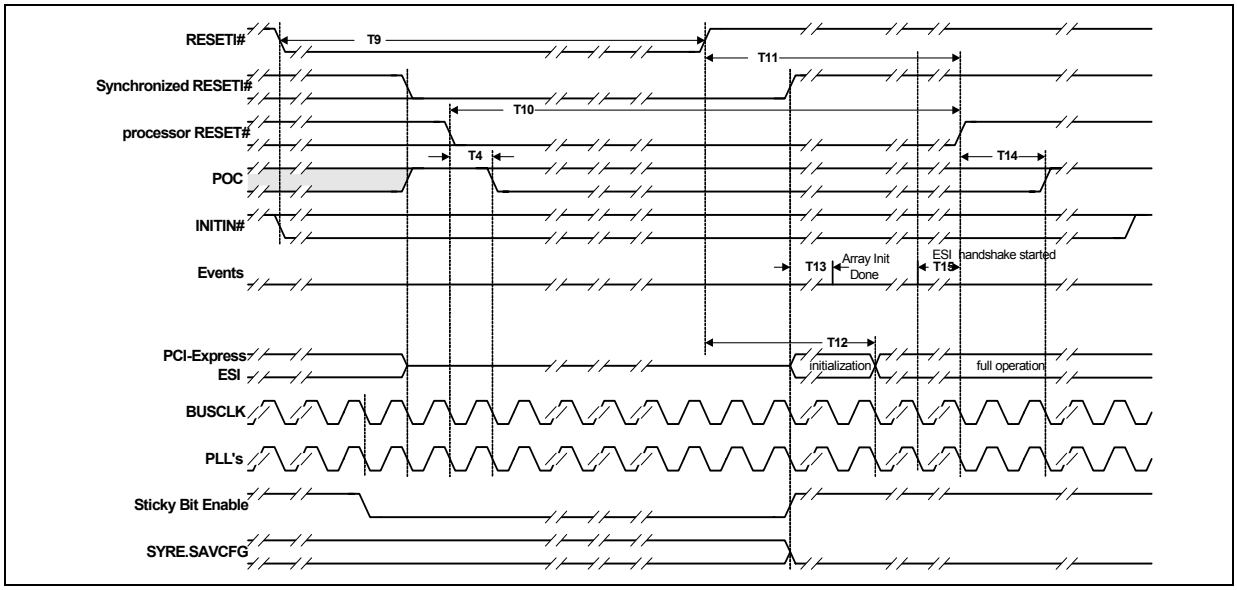
A hard reset is typically initiated by the Intel® 631xESB/632xESB I/O Controller Hub component via the PCIRST# output pin, which is commonly connected directly to the MCH RESET_N input pin. The Intel® 631xESB/632xESB I/O Controller Hub may be caused to assert PCIRST# via both software and hardware mechanisms. The MCH will recognize a hard reset any time RESET_N is asserted while PwrGd remains asserted.

The MCH will propagate a hard reset to the FSB and to all subordinate PCI Express subsystems. The FSB components are reset via the FSBxRESET# signals, while the PCI Express subsystems are reset implicitly when the root port links are taken down.

A hard reset will clear all internal state machines and logic, and initialize all "non-sticky" registers to their default states. Note that although the error registers will remain intact to facilitate root-cause of the hard reset, the platform in general will require a full configuration and initialization sequence to be brought back on-line.

The Hard Reset sequence is illustrated in Figure 7-30.

Figure 7-30. Hard Reset



7.11.3.3 Processor-Only Reset Mechanism

For power management and other reasons, the MCH supports a targeted processor only reset semantic. This mechanism was added to the platform architecture to eliminate double-reset to the system when reset-signaled processor information (such as clock gearing selection) must be updated during initialization bringing the system back to the S0 state after power had been removed from the processor complex.

7.11.3.4 Targeted Reset Mechanism

The targeted reset is provided for Hot-Plug events, as well as for port-specific error handling under Machine Check Architecture (MCA) or SMI software control. The former usage model is new with PCI Express technology, and the reader is referred to the *PCI Express Interface Specification, Rev 1.0a* for a description of the Hot-Plug mechanism.

A targeted reset may be requested by setting bit 6 (Secondary Bus Reset) of the Bridge Control Register (offset 3Eh) in the target root port device. This reset will be identical to a general hard reset from the perspective of the destination PCI Express device; it will not be differentiated at the next level down the hierarchy. Sticky error status will survive in the destination device, but software will be required to fully configure the port and all attached devices once reset and error interrogation have completed. After clearing bit 6, software may determine when the downstream targeted reset has

effectively completed by monitoring the state of bit 1 (Link Active) of the VS_STS1 register (offset 47h) in the target root port device. This bit will remain deasserted until the link has regained "link up" status, which implies that the downstream device has completed any internal and downstream resets, and successfully completed a full training sequence.

Under normal operating conditions it should not be necessary to initiate targeted resets to downstream devices, but the mechanism is provided to recover from combinations of fatal and uncorrectable errors which compromise continued link operation.

7.11.3.5 BINIT_N Mechanism

The BINIT_N mechanism is provided to facilitate processor handling of system errors which result in a hang on the FSB. The Machine Check Architecture (MCA) code responding to an error indication, typically IERR# or MCERR#, will cause an attempt to interrogate the MCH for error status, and if that FSB transaction fails to complete the processor will automatically time out and respond by issuing a BINIT_N sequence on the FSB.

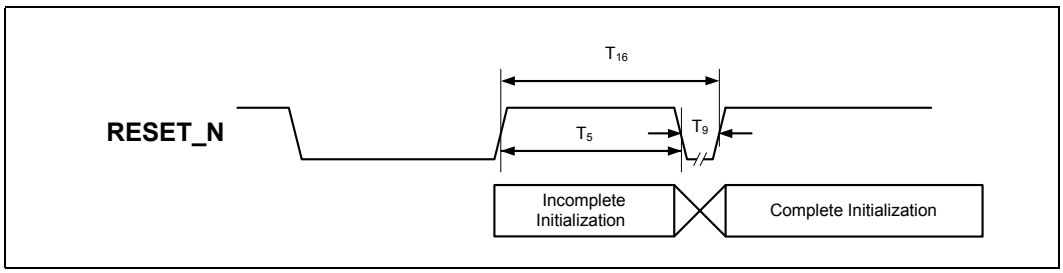
When BINIT_N is asserted on the FSB, all bus agents (CPUs and MCH) are required to reset their internal FSB arbiters and all FSB tracking state machines and logic to their default states. This will effectively "un-hang" the bus to provide a path into chipset configuration space. Note that the MCH device implements "sticky" error status bits, providing the platform software architect with free choice between BINIT_N and a general hard reset to recover from a hung system.

Although BINIT_N will not clear any configuration status from the system, it is not a recoverable event from which the platform may continue normal execution without first running a hard reset cycle. To guarantee that the FSB is cleared of any hang condition, the MCH will clear all pending transaction states within its internal buffers. This applies to outstanding FSB cycles as required, but also to in-flight memory transactions and inbound transactions. The resulting state of the platform will be highly variable depending upon what precisely got wiped-out due to the BINIT_N event, and it is not possible for hardware to guarantee that the resulting state of the machine will support continued operation. What the MCH will guarantee is that no subordinate device has been reset due to this event (PCI Express links will remain "up"), and that no internal configuration state (sticky or otherwise) has been lost. The MCH will also continue to maintain main memory via the refresh mechanism through a BINIT_N event, thus machine-check software will have access not only to machine state, but also to memory state in tracking-down the source of the error.

7.11.4 Reset Re-Triggering Timing

Figure 7-31 shows the timing for a RESET_N retrigger.

Figure 7-31. RESET_N Retriggering Limitations



7.11.5 Reset Timing Requirements

Table 7-25 specifies the timings drawn in Figure 7-28, Figure 7-29, Figure 7-30, and Figure 7-31. Nominal clock frequencies are described. Specifications still hold for derated clock frequencies.

Table 7-25. Power Up and Hard Reset Timings

Timing	Description	Min	Max	Comments
T1	Power and master clocks stable to PWRGOOD signal assertion	2 ms		PCI Express PLL specification
T2	PWRGOOD deassertion to straps active		40 ns	
T3	PWRGOOD deassertion	80 ns		Minimum PWRGOOD deassertion time while power and platform clocks are stable.
T4	Power On Configuration (POC) after RESET_N assertion delay	1 BUSCLK		
T5	RESET_N deassertion to RESET_N assertion	50 BUSCLK's		Minimum retrigger time on RESET_N deassertion.
T7	PWRGOOD assertion to POC active	2 BUSCLK's		POC turn-on delay after strap disable
T8	PWRGOOD assertion to straps inactive	12 ns	18 ns	Strap Hold Time
T9	RESET_N signal assertion during PWRGOOD signal assertion	1 ms		This delay can be provided by the Intel® 631xESB/632xESB I/O Controller Hub or by system logic
T10	RESET_N assertion during processor PWRGOOD assertion	1 ms	10 ms	Processor specification.
T11	RESET_N signal deassertion to processor RESET_N signal deassertion	480 us		
T12	RESET_N signal deassertion to completion of ESI initialization sequence		1,250,000 PECLK's	ESI clock is 100 MHz (PECLK)
T14	POC hold time after RESET_N deassertion	2 BUSCLK's	19 BUSCLK's	Processor specification
T15	Initiation of ESI reset sequence to processor RESET_N signal de-assertion		10,000 PECLK's + T17	Intel® 631xESB/632xESB I/O Controller Hub specification
T16	RESET_N retrigger delay	T5 + T9		

7.11.6 Miscellaneous Requirements and Limitations

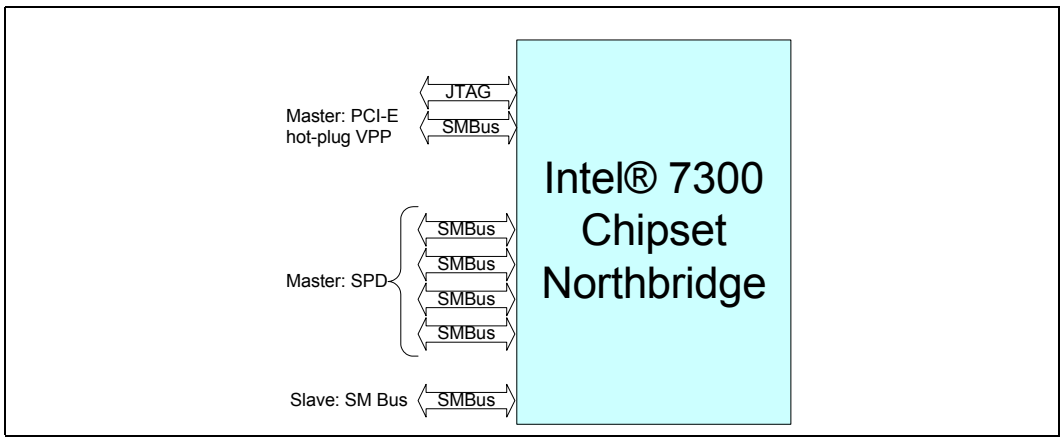
- Power rails and stable BUSCLK, FBD{0/1}CLK, and PECLK master clocks remain within specifications through all but power-up reset.
- Frequencies (for example, 266 MHz) described in this chapter are nominal. The MCH reset sequences must work for the frequency of operation range specified in the Clocking chapter.
- Hard Reset can be initiated by code running on a processor, JTAG, SMBus, or PCI agents.
- Hard Reset is not guaranteed to correct all illegal configurations or malfunctions. Software can configure sticky bits in the MCH to disable interfaces that will not be accessible after Hard Reset. Signaling errors or protocol violations prior to reset (from processor bus, FB-DIMM, or PCI Express) may hang interfaces that are not cleared by Hard Reset.
- System activity is initiated by a request from a processor bus. No I/O devices will initiate requests until configured by a processor to do so.
- The FB-DIMM channels will be enabled for packet levelization (FBDST.STATE="Ready" or "RecoveryReady" state) upon completion of a hard reset. Software should inspect the MCH FBDST.STATE configuration bits to determine which FB-DIMM channels are available.

- The default values of the POC configuration register bits do not require any processor request signals to be asserted when PWRGOOD is first asserted. Software sets these configuration registers to define these values, then initiates a hard reset that causes them to be driven during processor RESET_N signal assertion.
- Cleanly aborting an in-progress SPD command during a PWRGOOD deassertion is problematic. No guarantee can be issued as to the final state of the EEPROM in this situation. The MCH cannot meet the SPD data $t_{SU,STO}$ timing specification. Since the MCH floats the data output into a pull-up on the platform, a read will not degrade to a write. However, if the PWRGOOD deassertion occurs after the EEPROM has received the write bit, the data will be corrupted. The platform pull-up must be strong enough to complete a low-to-high transition on the clock signal within t_R = 1 microsecond (ATMEL AT24C01 timing specification) after deassertion of PWRGOOD to prevent clock glitches. Within these constraints, an in-progress write address will not be corrupted.

7.12 SMBus Interfaces Description

The MCH provides six fully functional System Management Bus (SMBus) Revision 2.0 compliant target interfaces. These interfaces are used to support platform level operations such as FB-DIMM memory Serial Presence Detect, PCI Hot-Plug, and configuration of platform devices. Each of these interfaces have dedicated uses as shown in Figure 7-32.

Figure 7-32. MCH SM Bus Interfaces



SM Buses 1, 2, 3 and 4 are dedicated to memory serial presence detect and FB-DIMM configuration. Each bus is dedicated to a single FB-DIMM channel. SM Bus 1 is assigned to FB-DIMM channel 0, SM Bus 2 is assigned to FB-DIMM channel 1, SM Bus 3 is assigned to FB-DIMM channel 2, and SM Bus 4 is assigned to FB-DIMM channel 3. SM Bus 6 is used to support PCI Express Hot-Plug.

The each SMBus interface consists of two interface pins; one a clock, and the other 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 MCH address is hard-coded to 01100000b (60h).

The SMBus protocol allows for traffic to stop in "mid sentence," requiring all targets to tolerate and properly "clean up" in the event of an access sequence that is abandoned by the initiator prior to normal completion. The MCH is compliant with this requirement.

The protocol comprehends "wait states" on read and write operations, which the MCH takes advantage of to keep the bus busy during internal configuration space accesses.

7.12.1 Internal Access Mechanism

All SMBus accesses to internal register space are initiated via a write to the CMD byte. Any register writes received by the MCH while a command is already in progress will receive a NAK to prevent spurious operation. The master is no longer expected to poll the CMD byte to prevent the obliteration a command in progress prior to issuing further writes. The SMBus access will be delayed by stretching the clock until such time that the data is delivered. Note that per the *System Management Bus (SMBus) Specification, Rev 2.0*, this can not be longer than 25 ms. To set up an internal access, the four ADDR bytes are programmed followed by a command indicator to execute a read or write. Depending on the type of access, these four bytes indicate either the Bus number, Device, Function, Extended Register Offset, and Register Offset, or the memory-mapped region selected and the address within the region. The configuration type access utilizes the traditional bus number, device, function, and register offset; but in addition, also uses an extended register offset which expands the addressable register space from 256 bytes to 4 Kilobytes. The memory-mapped type access redefines these bytes to be a memory-mapped region selection byte, a filler byte which is all zeroes, and then the memory address within the region. Refer to the earlier tables, which display this information. Note that the filler byte is not utilized, but enforces that both types of accesses have the same number of address bytes, and does allow for future expansion.

It is perfectly legal for an SMBus access to be requested while an FSB-initiated access is already in progress. The MCH supports "wait your turn" arbitration to resolve all collisions and overlaps, such that the access that reaches the configuration ring arbiter first will be serviced first while the conflicting access is held off. An absolute tie at the arbiter will be resolved in favor of the FSB. Note that SMBus accesses must be allowed to proceed even if the internal MCH transaction handling hardware and one or more of the other external MCH interfaces are hung or otherwise unresponsive.

7.12.2 SMBus Transaction Field Definitions

The SMBus target port has it's own set of fields which the MCH sets when receiving an SMBus transaction. They are not directly accessible by any means for any device.

Table 7-26. SMBus Transaction Field Summary (Sheet 1 of 2)

Position	Mnemonic	Field Name
1	CMD	Command
2	BYTCNT	Byte Count
3	ADDR3	Bus Number (Register Mode) or Destination Memory (Memory Mapped Mode)
4	ADDR2	Device / Function Number (Register Mode) or Address Offset [23:16] (Memory Mapped Mode)
5	ADDR1	Extended Register Number (Register Mode) or Address Offset [15:8] (Memory Mapped Mode)
6	ADDR0	Register Number (Register Mode) or Address Offset [7:0] (Memory Mapped Mode)
7	DATA3	Fourth Data Byte [31:24]
8	DATA2	Third Data Byte [23:16]
9	DATA1	Second Data Byte [15:8]

Table 7-26. SMBus Transaction Field Summary (Sheet 2 of 2)

Position	Mnemonic	Field Name
10	DATA0	First Data Byte [7:0]
11	STS	Status, only for reads

Table 7-26 indicates the sequence of data as it is presented on the SMBus following the byte address of the MCH itself. Note that the fields can take on different meanings depending on whether it is a configuration or memory-mapped access type. The command indicates how to interpret the bytes.

7.12.2.1 Command Field

The command field indicates the type and size of transfer. All configuration accesses from the SMBus port are initiated by this field. While a command is in progress, all future writes or reads will be negative acknowledged (NAK) by the MCH to avoid having registers overwritten while in use. The two command size fields allows more flexibility on how the data payload is transferred, both internally and externally. The begin and end bits support the breaking of the transaction up into smaller transfers, by defining the start and finish of an overall transfer.

Position	Description
7	**Begin Transaction Indicator.** 0 = Current transaction is NOT the first of a read or write sequence. 1 = Current transaction is the first of a read or write sequence. On a single transaction sequence this bit is set along with the End Transaction Indicator.
6	End Transaction Indicator. 0 = Current transaction is NOT the last of a read or write sequence. 1 = Current transaction is the last of a read or write sequence. On a single transaction sequence this bit is set along with the Begin Transaction Indicator.
5	Address Mode. Indicates whether memory or configuration space is being accessed in this SMBus sequence. 0 = Memory Mapped Mode 1 = Configuration Register Mode
4	Packet Error Code (PEC) Enable. When set, each transaction in the sequence ends with an extra CRC byte. The MCH would check for CRC on writes and generate CRC on reads. PEC is not supported by the MCH. 0 = Disable 1 = Not Supported
3:2	Internal Command Size. All accesses are naturally aligned to the access width. This field specifies the internal command to be issued by the SMBus slave logic to the MCH core. 00 = Read Dword 01 = Write Byte 10 = Write Word 11 = Write Dword
1:0	SMBus Command Size. This field specifies the SMBus command to be issued on the SMBus. This field is used as an indication of the length of the transfer so that the slave knows when to expect the PEC packet (if enabled). 00 = Byte 01 = Word 10 = DWord 11 = Reserved

7.12.2.2 Byte Count Field

The byte count field indicates the number of bytes following the byte count field when performing a write or when setting up for a read. The byte count is also used, when returning data, to indicate the number of bytes (including the status byte) which are returned prior to the data. Note that the byte count is only transmitted for block type accesses on SMBus. SMBus word or byte accesses do not use the byte count.

Position	Description
7:0	**Byte Count.** Number of bytes following the byte count for a transaction.

7.12.2.3 Address Byte 3 Field

This field should be programmed with the bus number of the desired configuration register in the lower 5 bits for a configuration access. For a memory-mapped access, this field selects which memory-map region is being accessed. There is no status bit to poll to see if a transfer is in progress, because by definition if the transfer completed when the task is done. Clock stretch is used to guarantee the transfer is truly complete.

The MCH does not support access to other logical bus numbers via the SMBus port. All registers “attached” to the SMBus have access to all other registers that are on logical bus#0. The MCH makes use of this knowledge to implement a modified usage of the Bus Number register providing access to internal registers outside of the PCI compatible configuration window.

Position	Configuration Register Mode Description	Memory Mapped Mode Description
7:5	Ignored.	Memory map region to access. 01h = DMA 08h = DDR 09h = CHAP Others = Reserved
4:0	Bus Number. Must be zero: the SMBus port can only access devices on the MCH and all devices are bus zero.	

7.12.2.4 Address Byte 2 Field

This field indicates the Device Number and Function Number of the desired configuration register if for a configuration type access, otherwise it should be set to zero.

Position	Configuration Register Mode Description	Memory Mapped Mode Description
7:3	**Device Number.** Can only be devices on the MCH.	Zeros used for padding.
2:0	Function Number.	

7.12.2.5 Address Byte 1 Field

This field indicates the upper address bits for the 4K region specified by the register offset. Only the lower bit positions of this field are used, the upper four bits are ignored.

Position	Description
7:4	Ignored.
3:0	Extended Register Number. Upper address bits for the 4K region of register offset.

7.12.2.6 Address Byte 0 Field

This field indicates the lower eight address bits for the register with the 4K region, regardless whether it is a configuration or memory-map type of access.

Position	Description
7:0	**Register Offset.**

7.12.2.7 Data Field

This field is used to receive read data or to provide write data associated with the addressed register.

At the completion of a read command, this field will contain the data retrieved from the addressed register. All reads will return an entire aligned DWord (32 bits) of data.

For write operations, the number of byte(s) of this 32 bit field is loaded with the desired write data. For a byte write only bits 7:0 will be used, for a Word write only bits 15:0 will be used, and for a DWord write all 32 bits will be used.

Position	Description
31:24	**Byte 3 (DATA3).** Data bits [31:24] for DWord.
23:16	**Byte 2 (DATA2).** Data bits [23:16] for DWord.
15:8	**Byte 1 (DATA1).** Data bits [15:8] for DWord and Word.
7:0	**Byte 0 (DATA0).** Data bits [7:0] for DWord, Word and Byte.

7.12.2.8 Status Field

For a read cycle, the returned data is preceded by one byte of status. The following table shows how the status byte bits are defined.

Position	Description
7	**Internal Time-out.** **0 = SMBus request is completed within 2 ms internally** 1 = SMBus request is not completed in 2 ms internally.
6	Ignored.
5	**Internal Master Abort.** 0 = No Internal Master Abort Detected. 1 = Detected an Internal Master Abort.
4	**Internal Target Abort.** 0 = No Internal Target Abort Detected. 1 = Detected an Internal Target Abort.
3:1	Ignored.
0	Successful. 0 = The last SMBus transaction was not completed successfully. 1 = The last SMBus transaction was completed successfully.

7.12.2.9 Unsupported Access Addresses

It is possible for an SMBus master to program an unsupported bit combination into the ADDR registers. The MCH does not support such usage, and may not gracefully terminate such accesses.

7.12.3 SMB Transaction Pictographs

The MCH SMBus target interface is targeted to enterprise domains. The enterprise domain is an extension of the original SMBus desktop domain. The following drawings are included to describe the SMBus enterprise transactions.

Figure 7-33. DWORD Configuration Read Protocol (SMBus Block Write / Block Read, PEC Disabled)

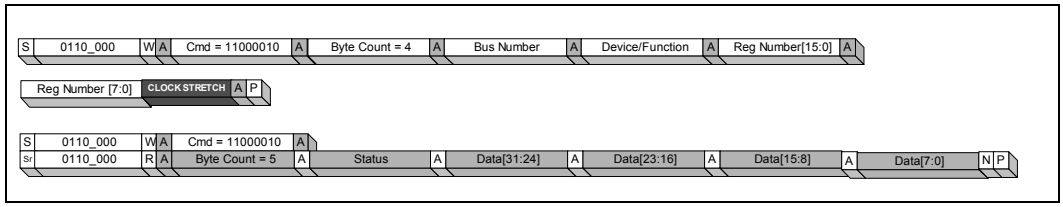


Figure 7-34. DWORD Configuration Write Protocol (SMBus Block Write, PEC Disabled)

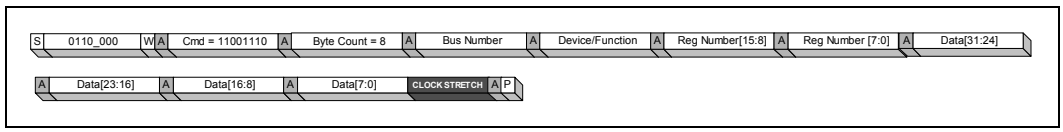


Figure 7-35. DWORD Memory Read Protocol (SMBus Block Write / Block Read, PEC Disabled)

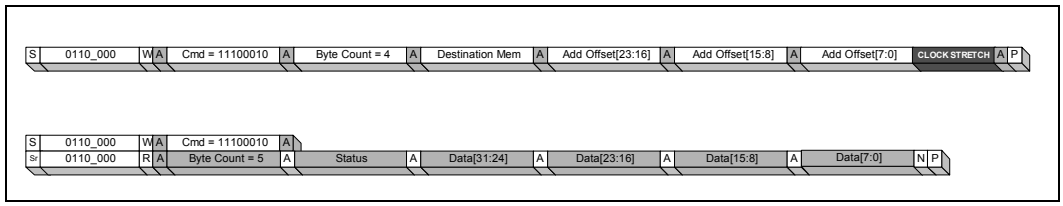


Figure 7-36. DWORD Memory Write Protocol

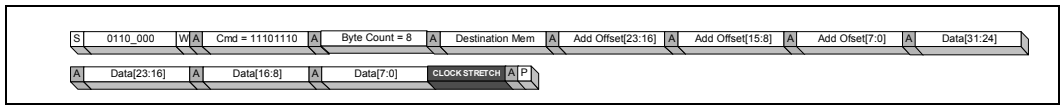


Figure 7-37. DWORD Configuration Read Protocol (SMBus Word Write / Word Read, PEC Disabled)

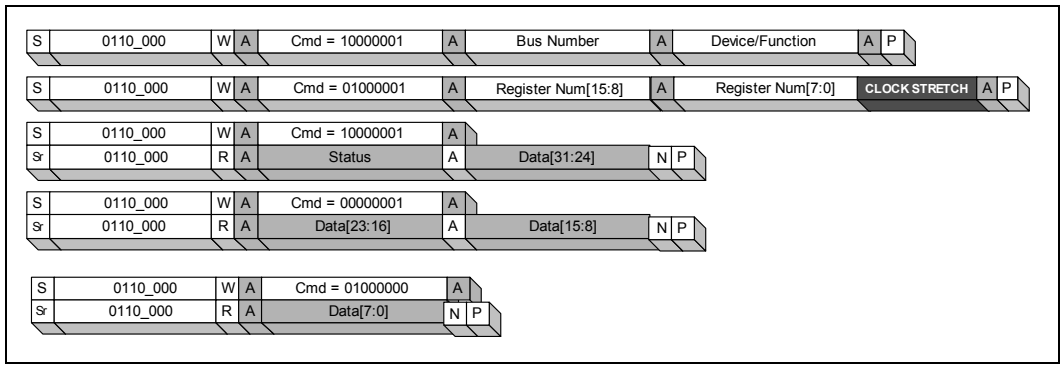


Figure 7-38. DWORD Configuration Write Protocol (SMBus Word Write, PEC Disabled)

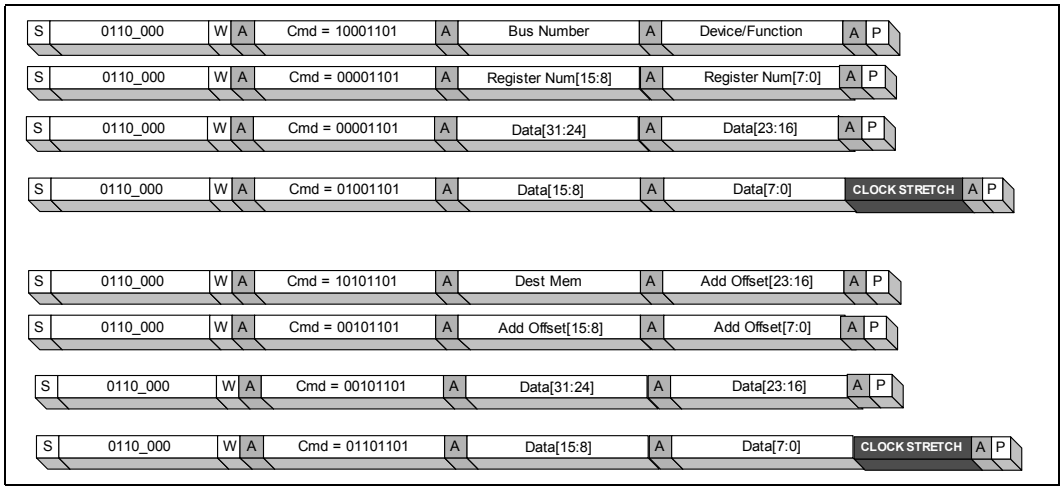


Figure 7-39. DWORD Memory Read Protocol (SMBus Word Write / Word Read, PEC Disabled)

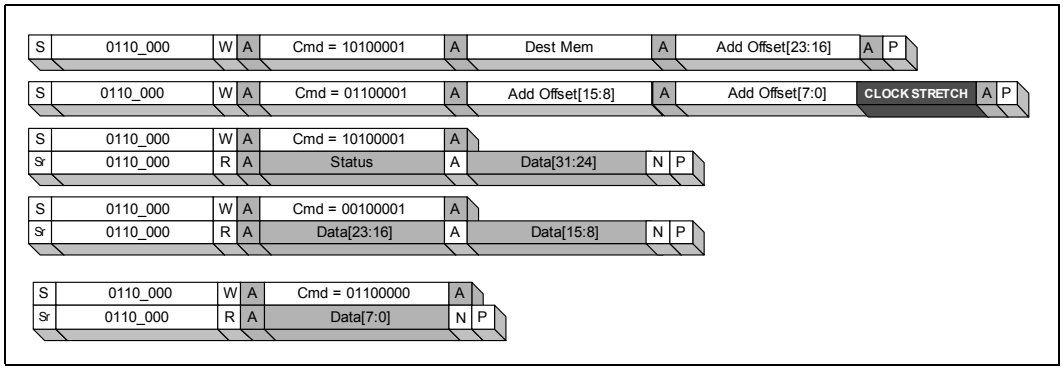
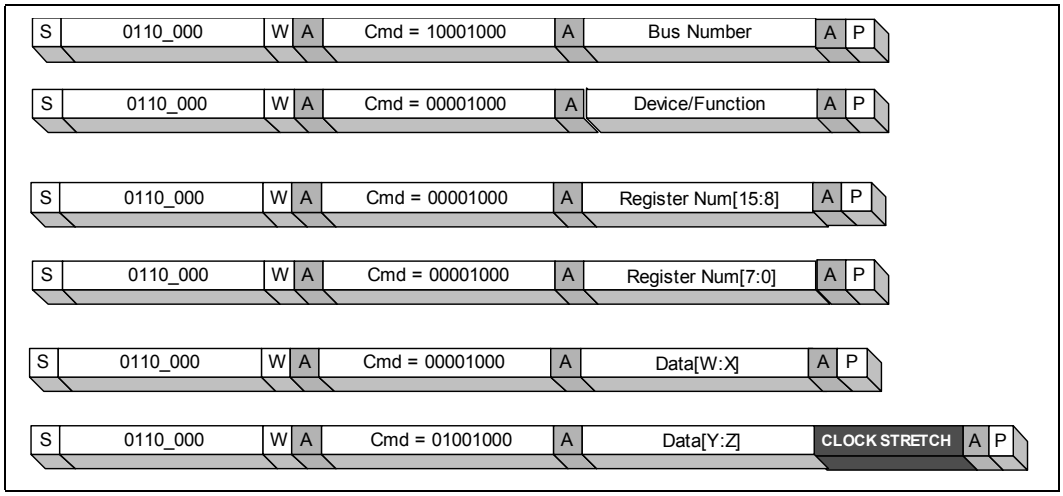


Figure 7-40. WORD Configuration Wrote Protocol (SMBus Byte Write, PEC Disabled)



7.12.4 Slave SM Bus, SM Bus 0

System Management software in a Intel® 7300 Chipset based platform can initiate system management accesses to the configuration registers via the Slave SM bus, SM Bus 0.

The mechanism for the Server Management (SM) software to access configuration registers is through a SMBus Specification, Revision 2.0 compliant slave port. Some Intel® 7300 Chipset components contain this slave port and allow accesses to their configuration registers. The product specific details are compatible with the Intel® 631xESB/632xESB I/O Controller Hub SMBus configuration access mechanism. Most of the MCH registers can be accessed through the SMBus configuration mechanism.

SMBus operations are made up of two major steps:

1. Writing information to registers within each component
2. Reading configuration registers from each component.

The following sections will describe the protocol for an SMBus master to access a Intel® 7300 Chipset based platform component's internal configuration registers. Refer to the SMBus Specification, Revision 2.0 for the bus protocol, timings, and waveforms.

Each component on the Intel® 7300 Chipset based platform must have a unique address. Intel® 7300 Chipset based platform component addresses are defined in the following table.

Table 7-27. SMBus Address for Intel® 7300 Chipset Memory Controller Hub (MCH) Platform

Component	SMBus Address (7:1)
Intel® 7300 Chipset MCH	1100_000

7.12.4.1 Supported SMBus Commands

Intel® 7300 Chipset Memory Controller Hub (MCH) components SMBus Rev. 2.0 slave ports support the following six SMBus commands:

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

Sequencing these commands will initiate internal accesses to the component's configuration registers.

Each configuration read or write first consists of an SMBus write sequence which initializes the Bus Number, Device Number, etc. 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 perform a configuration read) or a write sequence (which performs a configuration write).

Each SMBus transaction has an 8-bit command driven by the master. The format for this command is illustrated in Table 7-28 below.

Table 7-28. SMBus Command Encoding

7	6	5	4	3:2	1:0
Begin	End	Rsvd	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 - *Rsvd*

The *Begin* bit indicates the first transaction of a read or write sequence.

The *End* bit indicates the last transaction of a read or write sequence.

The *Pecan* bit enables the 8-bit Packet Error Code (PEC) generation and checking logic.

The *Internal Command* field specifies the internal command to be issued by the SMBus slave logic. Note that the Internal Command must remain consistent during a sequence that accesses a configuration register. Operation cannot be guaranteed if it is not consistent when the command setup sequence is done.

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 the slave knows when to expect the Packet Error Code packet.

Reserved bits should be written to zero to preserve future compatibility.

7.12.4.2 Configuration Register Read Protocol

Configuration 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. If the data is not available before the slave interface acknowledges this last write command (ACK), the slave will "clock stretch" until the data returns to the SMBus interface unit. If an error occurs during the internal access, the last write command will receive a NAK. A status field indicates abnormal termination and contains status information such as target abort, master abort, and time-outs. The status field encoding is defined in the following table.

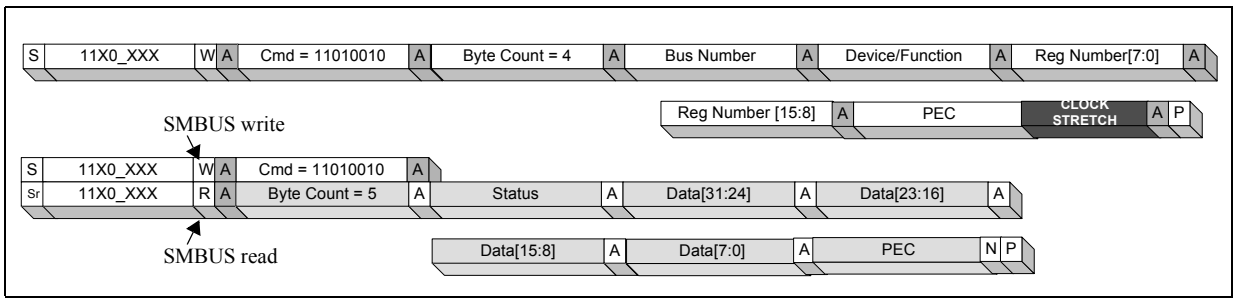
Table 7-29. Status Field Encoding for SMBus Reads

Bit	Description
7	Internal Time-out. This bit is set if an SMBus request is not completed in 2ms internally.
6	Reserved
5	Internal Master Abort
4	Internal Target Abort
3:1	Reserved
0	Successful

Examples of configuration reads are illustrated below. All of these examples have Packet Error Code (PEC) enabled. If the master does not support PEC, then bit 4 of the command would be cleared and there would not be a PEC phase. For the definition of

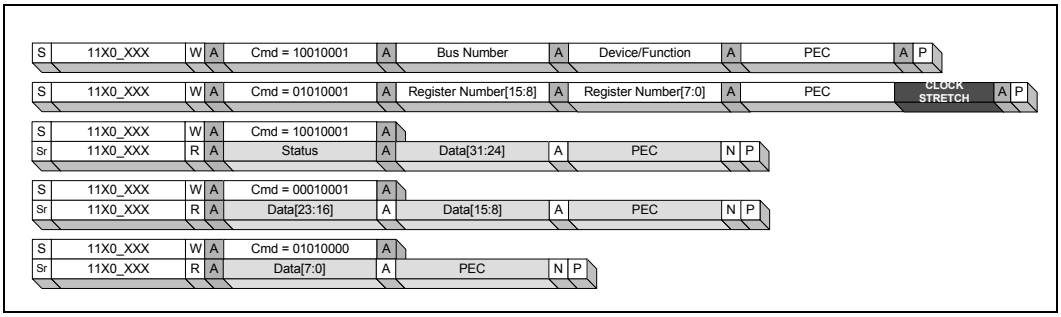
the diagram conventions below, refer to the *SMBus Specification*, Revision 2.0. For SMBus read transactions, the last byte of data (or the PEC byte if enabled) is NAKed by the master to indicate the end of the transaction. For diagram compactness, "Register Number[]" is also sometimes referred to as "Reg Number" or "Reg Num".

Figure 7-41. SMBus Configuration Read (Block Write / Block Read, PEC Enabled)



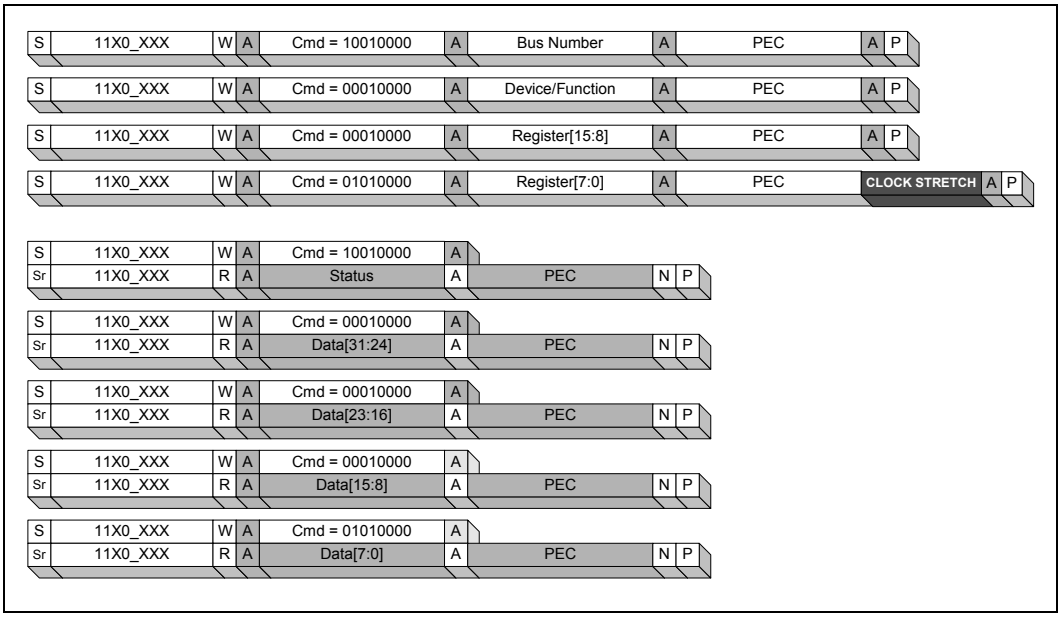
This is an example using word reads. The final data is a byte read.

Figure 7-42. SMBus Configuration Read (Word Writes / Word Reads, PEC Enabled)



The following example uses byte reads.

Figure 7-43. SMBus Configuration Read (Write Bytes / Read Bytes, PEC Enabled)



7.12.4.3 Configuration Register Write Protocol

Configuration 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).

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

Figure 7-44. SMBus Configuration Write (Block Write, PEC Enabled)

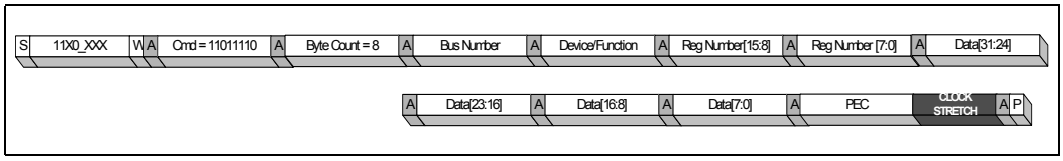


Figure 7-45. SMBus Configuration Write (Word Writes, PEC Enabled)

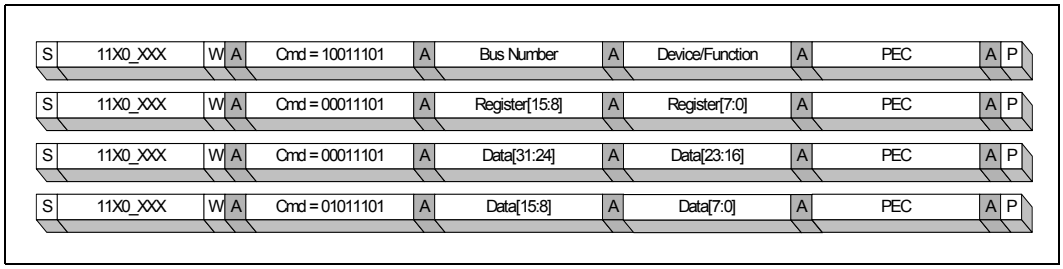
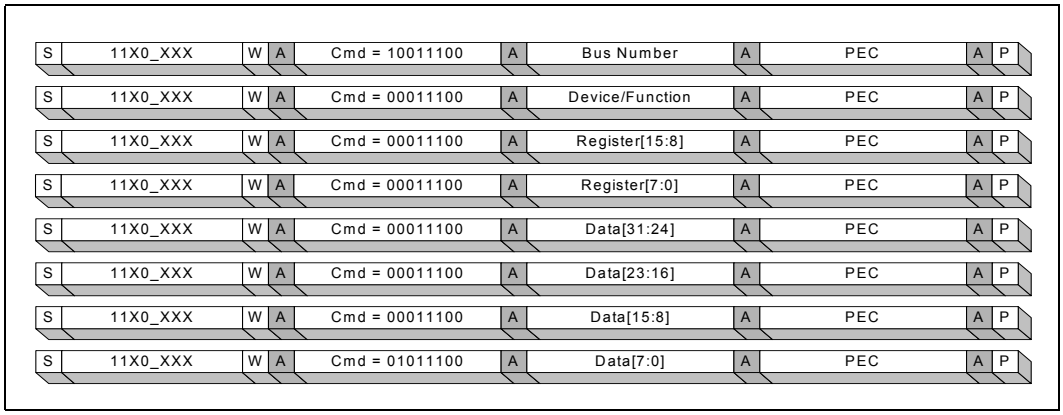


Figure 7-46. SMBus Configuration Write (Write Bytes, PEC Enabled)



7.12.4.4 SMBus Error Handling

The SMBus slave interface handles two types of errors: Internal and PEC. For example, internal errors can occur when the MCH issues a configuration read on the PCI Express port that read's terminates in error. These errors manifest as a not-acknowledge (NAK) for the read command (*End* bit is set). If an internal error occurs during a configuration write, the final write command receives a NAK just before the stop bit. If the master receives a NAK, 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 NAK the PEC packet.

7.12.4.5 SMBus Interface Reset

- The slave interface state machine can be reset by the master in two ways:
- The master holds SCL low for 25 ms cumulative. Cumulative in this case means that all the "low time" for SCL is counted between the Start and Stop bit. If this totals 25ms before reaching the Stop bit, the interface is reset.
- The master holds SCL continuously high for 50ms.

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

7.12.5 FB-DIMM SPD Interface, SM Buses 1, 2, 3 and 4

The MCH integrates a 100 KHz SPD controller to access the FB-DIMM configuration information. SMBus 1 is dedicated to FB-DIMM branch 0, channel 0 DIMMs. SMBus 2 is dedicated to FB-DIMM branch 0, channel 1 DIMMs. SMBus 3 is dedicated to FB-DIMM branch 1, channel 0 DIMMs and SMBus 4 is dedicated to FB-DIMM branch 1, channel 1 DIMMs. There can be a maximum of four SPD EEPROM's associated with each SPD bus. The FB-DIMM SPD interfaces are wired as depicted in Figure 7-3.

Board layout must map chip selects to SPD Slave Addresses as shown in Table 7-5. The slave address is written to the SPDCMD configuration register.

7.12.5.1 SPD Asynchronous Handshake

The SPD bus is an asynchronous serial interface. Once software issues an SPD command (SPDCMD.CMD = SPDW or SPDR), software is responsible for verifying command completion before another SPD command can be issued. Software can determine the status of an SPD command by observing the SPD configuration register.

An SPD command has completed when any one command completion field (RDO, WOD, SBE) of the SPD configuration register is observed set to 1. An SPDR command has successfully completed when the RDO field is observed set to 1. An SPDW command has successfully completed when the WOD field is observed set to 1. An unsuccessful command termination is observed when the SBE field is set to 1. The MCH will clear the SPD configuration register command completion fields automatically whenever an SPDR or SPDW command is initiated. Polling may begin immediately after initiating an SPD command.

Software can determine when an SPD command is being performed by observing the BUSY field of the SPD configuration register. When this configuration bit is observed set to 1, the interface is executing a command.

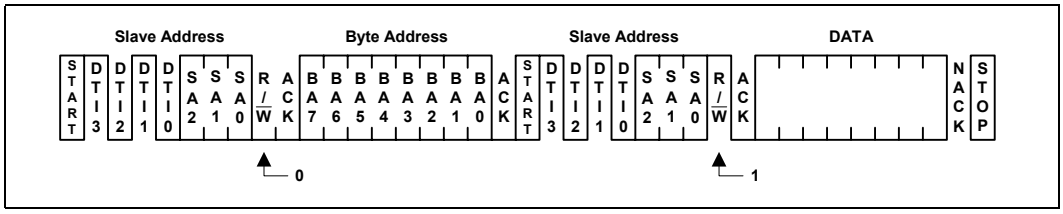
Valid SPD data is stored in the DATA field of the SPD configuration register upon successful completion of the SPDR command (indicated by 1 in the RDO field). Data to be written by an SPDW command is placed in the DATA field of the SPDCMD configuration register.

Unsuccessful command termination will occur when an EEPROM does not acknowledge a packet at any of the required ACK points, resulting in the SBE field being set to 1.

7.12.5.2 Request Packet for SPD Random Read

Upon receiving the SPDR command, the MCH generates the Random Read Register command sequence as shown in Figure 7-47. The returned data is then stored in the MCH SPD configuration register in bits [7:0], and the RDO field is set to 1 by the MCH to indicate that the data is present and that the command has completed without error.

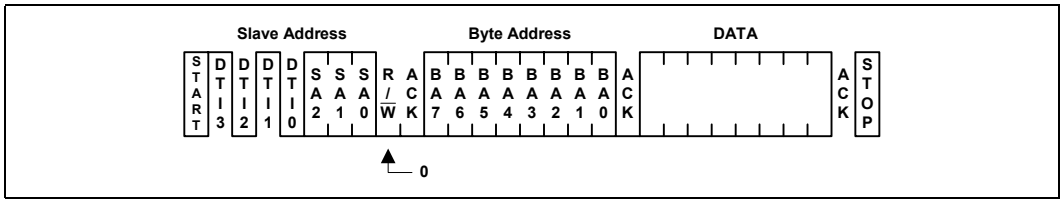
Figure 7-47. Random Byte Read Timing



7.12.5.3 Request Packet for SPD Byte Write

Upon receiving the SPDW command, the MCH generates the Byte Write Register command sequence as shown in Figure 7-48. The MCH indicates that the SIO command has completed by setting the WOD bit of the SPD configuration register to 1.

Figure 7-48. Byte Write Register Timing



7.12.5.4 SPD Protocols

The MCH supports the SPD protocols shown in Table 7-30.

Table 7-30. MCH Supported SPD Protocols

MCH Supported SPD Protocols
Random Byte Read
Byte Write

7.12.5.5 SPD Bus Time-out

If there is an error in the transaction, such that the SPD EEPROM does not signal an acknowledge, the transaction will time out. The MCH will discard the cycle and set the **SBE** bit of the **SPD** configuration register to 1 to indicate this error. The time-out counter within the MCH begins counting after the last bit of data is transferred to the DIMM, while the MCH waits for a response.

7.12.6 PCI Express Hot-Plug Support, VPP SMBus

The VPP SMBus is the PCI Express Hot-Plug port, a Hot-Plug Virtual Pin Port (VPP) that operates using the SM Bus Masters protocol as defined in *System Management Bus Specification 2.0*.

The VPP SMBus is dedicated to support PCI Express Hot-Plug devices. Support for PCI Express is an option described in *PCI Express Base Specification*, Revision 1.0a. The PCI Express Hot-Plug model implies a hot-plug controller per port which is identified to software as a capability of the P2P Bridge configuration space.

PCI Express hot-plug support requires that the MCH supports a set of hot-plug messages (listed in Figure 7-16 and Figure 7-20) to manage the states between the hot-plug controller and the device.

The PCI Express form factor has an impact to the level of support required of the MCH. For example, some of the hot-plug messages are required only if the LED indicators reside on the actual card and are accessed through the endpoint device. The MCH supports all of the hot-plug messages so that the platform is not constrained to any particular form factor.

A standard hot-plug usage model is beneficial to customers who buy systems with hot-plug slots because many customers utilize hardware and software from different vendors. A standard usage model allows customers to use the PCI hot-plug slots on all of their systems without having to retrain operators.

In order to define a programming model for the PCI Standard Hot-Plug Controller (SHPC), it is necessary to make some assumptions about the interface between a human operator and a hot-plug slot. The SHPC programming model includes two indicators, one optional push button, and a sensor on the manually-operated retention latch for each supported slot.

7.12.6.1 Hot-Plug Indicators

The Standard Usage Model assumes that the platform provides two indicators per slot (the Power Indicator and the Attention Indicator). Each indicator is in one of three states: on, off, or blinking. Hot-plug system software has exclusive control of the indicator states by issuing commands to the SHPC.

The SHPC controls blink frequency, duty cycle, and phase. Blinking indicators operate at a frequency of 1.5 Hz and 50% (+/- 5%) duty cycle. Both indicators are completely under the control of system software.

7.12.6.2 Attention Button

An Attention Button is a momentary-contact push-button, located adjacent to each hotplug slot, that is pressed by the user to initiate a hot-insertion or a hot-removal at that slot. The Power Indicator provides visual feedback to the human operator (if the system software accepts the request initiated by the Attention Button) by blinking. Once the Power Indicator begins blinking, a 5-second abort interval exists during which a second depression of the Attention Button cancels the operation. Software has the responsibility to implement this 5-second abort interval.

7.12.6.3 Hot-Plug Controller

PCI Express Hot-Plug requires that the MCH implement a Hot-Plug controller for every Hot-Pluggable interface. The Hot-Plug controller is a capability of the bridge configuration space and the register set is accessible through the standard PCI capability mechanism defined in the *PCI Express Base Specification*, Revision 1.0a.

7.12.6.4 PCI Express Hot-Plug Usage Model

Not all concepts from the PCI standard hot-plug definition apply directly to PCI Express interfaces. The PCI Express specification still calls for an identical software interface in order to facilitate adoption with minimal development overhead on this aspect of the implementation. The largest variance from the old PCI hot-plug model is in control of the interface itself. PCI required arbitration support for idling already connected components, and “quick switches” to isolate the bus interface pins of a hot-plug slot. PCI Express is a point-to-point interface, making hot-plug a degenerate case of the old model that doesn’t require such arbiter support. Furthermore, the PCI Express interface is inherently tolerant of hot connect or disconnect, and does not have explicit clock or reset pins defined as a part of the bus (although they are standard pieces of some

defined PCI Express connector form factors). As a result of these differences, some of the inherited hot-plug command and status codes are misleading when applied to PCI Express.

The compatible set of hot-plug registers may be accessed via memory-mapped transactions, or via the MCH configuration mechanism as defined in the configuration mechanism chapter of this document. For specific information on the hot-plug register set, refer to the chapter on configuration register details.

The messages used for the hot-plug model are listed in Table 7-16, "Incoming PCI Express Requests" on page 401 and Table 7-20, "PCI Express Transaction ID Handling" on page 403 describe the behavior of the button and LEDs.

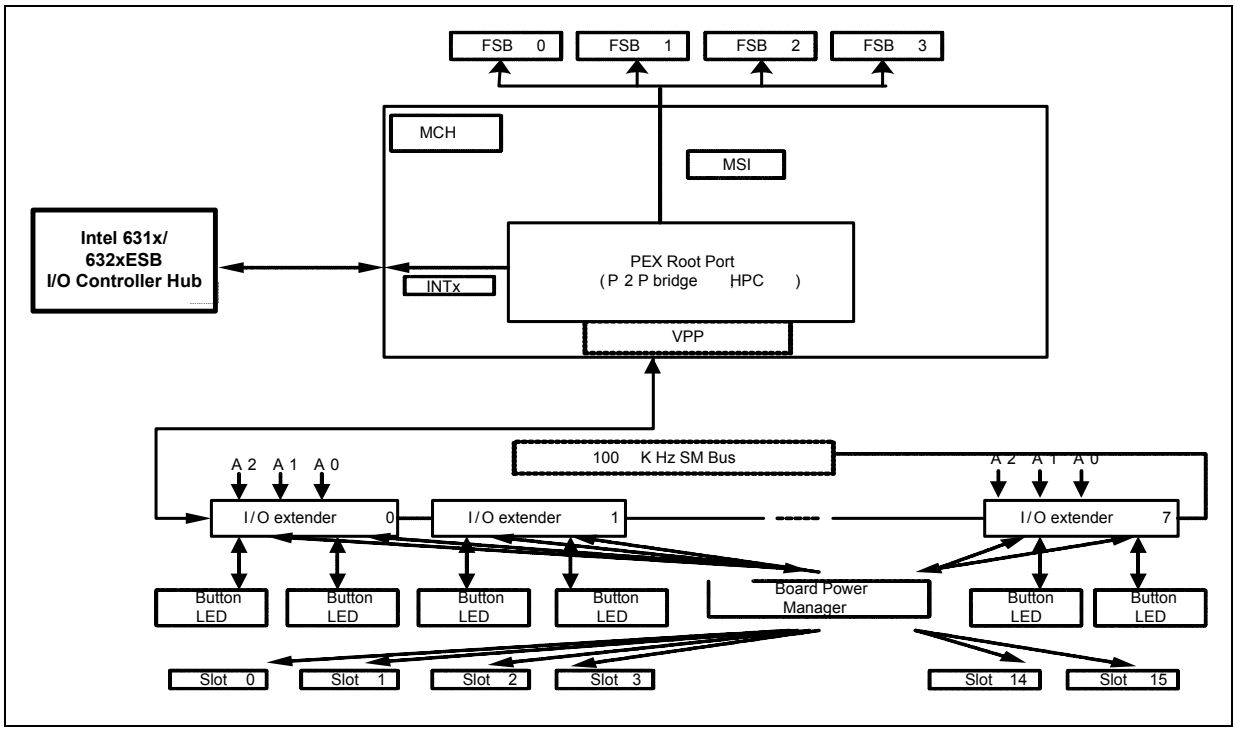
7.12.6.5 Virtual Pin Ports

Shown in the Figure 7-49 is a high level block diagram of virtual pin ports and theoretical maximum number of PCI Express card slots that could be supported for hot-plug operations. In this VPP usage model, 16 slots (max) are shown in Figure 7-49 but for the Intel® 7300 Chipset based platform only 6 PCI Express slots[1] will be used for the I/O hot-plug operations.

Note: Port 0, the ESI slot, is not hot-pluggable.

Since MCH has only six PCI Express ports, only six hot-plug slots should be present in a MCH platform. MCH PCI Express virtual pin port will only process six hot-plug slots accordingly.

Figure 7-49. PCI Express Hot-Plug/VPP Block Diagram



1. This does not include the ESI (port 0) which is not hot-pluggable.

The MCH masters a 100 KHz hot-plug SMBus interface thru pins GPIOSMBCLK,and GPIOSMBDATA, for PCI Express ports that connect to a variable number of serial to parallel I/O ports such as the Phillips PCA9555[1] I/O Extender. The MCH only supports SMBus devices with registers mapped as per Table 7-31. These I/O Extender components have 16 I/Os, divided into two 8-bit ports that can be configured as inputs or outputs. The MCH has a crossbar which associates each PCI Express Hot-Plug Unit (HPU) slots with one of these 8-bit ports. The mapping is defined by a Virtual Pin Port register field, PEXCTRL.VPP, for each of the PCI Express HPU slots. The VPP register holds the SMBus address and port number of the I/O Port associated with the PCI Express HPU. A[2:0] pins on each I/O Extender (i.e. PCA9555 or compatible components) connected to the MCH must strapped uniquely. Table 7-32 defines how the eight hot-plug signals are mapped to pins on the VPP.

Table 7-31. I/O Port Registers in I/O Extender supported by the MCH

Register	Name	MCH 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	Not written by the MCH
5	Polarity Inversion Port 1	
6	Configuration Port 0	Direction set as per Table 7-32
7	Configuration Port 1	

7.12.6.6 Operation

When the MCH comes out of reset, the I/O ports are inactive. After a reset, the MCH is not aware of how many I/O Ports are connected to it, what their addresses are, nor what PCI Express ports are hot-pluggable. The MCH does not master any commands on the SMBus until a hot-plug Capable bit is set.

For a PCI Express slot, an additional DIS_VPP bit is used to differentiate card or module hot-plug support, DIS_VPP bit needs to be set to 0 to enable hot-plug support for PCI Express card slot.

When BIOS sets a Hot-plug Capable bit (PEXSLOTCAP.HPC and PEXCTRL.DIS_VPP for PCI Express; HPCTL.HPC for FB-DIMM HPU), the MCH initializes the associated VPP with Direction and Voltage Logic Level configuration as per Table 7-32. VPP registers for PCI Express which do not have the hot-plug capable bit set are invalid. Additionally, if the DIS_VPP bit is set to 1, then the corresponding VPP register is invalid for the PCI Express slot. This is intended for PCI Express module hot-plug which no VPP support is required. The I/O Extender's Polarity is left at its default value and never written, but the direction and voltage logic levels are written using the addresses defined in Table 7-32.

When the MCH is not doing a direction write, it performs input register reads and output register writes to all valid VPPs. This sequence repeats indefinitely until a new hot-plug capability bit is set. To minimize the completion time of this sequence and minimize complexity, both ports are always read or written. For the maximum number of 6 I/O Ports, and assuming no clock stretching, this sequence can take up to 51ms. If new hot-plug capability bits are not being set, this is the maximum timing uncertainty in sampling or driving these signals.

1. The MCH VPP supports PCA9555 or compatible I/O Extender only.

Table 7-32 describes the Hot-Plug Signals used for hot-plug.

Table 7-32. Hot-Plug Signals on a Virtual Pin Port

Bit	Direction	Voltage Logic Level	Signal	Logic True Meaning	Logic False Meaning
0	Output	High_true	ATNLED	ATTN LED is to be turned ON	ATTN LED is to be turned OFF
1	Output	High_true	PWRLED	PWR LED is to be turned ON	PWR LED is to be turned OFF
2	Input	Low_true	BUTTON#	ATTN Button is Pressed	ATTN Button is NOT Pressed
3	Input	Low_true	PWRFLT#	PWR Fault in the VRM	No PWR Fault in the VRM
4	Input	Low_true	PRSNT#	Card Present in Slot	Card NOT Present in Slot
5	Output	High_true	PWREN	Power is to be enabled on the Slot	Power is NOT to be enabled on the Slot
6	Input	Low_true	MRL#	MRL is open	MRL is closed
7	Input	Low_true	GPI#	Power good on Slot	No Power good on Slot

The MCH will send Assert_intx/Deassert_intx or Assert_HPGPE/Deassert_HPGPE messages to the ESI port as virtual pin messages to enable the Intel® 631xESB/632xESB I/O Controller Hub take the appropriate action for handling the hot-plug (legacy/ACPI interrupt mode) in non-MSI mode.

7.13 Clocking

The following section describes the MCH Clocks.

7.13.1 Reference Clocks

The CORECLK reference clocks, operating at 266 MHz, are supplied to the MCH. This frequency is common between all processor bus agents. Phase matching between agents is required.

The FBD(0/1)CLK reference clocks, (herein referred to as FBDCLK) operating at 25% of the DDR II frequency (operating at half the SDRAM command-clock frequency... the SDRAM command-clock frequency and the FBD packet frequency are identical), are supplied to the MCH. This is the FBD PLL reference clock. This frequency is common between the MCH and DIMM's. Phase matching between agents is not required (plesiochronous). The MCH and DIMM's treat this frequency domain synchronously. The FBD unit-interval (UI) PLL outputs 24 x the FBDCLK frequency. E.g. For DDR II 667 MHz DIMM's, the FBDCLK frequency is 167 MHz and the UI (link) frequency is 4.0 GHz.

The PECLK reference clock, operating at 100MHz, is supplied to the MCH. This is the PCI Express PLL reference clock. The PCI Express flit PLL outputs 250 MHz. The PCI Express phit PLL outputs 2.5 GHz. The phit clock frequency must be tightly matched (mesochronous mode) between both PCI Express agents when spectrum-spreading is not employed. The phit clock frequency is common to both PCI Express agents when spectrum-spreading is employed. When the phit clock frequency is common to both PCI Express agents, no phase matching between them is required (plesiochronous mode). The MCH core treats this frequency domain asynchronously.

The CORECLK and FBDCLK reference clocks are derived from the same oscillator. The PECLK reference clock may be derived from a different oscillator.

The PCI Express interfaces operate asynchronously with respect to the core clock.

Table 7-33. Intel® 7300 Chipset MCH Frequencies

Domain	Frequency	Reference Clock
CORECLK	266 MHz	CORECLK
FSB 1X (common clock)	266 MHz	
FSB 2X (address)	533 MHz	
FSB 4X (data)	1,067 MHz	

Table 7-34. Intel® 7300 Chipset MCH Frequencies for Memory

DDR	Domain	Frequency	Reference Clock
533 MHz	FBD U	3.2 GHz	FBDCLK
	FBD packet	266 MHz	
	FBDCLK	133 MHz	
667 MHz	FBD U	4.0 GHz	
	FBD packet	333 MHz	
	FBDCLK	167 MHz	

Table 7-35. Intel® 7300 Chipset MCH Frequencies for PCI Express

Domain	Frequency	Reference Clock
PCI Express phit	2.5 GHz	PECLK
PCI Express flit	250 MHz	
PECLK	100 MHz	

7.13.2 JTAG

TCK is asynchronous to core clock. For private TAP register accesses, one TCK cycle is a minimum of 10 core cycles. The TCK high time is a minimum of 5 core cycles in duration. The TCK low time is a minimum of 5 core cycles in duration. The possibility of metastability during private register access is mitigated by circuit design. A metastability hardened synchronizer will guarantee an MTBF greater than 10^7 years.

For public TAP register accesses, TCK operates independently of the core clock.

7.13.3 SMBus Clock

The SMBus clock is synchronized to the core clock. Data is driven into the MCH with respect to the serial clock signal. Data received on the data signal with respect to the clock signal will be synchronized to the core using a metastability hardened synchronizer guaranteeing an MTBF greater than 10^7 years. The serial clock can not be active until 10 mS after RESETI# deassertion. When inactive, the serial clock should be deasserted (High). The serial clock frequency is 100 KHz.

7.13.4 GPIO Serial Bus Clock

The transmitted 100 KHz Virtual Pin Interface (VPI) clock (VPPSMBCLK) is derived from the core clock. The PCI Express Hot-Plug signals reside on the Virtual Pin Interface.

7.13.5 Clock Pins

Table 7-36. Clock Pins

Pin Name	Pin Description
CORECLKP	Processor bus clock
CORECLKN	Processor bus clock (Complement)
PECLKP	PCI Express clock
PECLKN	PCI Express clock (Complement)
FBD{01/23}CLKP	FB-DIMM clocks
FBD{01/23}CLKN	FB-DIMM clocks (Complement)
PSEL[2:0]	Processor bus clock speed select. Only “000” configuration is supported. These 3 pins should be pulled-down to ground
FBD{01/23}VCCA	Analog power supply for FB-DIMM PLLs
FBD{01/23}VSSA	Analog ground for FB-DIMM PLLs
FSBVCCA	Analog power supply for processor bus PLL
PEVCCA	Analog power supply for PCI Express PLLs
PEVSSA	Analog ground for PCI Express PLLs
COREVCCA	Analog power supply for Core PLL
COREVSSA	Analog ground for Core PLL
TCK	TAP clock
VPPSMBSCL	PCI Express Hot Plug (Virtual Pin Port) clock
CFGSMBCLK	System Management SMBus clock
SPD{0/1/2/3}SMBCLK	FB-DIMM Channel Serial Present
XDPSTBP_N	XDP Port data strobe
XDPSTBN_N	XDP Port data strobe (Complement)
FSB{0/1/2/3}STBP[3:0]_N	Processor bus data strobes
FSB{0/1/2/3}STBN[3:0]_N	Processor bus data strobes (Complements)
FSB{0/1/2/3}ADSTB[1:0]_N	Processor bus address strobes

7.13.6 High Frequency Clocking Support

7.13.6.1 Spread Spectrum Support

The MCH PLLs will support Spread Spectrum Clocking (SSC). SSC is a frequency modulation technique for EMI reduction. Instead of maintaining a constant frequency, SSC modulates the clock frequency/period along a predetermined path, i.e., the modulation profile.The MCH is designed to support a nominal modulation frequency of 30 KHz with a down spread of 0.5%.

7.13.6.2 Stop Clock

PLLs in the MCH cannot be stopped.

7.13.6.3 Jitter

The FB-DIMM UI clocks are produced by PLLs that multiply the FBDCLK frequency by 12. The PCI Express phit clocks are produced by PLLs that multiply the PECLK frequency by 25. These multi-GHz phit clocks require ultra-clean sources, ruling out all but specifically-crafted low-jitter clock synthesizers.

7.13.6.4 External Reference

An external crystal oscillator is the preferred source for the PLL reference clock. A spread spectrum frequency synthesizer that meets the jitter input requirements of the PLL is acceptable.

7.13.6.5 PLL Lock Time

The reference clocks must be stable 1ms before the assertion of the PWRGOOD signal. The assertion of the PWRGOOD signal initiates the PLL lock process. External clocks dependent on PLLs are GPIO clock and SMBus clock.

7.13.6.6 Other PLL Characteristics

The PLL VCOs oscillate continually from power-up. The PLL output dividers consistently track the VCO, providing pulses to the clock trees. Logic that does not receive an asynchronous reset can thus be reset “synchronously”.

A “locked” PLL will only serve to prove that the feedback loop is continuous. It will not prove that the entire clock tree is continuous.

7.13.6.7 Analog Power Supply Pins

The MCH incorporates seven PLLs. Each PLL requires an Analog Vcc and Analog Vss pad and external LC filter. Therefore, there will be external LC filters for the MCH. IMPORTANT: The filter is NOT to be connected to board Vss. The ground connection of the filter will be routed through the package and grounded to on-die Vss.

7.13.6.8 I/O Interface Metastability

PCI Express can be operated frequency-locked to the core. Flits are fifteen-sixteenths of the core frequency in 266 MHz mode, three-quarters of the core frequency in 333MHz mode.

However, the phase between the frequency-locked domains is not controlled. This scheme results in the possibility of a metastability resonance where, e.g., the commands generated by the core miss setup and hold to I/O every time. This condition can be tolerated by carefully hardened metastability design.

7.14 Error List

This section provides a summary of errors detected by the Intel® 7300 Chipset. In the following table, errors are listed by the unit / interfaces. Some units / interfaces may provide additional error logging registers.

The following table provides the list of detected errors of a the MCH.

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 1 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
F1	Request/Address Parity Error	MCH monitors the address and request parity signals on the FSB. A parity discrepancy over these fields during a valid request. MCH only detects this error caused by CPUs.	Fatal	FERR_FAT_FSB/NERR_FAT_ FSB. NRECFSB, NRECFSB_ADDRH, NRECFSB_ADDRL for FERR only.	Complete transaction on FSB with response (non-hard fail response)
F2	Unsupported Request or data size on FSB.	MCH detected an FSB Unsupported transaction. MCH only detects this error caused by CPUs.	Fatal	FERR_FAT_FSB/NERR_FAT_ FSB. NRECFSB for FERR only.	Treat as NOP. No Data Response or Retry by MCH
F6	Data Parity Error	MCH monitors the data/parity signals on the FSB. Set when the MCH detects an parity error during the data transfer. MCH only detects this error caused by CPUs.	UnCorr	FERR_NF_FSB/NERR_NF_ FSB. RECFSB for FERR only	Received a parity error. Poison Data and forward to the appropriate interface.
F7	Detected MCERR	MCH detected that a processor issued an MCERR.	UnCorr	FERR_NF_FSB/NERR_NF_ FSB. based on POC[5] setting	If (receive an MCERR) forward the MCERR to the other FSB bus, adhering to the MCERR protocol
F8	B-INIT	MCH detected that a processor issued an B-INIT.	UnCorr	FERR_NF_FSB/NERR_NF_ FSB. based on POC[6] setting	Do not propagate to other FSB bus, reset arb. unit, and programatically reset platform
F9	FSB protocol Error	MCH detected FSB protocol error, for example, HitM on BIL and HitM on EWB.	Fatal	FERR_FAT_FSB/NERR_FAT_ FSB. NRECFSB, NRECFSB_ADDRH, NRECFSB_ADDRL for FERR only.	Complete transaction on FSB with response

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 2 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
DMA0	Source Address Error	The DMA channel sets this bit indicating that the current descriptor has an illegal source address.	Fatal	Log FERR_CHANERR/NERR_ CHANERR,FERR_CHANSTS, FERR_CHANCMD, FERR_DESC_CTRL, FERR_SADDR, FERR_DADDR, FERR_TRANSFER_SIZE, FERR_NADDR, FERR_CHANCMP(Check Channel Completion enable) by taking a snap shot of the respective register fields when an error is met. The SADDR and DADDR will be the runtime addresses that the DMA engine is executing	Halt DMA engine
DMA1	Destination Address Error	The DMA channel sets this bit indicating that the current descriptor has an illegal destination address.	Fatal		Halt DMA engine
DMA2	Next Descriptor Address Error	The DMA channel sets this bit indicating that the next descriptor in the link list has an illegal address.	Fatal		Halt DMA engine
DMA3	Descriptor Error	The DMA channel sets this bit indicating that the current transfer has encountered an error (not otherwise covered under DMA error bits) when executing a DMA descriptor.	Fatal		An illegal next descriptor address flagged by the system Address decoder, which the DMA engine encounters in the current descriptor after having successfully completed the data transfer for the current descriptor including any associated completions/interrupts Halt DMA engine
DMA4	Chain Address Value Error	The DMA channel sets this bit indicating that the CHAINADDR register has an illegal address including an alignment error (not on a 64-byte boundary).	Fatal	Log FERR_CHANERR/NERR_CHANER R,FERR_CHANSTS, FERR_CHANCMD, FERR_DESC_CTRL, FERR_SADDR, FERR_DADDR, FERR_TRANSFER_SIZE, FERR_NADDR, FERR_CHANCMP(Check Channel Completion enable) by taking a snap shot of the respective register fields when an error is met. The SADDR and DADDR will be the runtime addresses that the DMA engine is executing	Halt DMA engine
DMA5	CHANCMD Error	The DMA channel sets this bit indicating that a write to the CHANCMD register contained an invalid value (e.g. more than one command bit set)	Fatal		Halt DMA engine
DMA6	Chipset Data Parity error	The DMA channel sets this bit indicating that there is a data parity error during a read/write operation of a given DMA descriptor.	Fatal		Halt DMA engine

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 3 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
DMA8	Read Data error	The DMA channel sets this bit indicating that a read could not be completed (e.g. starvation).	Fatal	Log FERR_CHANERR/NERR_CHANERR/FERR_CHANCMD, FERR_DESC_CTRL, FERR_SADDR, FERR_DADDR, FERR_TRANSFER_SIZE, FERR_NADDR, FERR_CHANCMP NERR_CHANCMD, NERR_DESC_CTRL, NERR_SADDR, NERR_DADDR, NERR_TRANSFER_SIZE, NERR_NADDR, NERR_CHANCMP by taking a snap shot of the respective register fields when an error is met. The SADDR and DADDR will be the runtime addresses that the DMA engine is executing	Halt DMA engine
DMA9	Write Data error	The DMA channel sets this bit indicating that a write was unable to be completed at the destination(e.g. no space available in DM).	Fatal		Halt DMA engine
DMA10	Descriptor Control Error	The DMA channel sets this bit indicating that the current descriptor has an illegal control field value in the “desc_control” field.	Fatal		Halt DMA engine
DMA11	Descriptor length Error	The DMA channel sets this bit indicating that the current transfer has an illegal length field value	Fatal		Halt DMA engine
DMA12	Completion Address Error	The DMA channel sets this bit indicating that the completion address register was configured to an illegal address	Fatal		Halt DMA engine
DMA13	Interrupt Configuration Error	The DMA channel sets this bit indicating that the interrupt related registers were not configured properly and an interrupt could not be generated	Fatal		Halt DMA engine

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 4 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
IO0	PCI Express - Data Link Layer Protocol Error	MCH detects a DL layer protocol error from the DLLP.	Default=Fatal (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (UNCERRSEV) Log RPERRSTS for IO1, IO11 and IO17. Log UNCERRSTS for their respective Error Types. Log the first error pointer for UNCERRSTS in AERRCAPCTRL. Log CORRERSTS for their respective Error Types. Log PEXDEVSTS for IO12 and other I/O errors based on UNCERSEV	Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)
IO1	PCI Express - Received Fatal Error Message	MCH received a Fatal error message from the south bridge.	Fatal		Log header of packets with errors
IO2	PCI Express - Received Unsupported Request	Received an unsupported request, similar to master abort.	Default=UnCorr (Check UNCERRSEV)		Log header of packet Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)
IO4	PCI Express - Poisoned TLP	Received a poisoned transaction layer packet from the South Bridge.	Default=UnCorr (Check UNCERRSEV)		Log header of packets with errors Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)
IO5	PCI Express - Flow Control Protocol Error	MCH has detected a PCI Express Flow Control Protocol Error	Default=Fatal (Check UNCERRSEV)		Log header of packets with errors Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)
IO6	PCI Express - Completion Time-out	Pending transaction was ACKed in the data link layer but not within the time limit.	Default=UnCorr (Check UNCERRSEV)		Log header of packets with errors Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)
IO7	PCI Express - Completer Abort	Received return CA status for horrible error on the component. This is equivalent to a target abort on PCI.	Default=UnCorr (Check UNCERRSEV)		Log header of packets with errors Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)
IO8	PCI Express - Unexpected Completion Error	Received a Completion RequestorID that matches the requestor but the Tag does not match any pending entries.	Default=UnCorr (Check UNCERRSEV)		Log header of packets with errors Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)
IO9	PCI Express - Malformed TLP	Received a transaction layer packet that does not follow the TLP formation rules.	Default=UnCorr (Check UNCERRSEV)		Log header of packets with errors Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 5 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
IO10	PCI Express - Receive Buffer Overflow Error	Receiver gets more data or transactions than credits allow.	Default=Fatal (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (UNCERRSEV) Log RPERRSTS for IO1, IO11 and IO17. Log UNCERRSTS for their respective Error Types. Log the first error pointer for UNCERRSTS in AERRCAPCTRL. Log CORRERSTS for their respective Error Types. Log PEXDEVSTS for IO12 and other I/O errors based on UNCERSEV	Log header of packets with errors Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)
IO11	PCI Express - Received NonFatal Error Message	MCH received a NonFatal error message from the south bridge.	UnCorr		Log header of packets with errors
IO12	PCI Express - Receiver Error	Log header of packets with errors	Corr		Log header of packets with errors
IO13	PCI Express - Bad TLP Error	Received bad CRC or a bad sequence number in a transport layer packet.	Corr		Log header of packets with errors
IO14	PCI Express - BAD DLLP	Received bad CRC in a data link layer packet.	Corr		Log header of packets with errors
IO15	PCI Express - Replay_Num Rollover	Replay maximum count for the Retry Buffer has been exceeded.	Corr		Log header of packets with errors
IO16	PCI Express - Replay Timer Time-out	Replay timer timed out waiting for an Ack or Nak DLLP.	Corr		Log header of packets with errors
IO17	PCI Express - Received Correctable Error Message	MCH received a correctable error message from the south bridge.	Corr		Log header of packets with errors
IO18	ESI reset time-out	Did not receive ESI CPU_Reset_Done_Ack or CPU_Reset_Done_Ack_Secrets messages within T_{10max} after assertion of processor RESET# while PWRGOOD was asserted	Fatal	Log PEX_FAT_FERR/NERR	Deassert processor RESET#. Necessary to prevent processor thermal runaway.
IO19	Surprise Link Down Error	IOU LTSSM detected a link down condition (surprise) during normal operation	Fatal	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (UNCERRSEV)	Link went down suddenly and status bits are set for software to take any action
B1	MCH -Parity Error from DM (Do not Include Poisoned Data)	MCH detected internal DM parity error. (This error was not generated by receiving bad data from an external interface)	Fatal	FERR_FAT_INT/NERR_FAT_INT and NRECINT	log DM Entry on FERR.
B2	MCH -Multi-Tag Hit from snoop filter on any SF lookup port	MCH detected multiple hits in the SF lookup on any SF lookup port	Fatal	FERR_FAT_INT/NERR_FAT_INT and NRECSF	Log, Hit/Miss, Set, Tag, State and Presence vector on FERR.

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 6 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
B3	MCH-Coherency Violation EWB Error	MCH detected a cache coherency protocol error for EWB. Any requester not in 'E/M' state in the Snoop Filter.	Fatal	FERR_FAT_INT/NERR_FAT_ INT NRECINT and NRECBF	Log CE entry on FERR
B4	Virtual Pin Interface Error	MCH detected an error on the virtual pin interface	Fatal	FERR_FAT_INT/NERR_FAT_ INT and NRECINT	
B5	MCH-Address Map Error	MCH detected an address mapping error due to software programming error. Detected mapping errors: transaction decodes to more than one type (mem, IO, cfg, interrupt etc...), transaction maps to more than one PCIe port, VGA access with VGA enabled on more than one port. Programming errors are described in Section 6, "System Address Map" .	UnCorr	FERR_NF_INT/NERR_NF_INT and NRECINT	MCH might malfunction.
B6	Single bit ECC error on snoop filter lookup	MCH detected a hit in SF lookup and the entry has a single bit ECC error, or MCH detected a miss in SF lookup and the victim entry has a single bit error.	Corr	FERR_NF_INT/NERR_NF_INT and RECSF	Log, Hit/Miss, Set, Tag, State and Presence vector on FERR.
B7	Multiple bit ECC error on snoop filter lookup	MCH detected a multiple ECC error in any of the ways during snoop filter lookup	Fatal	FERR_FAT_INT/NERR_FAT_ INT and NRECSF	Log, Hit/Miss, Set, Tag, State and Presence vector on FERR.
B8	MCH-Coherency violation BIL Error	MCH detected a cache coherency protocol error for a BIL. Any requestor from the bus that issued BIL not present in the SF.	Non Fatal	FERR_NF_INT/NERR_NF_ INT and NRECINT, and NRECSF	Log CE entry on FERR This applies to SF enable mode only

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 7 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
M1	Memory Write error on non-redundant retry or FBD configuration Write error on retry	MCH detected a corrupted acknowledgement on a retry of a non-redundant memory write or MCH detected a corrupted acknowledgement on a retry of an FBD configuration write	Fatal	FERR_FAT_FBD NERR_FAT_FBD NRECMEM NRECFGLOG	Memory read: poison to requestor, update NRECMEM Configuration read: master-abort to requestor, update CFGLOG All others: drop.
M2	Memory or FBD configuration CRC Read error.	The MCH detected a corrupted CRC error on a non-redundant retry at a memory or FBD configuration read.	Fatal	FERR_FAT_FBD NERR_FAT_FBD NRECMEM NRECFGLOG NRECFBD	Memory read: poison to requestor, update NRECMEM Configuration read: master-abort to requestor, update CFGLOG All others: drop.
M3	Tmid thermal event with S-CLTT disabled	S-CLTT is disabled and the thermal sensor transitions from “below Tmid” to “above Tmid”.	Fatal	FERR_FAT_FBD NERR_FAT_FBD	
M4	Uncorrectable Data ECC Error on FB-DIMM Replay	The MCH detected an uncorrectable data ECC error during replay of the head of the FB-DIMM replay queue	Uncorr	FERR_NF_FBD NERR_NF_FBD	Poison to requestor. Don’t log error again... it was logged when the replay was launched.
M5	Aliased Uncorrectable Non-Mirrored Demand Data ECC Error	The MCH determined that a normally “correctable” error could be an aliased (x8 only) full device failure plus an additional single bit error.	Rec	FERR_NF_FBD NERR_NF_FBD RECMEM REDMEM UERRCNT	Re-read once. If ECC is uncorrectable with good CRC after re-read, then poison the data in memory and to the requestor. If correctable after re-read, then correct the data in memory and to the requestor.

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 8 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
M6	Aliased Uncorrectable Mirrored Demand Data ECC Error	In mirrored mode, the MCH determined that a normally "correctable" error could be an aliased (x8 only) full device failure plus an additional single bit error.	Rec	FERR_NF_FBD NERR_NF_FBD RECMEM REDMEM UERRCNT	First redundant read to branch X fails. MCH performs a fast reset on both branches X and Y. If both branches pass, then replay on branch Y. If branch X fails the disable branch X and replay on branch Y. If both branches fail or branch Y fails disable branch X and poison data. Under these conditions we get an M1 error. Second redundant read to branch X fails with an uncorrectable error. Perform fast reset and disable branch X and replay on branch Y.
M7	Aliased Uncorrectable Spare-Copy Data ECC Error	During a Sparing copy read from the failing DIMM the MCH determined that a normally "correctable" error could be an aliased (x8 only) full device failure plus an additional single bit error.	Rec	FERR_NF_FBD NERR_NF_FBD RECMEM REDMEM UERRCNT	Re-read once. If ECC is uncorrectable with good CRC after re-read, then poison the data in the spare DIMM or the off-line branch. If correctable after re-read, then correct the data in the spare DIMM or the off-line branch.
M8	Aliased Uncorrectable Patrol Data ECC Error	During a Patrol Scrub, the MCH determined that a normally "correctable" error could be an aliased (x8 only) full device failure plus an additional single bit error.	Rec	FERR_NF_FBD NERR_NF_FBD RECMEM REDMEM UERRCNT	The patrol read is dropped.
M9	Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC Error	The MCH detected uncorrectable data with good CRC.	Rec	FERR_NF_FBD NERR_NF_FBD RECMEM UERRCNT	Re-read once. If ECC is uncorrectable with good CRC after re-read, then poison the data in memory and to the requestor. If correctable after re-read, then correct the data in memory and to the requestor. Does not include poisoned northbound data.

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 9 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
M10	Non-Aliased Uncorrectable Mirrored Demand Data ECC Error	In mirrored mode, the MCH detected uncorrectable or poisoned data with good CRC.	Rec	FERR_NF_FBD NERR_NF_FBD RECMEM UERRCNT	First redundant read to branch X fails. MCH performs a fast reset on both branches X and Y. If both branches pass, then replay on branch Y. If branch X fails the disable branch X and replay on branch Y. If both branches fail or branch Y fails disable branch X and poison data. Under these conditions we get an M1 error. Second redundant read to branch X fails with an uncorrectable error. Perform fast reset and disable branch X and replay on branch Y
M11	Non-Aliased Uncorrectable Spare-Copy Data ECC Error	The MCH detected uncorrectable data with good CRC from the failing DIMM rank during a sparing copy.	Rec	FERR_NF_FBD NERR_NF_FBD RECMEM UERRCNT	Re-read once. If ECC is uncorrectable with good CRC after re-read, then poison the data in the spare DIMM or the off-line branch. If correctable after re-read, then correct the data in the spare DIMM or the off-line branch. Does not include poisoned northbound data.
M12	Non-Aliased Uncorrectable Patrol Data ECC Error	During a patrol scrub, the MCH detected uncorrectable data with good CRC.	Rec	FERR_NF_FBD NERR_NF_FBD RECMEM UERRCNT	The patrol read is dropped.
M13	Memory write error	The MCH detected a corrupted acknowledgement on a first attempt at a memory write.	Rec	FERR_NF_FBD NERR_NF_FBD RECMEM	Fast reset and Initiate replay.
M14	FBD Configuration write error	The MCH detected a corrupted acknowledgement on a first attempt at an FBD Configuration write	Rec	FERR_NF_FBD NERR_NF_FBD CFGLOG	Fast reset and Initiate replay.

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 10 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
M15	Memory or FBD Configuration CRC read error.	The MCH detected a corrupted CRC error on s non-redundant first attempt, a redundant first attempt, or a redundant retry at a memory or FBD configuration read.	Rec	FERR_NF_FBD NERR_NF_FBD RECFBD	1st redundant memory read: re-read once from other image. Replay redundant memory read: auto-degrade All others: fast reset and initiate replay from the same image, branch, or channel.
M17	Correctable Non-Mirrored Demand Data ECC Error.	The MCH detected correctable data.	Corr	FERR_NF_FBD NERR_NF_FBD RECMEM REDMEM CERRCNT BADCNT BADRANK	Correct the data in memory and to the requestor.
M18	Correctable Mirrored Demand Data ECC Error	The MCH detected correctable data.	Corr	FERR_NF_FBD NERR_NF_FBD RECMEM REDMEM CERRCNT BADCNT BADRANK	Correct the data in memory and to the requestor.
M19	Correctable Spare-Copy Data ECC Error	The MCH detected correctable data from the failing DIMM rank during a sparing copy.	Corr	FERR_NF_FBD NERR_NF_FBD RECMEM REDMEM CERRCNT BADCNT BADRANK	Correct the data in the spare DIMM or the off-line branch.
M20	Correctable Patrol Data ECC Error	During a patrol scrub, the MCH detected correctable data.	Corr	FERR_NF_FBD NERR_NF_FBD RECMEM REDMEM CERRCNT BADCNT BADRANK	Correct the data in memory.
M21	FB-DIMM Northbound parity error on a FB-DIMM Sync Status	The MCH detected a northbound parity error on a Sync Status	Corr	FERR_NF_FBD NERR_NF_FBD RECFBD	Drop. If sync was issued to prepare a fast reset for alert recovery then replay any queued configuration command destined for an alerting DIMM or a DIMM with a corrupted status CRC. **WARNING:** Possible double DIMM configuration command execution may incur undesirable side-effects.
M22	SPD protocol Error	The MCH detected an SPD interface error.	Corr	FERR_NF_FBD NERR_NF_FBD	Successive correction attempts performed by software.

Table 7-37. Intel® 7300 Chipset Chipset Error List (Sheet 11 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
M23	Non-Redundant Fast Reset Timeout	The MCH hit a timeout on a non-redundant fast reset	Fatal	FERR_FAT_FBD NERR_FAT_FBD	On a read, this will poison data to requester. On a write, the transaction is dropped. On a config read, master abort is sent to requester.
M25	Memory Write error on redundant retry	The MCH detected a corrupted acknowledgement on a redundant retry at a memory write	Rec	FERR_NF_FBD NERR_NF_FBD RECMEM	Fast resets performed on both branches, followed by a replay on the branch with the failed write.
M26	Redundant Fast Reset Timeout	The MCH hit a timeout on a redundant fast reset	Rec	FERR_NF_FBD NERR_NF_FBD	On a redundant read, if the branch that had the initial error (that caused the fast reset) has a fast reset timeout, that branch is disabled and the read is tried on the other branch. If the other branch has a fast reset timeout, poisoned data is returned to the requester. On a redundant write, if the branch that had the initial write error (that caused the fast error) has a fast reset timeout, that branch is disabled. If the other branch has a fast reset timeout, the transaction will be replayed on the original branch (and if it fails that, it will be dropped). On a redundant write fast reset, if both branches timeout, then the write is dropped.
M27	DIMM-Spare Copy initiated	DIMM-Spare copy started due to Correctable Error Threshold exceeded, or forced spare copy	Corr	FERR_NF_FBD NERR_NF_FBD	Start DIMM-spare copy
M28	DIMM-Spare Copy complete	DIMM-Spare copy completed normally	Corr	FERR_NF_FBD NERR_NF_FBD	No Action

§

8 Thermal Specifications

8.1 Thermal Design Power (TDP)

Analysis indicates that real applications are unlikely to cause the MCH component to consume maximum power dissipation for sustained time periods. Therefore, in order to arrive at a more realistic power level for thermal design purposes, Intel characterizes power consumption based on known platform benchmark applications. The resulting power consumption is referred to as the Thermal Design Power (TDP). TDP is the target power level to which the thermal solutions should be designed. TDP is not the maximum power that the chipset can dissipate.

For TDP specifications, see Table 8-1. FC-BGA packages have poor heat transfer capability into the board and have minimal thermal capability without a thermal solution. Intel recommends that system designers plan for a heatsink when using the Intel® 7300 Chipset Memory Controller Hub (MCH).

8.2 Case Temperature

To ensure proper operation and reliability of the Intel® 7300 Chipset Memory Controller Hub (MCH), the case temperatures on top of the Integrated Heat Spreader must be at, or between, the maximum operating temperature, as specified in Table 8-1. System and/or component level thermal solutions are required to maintain these temperature specifications.

Table 8-1. Intel® 7300 Chipset MCH Thermal Specifications

Parameter	Value	Notes
T_{case_max}	77°C	
T_{case_min}	5°C	
TDP	47W	4 Memory Channels
TDP	40W	2 memory Channels
Idle Power	33W	4 Memory Channels
Idle Power	27W	2 Memory Channels

§

9 Mechanical Specificatons

The Intel® 7300 Chipset consists of three individual components: the MCH, the Intel® 631xESB/632xESB I/O Controller Hub and the Intel 6700PXH/6702PXH 64-bit PCI Hub. The Intel® 7300 Chipset MCH component uses a 49.5mm squared, 12-layer flip chip ball grid array (FC-BGA) package (see Figure 9-1). For information on the PXH package, refer to the *Intel® 6700PXH 64-bit PCI Hub/6702PXH 64-bit PCI Hub (PXH/PXH-V) Datasheet*. For information on the Intel® 631xESB/632xESB I/O Controller Hub package, refer to the *Intel® 631xESB/632xESB I/O Controller Hub Datasheet*.

9.1 Mechanical Drawings

Figure 9-1. Intel® 7300 Chipset MCH Package Drawing (Sheet 1 of 2)

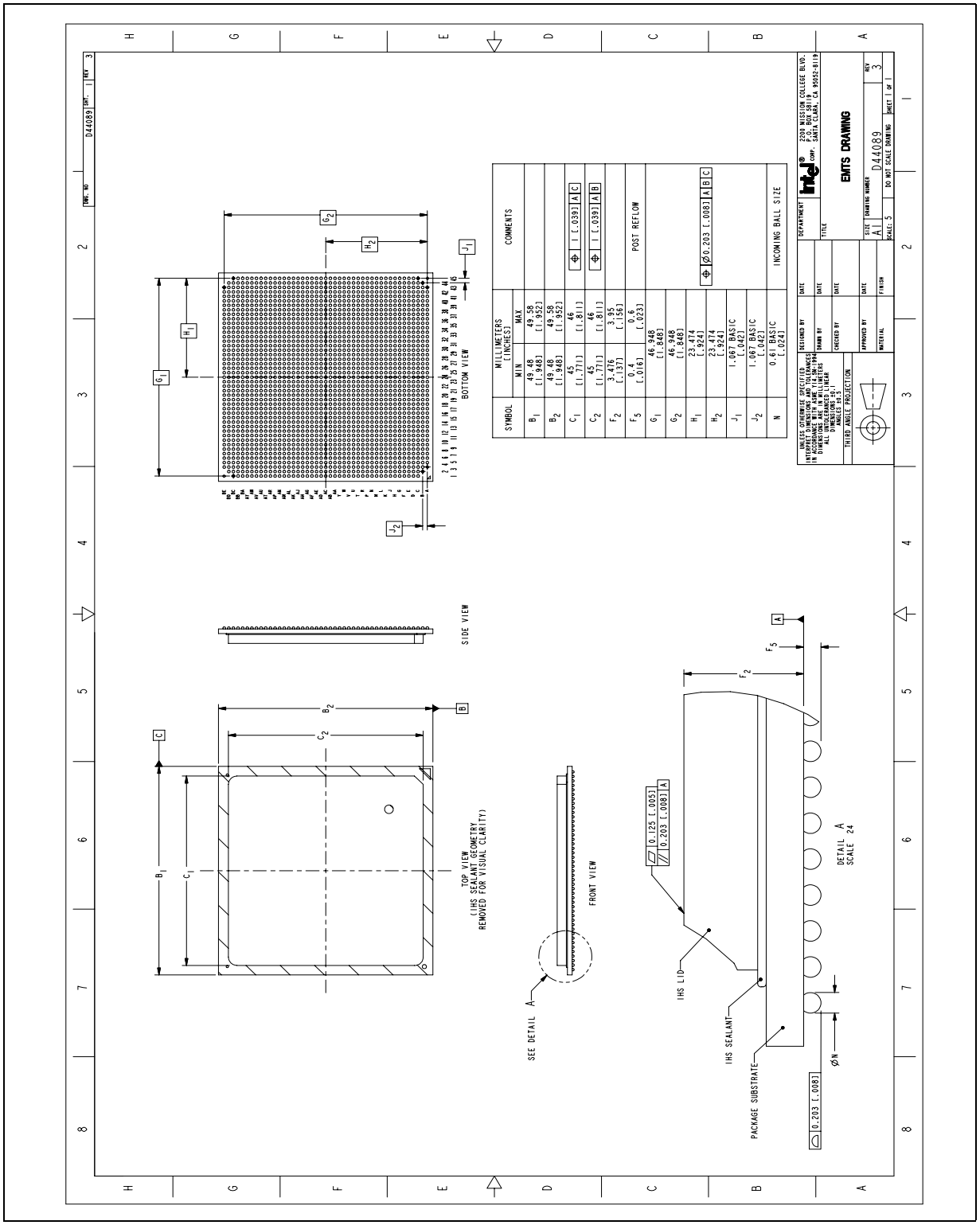
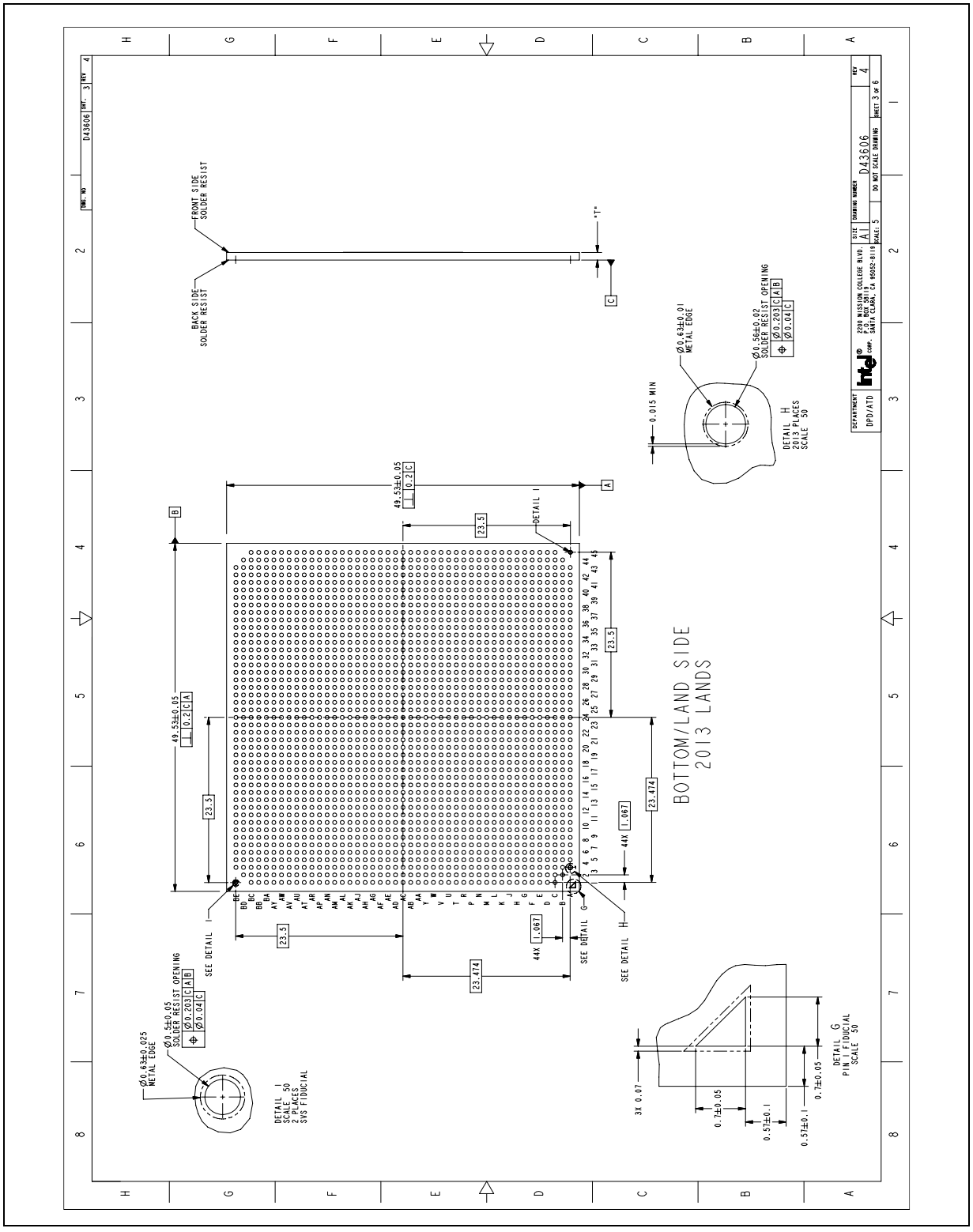


Figure 9-1. Intel® 7300 Chipset MCH Package Drawing (Sheet 2 of 2)



9.2 Mechanical Requirements

The Intel® 7300 Chipset MCH package has an IHS which is capable of sustaining a maximum static normal load of 15-lbf. This mechanical load limit must not be exceeded during heatsink installation, mechanical stress testing, standard shipping conditions and/or any other use condition.

Notes:

1. The heatsink attach solutions must not include continuous stress onto the chipset package with the exception of a uniform load to maintain the heatsink-to-package thermal interface. And, this uniform load should not exceed the maximum allowable static normal compressive load of 15 lbf.
2. These specifications apply to uniform compressive loading in a direction perpendicular to the IHS top surface.
3. These specifications are based on limited testing for design characterization. Loading limits are for the package only.

Table 9-1. Maximum Design limits[1]

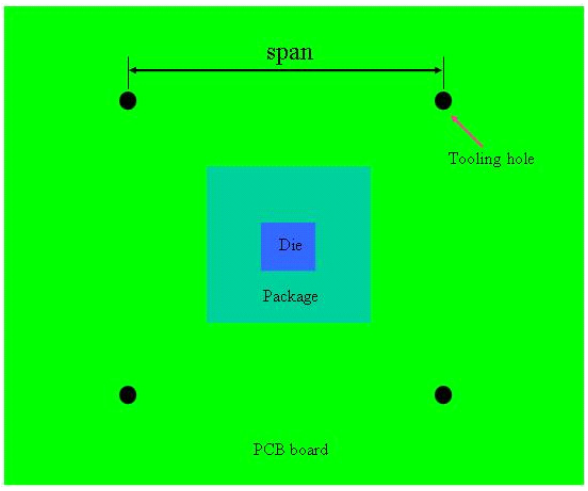
Attributes	Maximum design limit by board thickness, 93mil (2.36mm)	Comments
Static Compressive	15lbf with 55mm span[1] (without back plate) 20lbf with Pb-free (with back plate)	Meet Intel® 7300 Chipset reliability Requirement using PbSn (worse case). Lead Free has better performance than PbSn.
Dynamic Bend and Dynamic Compressive	1300 με	

Notes:

1. 1. These maximum design limits are for 1.067 ball pitch with package form factor less than 49.53 mm sq, 3 corner BGA balls depopulated and 12 BGAs NCTF at each corner

See Figure 9-2 for span definition.

Figure 9-2. Span definition for Static Compressive Test



§

10 Testability

10.1 JTAG Port

Each component in the Intel® 7300 Chipset includes a Test Access Port (TAP) slave which complies with the IEEE 1149.1 (JTAG) test architecture standard. Basic functionality of the 1149.1- compatible test logic is described here, but this document does not describe the IEEE 1149.1 standard in detail. For this, the reader is referred to the published standard1, and to the many books currently available on the subject.

10.1.1 JTAG Access to Configuration Space

JTAG has become a name that is synonymous with the IEEE 1149.1 test access port (TAP). Besides the boundary scan capabilities for low speed buses and pins, it provides an inexpensive serial interface port to up/download data to and from the chip. Throughout this document any reference to JTAG will imply the test access port (TAP) and the private chains that it is connected too, unless specifically mentioning the boundary scan attributes.

The feature described here is a JTAG private data chain that initiate a configuration request to the components configuration arbitration logic. During platform debug it is helpful to have a back door access to register space to determine correct configuration states. The In-Target Probe (ITP) provides an effective observation capability that links the hardware and the user together to examine and control a number of DFT and debug features.

Access to a component's configuration space must be non-blocking to a JTAG initiated configuration request to the MCH's register space. Since the MCH can source configuration transactions to other components and an errant configuration transaction that could potentially hang the system and prevent a JTAG access to the MCH's configuration space. An additional chain is provided to ensure the ITP tool has unconditional access privilege to the MCH in case there are configuration transaction hangs from another source.

10.1.2 TAP Signals

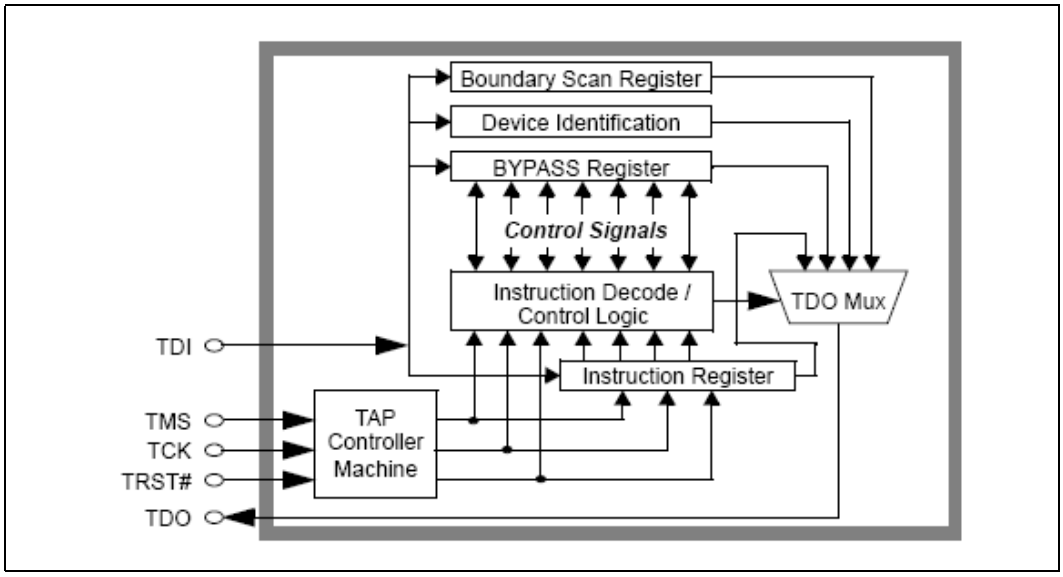
The TAP logic is accessed serially through 5 dedicated pins on each component as shown in Table 10-1.

Table 10-1. TAP Signal Definitions

TCK	TAP Clock input
TMS	Test Mode Select. Controls the TAP finite state machine.
TDI	Test Data Input. The serial input for test instructions and data.
TDO	Test Data Output. The serial output for the test data.
TRST#	Test Reset input.

TMS, TDI and TDO operate synchronously with TCK (which is independent of all other clocks). TRST# is an asynchronous reset input signal. This 5-pin interface operates as defined in the 1149.1 specification. A simplified block diagram of the TAP used in the Intel® 7300 Chipset components is shown in Figure 10-1.

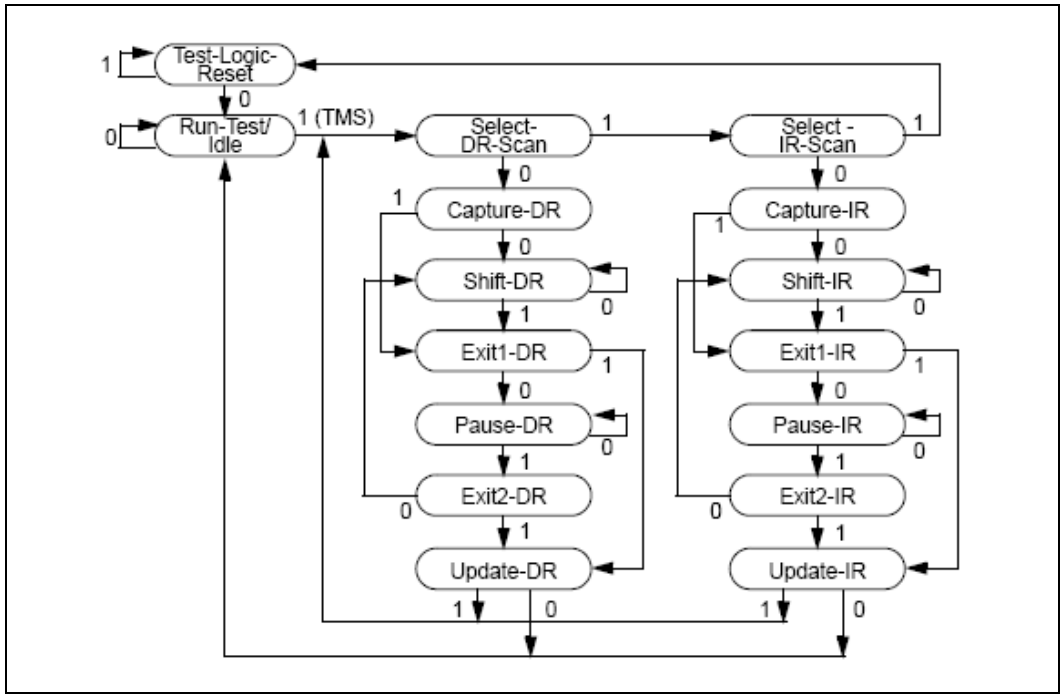
Figure 10-1. Simplified TAP Controller Block Diagram



The TAP logic consists of a finite state machine controller, a serially-accessible instruction register, instruction decode logic and data registers. The set of data registers includes those described in the 1149.1 standard (the bypass register, device ID register, and so forth.),Accessing the TAP Logic

The TAP is accessed through an 1149.1-compliant TAP controller finite state machine, which is illustrated in Figure 10-1. The two major branches represent access to either the TAP Instruction Register or to one of the component-specific data registers. The TMS pin controls the progress through the state machine. TAP instructions and test data are loaded serially (in the Shift-IR and Shift-DR states, respectively) using the TDI pin. A brief description of the controller's states follows; refer to the IEEE 1149.1 standard for more detailed descriptions.

Figure 10-2. TAP Controller State Machine



The following list describes the behavior of each state in the TAP.

Test-Logic-Reset: In this state, the test logic is disabled so that normal operation of the device can continue unhindered. The instruction in the Instruction Register is forced to IDCODE. The controller is guaranteed to enter Test- Logic-Reset when the TMS input is held active for at least five clocks. The controller also enters this state immediately when TRST# is pulled active. The TAP controller cannot leave this state as long as TRST# is held active.

Run-Test/Idle: The TAP idle state. All test registers retain their previous values.

Capture-IR: In this state, the shift register contained in the Instruction Register loads a fixed value (of which the two least significant bits are "01") on the rising edge of TCK. The parallel, latched output of the Instruction Register ("current instruction") does not change.

Shift-IR: The shift register contained in the Instruction Register is connected between TDI and TDO and is shifted one stage toward its serial output on each rising edge of TCK. The output arrives at TDO on the falling edge of TCK. The current instruction does not change.

Pause-IR: Allows shifting of the Instruction Register to be temporarily halted. The current instruction does not change.

Update-IR: The instruction which has been shifted into the Instruction Register is latched onto the parallel output of the Instruction Register on the falling edge of TCK. Once the new instruction has been latched, it remains the current instruction until the next Update-IR (or until the TAP controller state machine is reset).

Capture-DR: In this state, the Data Register selected by the current instruction may capture data at its parallel inputs.

Shift-DR: The Data Register connected between TDI and TDO as a result of selection by the current instruction is shifted one stage toward its serial output on each rising edge of TCK. The output arrives at TDO on the falling edge of TCK. The parallel, latched output of the selected Data Register does not change while new data is being shifted in.

Pause-DR: Allows shifting of the selected Data Register to be temporarily halted without stopping TCK. All registers retain their previous values.

Update-DR: Data from the shift register path is loaded into the latched parallel outputs of the selected Data Register (if applicable) on the falling edge of TCK. This and Test-Logic-Reset are the only controller states in which the latched paralleled outputs of a data register can change.

All other states are temporary controller states, used to advance the controller between active states. During such temporary states, all test registers retain their prior values.

10.1.3 Reset Behavior of the TAP

The TAP and its related hardware are reset by transitioning the TAP controller finite state machine into the Test-Logic-Reset state. Once in this state, all of the reset actions listed in Figure 10-2 are performed. The TAP is completely disabled upon reset (i.e. by resetting the TAP, the device will function as though the TAP did not exist).

Table 10-2. TAP Reset Actions

TAP Logic Affected	TAP Reset State Action	Related TAP Instructions (instr equivalent to reset is highlighted)
TAP instruction register	IDCODE	—
Boundary scan logic	Disabled	EXTEST
TDO pin	Tri-stated	—

The TAP can be transitioned to the Test-Logic-Reset state in one of two ways:

- Assert the TRST# pin at any time. This asynchronously resets the TAP controller.
- Hold the TMS pin high for 5 consecutive cycles of TCK. This is guaranteed to transition the TAP controller to the Test-Logic-Reset state on a rising edge of TCK.

Cycling power on a device does not ensure that the TAP is reset. System designers must utilize one of the two methods stated above to reset the TAP. The method used depends on the manufacturing and debug requirements of the system.

10.1.4 Clocking the TAP

There is no minimum frequency at which the Intel® 7300 Chipset TAP will operate. Because the private chains are synchronized to the local core clock of that chain there is a maximum rate relative to the core that the interface can operate. The ratio is 12:1 providing a maximum rate of 27 MHz for a core frequency of 333 MHz.

10.1.5 Accessing the Instruction Register

Figure 10-3 shows the (simplified) physical implementation of the TAP instruction register. This register consists of a 7-bit shift register (connected between TDI and TDO), and the actual instruction register (which is loaded in parallel from the shift register). The parallel output of the TAP instruction register goes to the TAP instruction decoder.

Figure 10-3. TAP Instruction Register

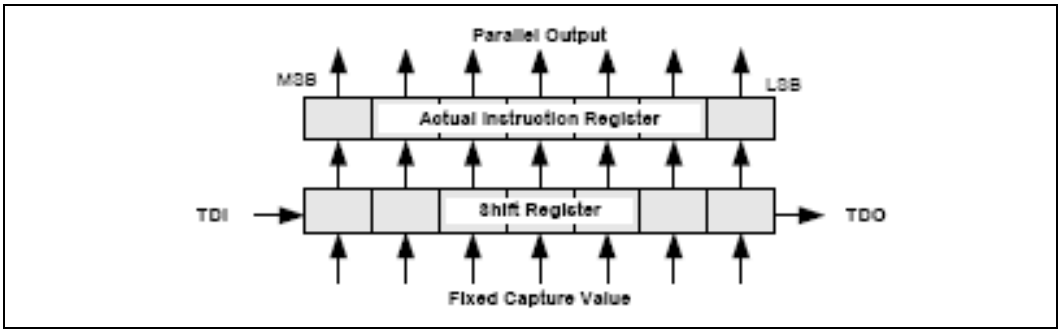


Figure 10-4 shows the operation of the instruction register during the Capture-IR, Shift-IR and Update-IR states. Shaded areas indicate the bits that are updated. In Capture-IR, the shift register portion of the instruction register is loaded in parallel with the fixed value "0000001". In Shift-IR, the shift register portion of the instruction register forms a serial data path between TDI and TDO. In Update-IR, the shift register contents are latched in parallel into the actual instruction register. Note that the only time the outputs of the actual instruction register change is during Update-IR. Therefore, a new instruction shifted into the TAP does not take effect until the Update-IR state is visited.

Figure 10-4. TAP Instruction Register Operation

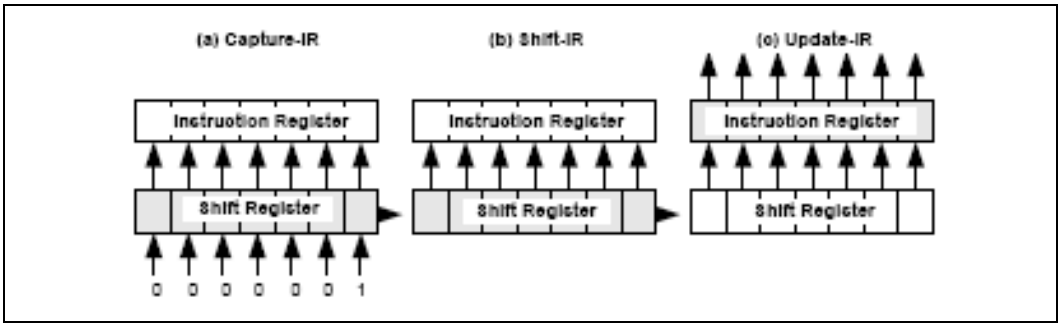
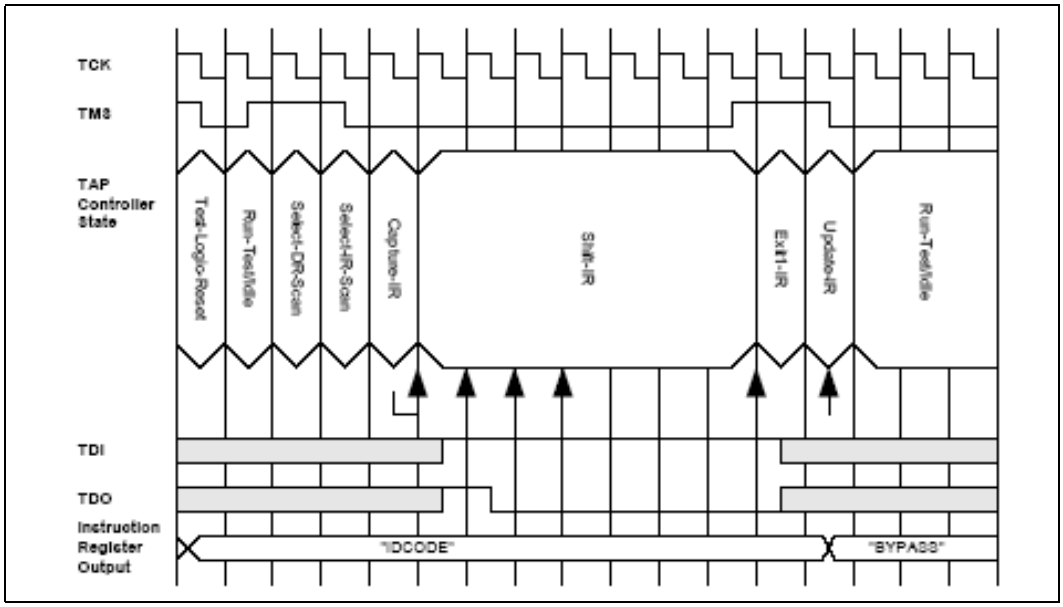


Figure 10-5 illustrates the timing when loading the BYPASS instruction (opcode 1111111b) into the TAP instruction register. Vertical arrows on the figure show the specific clock edges on which the Capture-IR, Shift-IR and Update-IR actions actually take place. Capture-IR (which preloads the instruction register with 0000001b) and Shift-IR operate on rising edges of TCK, and Update- IR (which updates the actual instruction register) takes place on the falling edge of TCK.

Figure 10-5. TAP Instruction Register Access



10.1.6 Accessing the Data Registers

The test data registers in the Intel® 7300 Chipset components are architected in the same way as the instruction register, with components (i.e., either the "capture" or "update" functionality) removed from the basic structure as needed. Data registers are accessed just as the instruction register is, only using the "select-DR-scan" branch of the TAP finite state machine in Table 10-2. A specific data register is selected for access by each TAP instruction. Note that the only controller states in which data register contents actually change are Capture-DR, Shift-DR, Update-DR and Run-Test/ Idle. For each of the TAP instructions described below, therefore, it is noted what operation (if any) occurs in the selected data register in each of these four states.

10.1.7 Public TAP Instructions

Table 10-3 contains descriptions of the encoding and operation of the public TAP instructions. There are four 1149.1-defined instructions implemented in the Intel® 7300 Chipset devices. These instructions select from among three different TAP data registers – the boundary scan, device ID, and bypass registers. The public instructions can be executed with only the standard connection of the JTAG port pins. This means the only clock required will be TCK. Full details of the operation of these instructions can be found in the 1149.1 standard. The opcodes are 1149.1-compliant, and are consistent with the Intel-standard encodings. A brief description of each instruction follows. For more thorough descriptions refer to the IEEE 1149.1 specification.

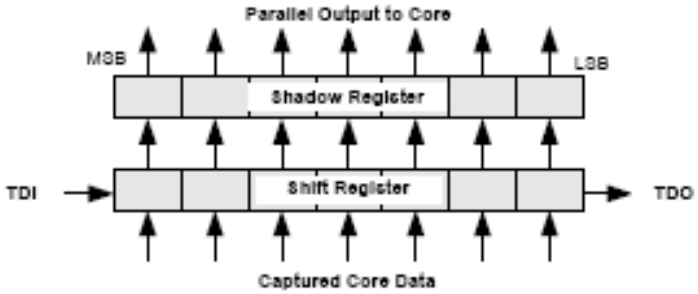
Table 10-3. Public TAP Instructions

Instruction	Encoding	Data Register Selected	Description
BYPASS	11111111	Boundary Scan	The BYPASS command selects the Bypass register, a single bit register connected between the TDI and TDO pins. This allows more rapid movement of test data to and from other components in the system.
EXTEST	00000000	Boundary Scan	The EXTEST instruction allows circuitry or wiring external to the devices to be tested. Boundary Scan register cells at outputs are used to apply stimulus, while Boundary Scan register cells at inputs are used to capture data.
SAMPLE/ PRELOAD	00000001	Boundary Scan	The SAMPLE/PRELOAD instruction is used to allow scanning of the Boundary Scan register without causing interference to the normal operation of the device. Two functions can be performed by use of the SAMPLE/PRELOAD instruction: 1. SAMPLE allows a snapshot of the data flowing into and out of the device to be taken without affecting the normal operation of the device. 2. PRELOAD allows an initial pattern to be placed into the Boundary Scan register cells. This allows initial known data to be present prior to the selection of another Boundary Scan test operation.
IDCODE	0000010	IDCODE	The IDCODE instruction is forced into the parallel output latches of the instruction register during the Test-Logic-Tap state. This allows the Device Identification register to be selected by manipulation of the broadcast TMS and TCK signals for testing purposes, as well as by a conventional instruction register scan operation.
CLAMP	0000100	Bypass	This allows static "guarding" values to be set into components that are not specifically being tested while maintaining the Bypass register as the serial path through the device.
HIGHZ	0001000	Bypass	The HIGHZ instruction is used to force all outputs of the device (except TDO) into a high impedance state. This instruction shall select the Bypass register to be connected between TDI and TDO in the Shift-DR controller state.

10.1.8 Public Data Instructions

This section describes the data registers that are accessed by the public and private instructions. Data shifts into all chains through the MSB of the data register as shown in Figure 10-6 which is the same as the instruction register.

Figure 10-6. TAP Data Register



10.1.9 Public Data Register Control

Table 10-4 define the actions that occur in the selected data register in controller states that can alter data register contents. If a TAP state does not affect the selected data register, then the corresponding table entry will be blank. Not all data registers have a parallel output latch. All data registers have a parallel input latch. Several table entries are still under investigation.

Table 10-4. Actions of Public TAP Instructions During Various TAP States

Instruction	Capture-DR	Shift-DR	Update-DR
Bypass	Reset Bypass Register	Shift Bypass register	
HighZ	Reset Bypass Register	Shift Bypass register	
IDcode	Load device ID into register	Shift ID register	
Extest	Load input pin values into Boundary Scan shift register	Shift Boundary Scan shift register	Load Boundary Scan shift register into Boundary Scan register; drive pins accordingly
Sample/Preload	Load pin values into Boundary Scan shift register	Shift Boundary Scan shift register	Load Boundary Scan shift register into Boundary Scan register

10.1.10 Bypass Register

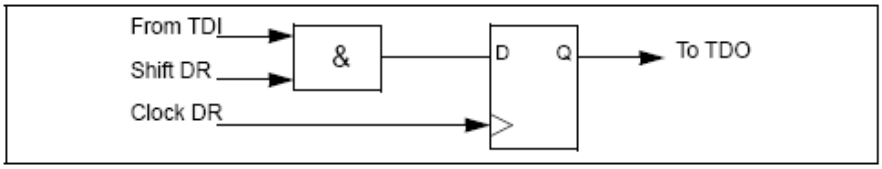
This register provides the minimal length path between TDI and TDO. It is loaded with a logical 0 during the Capture-DR state. The Bypass Register is a single bit register and is used to provide a minimum-length serial path through the device. This allows more rapid movement of test data to and from other components in the system. When in Bypass Mode, the operation of the test logic shall have no effect on the operation of the devices normal logic. Refer to Figure 10-7 for an implementation example.

10.1.10.1 Bypass Register Definition

JTAG encode: 1111111

Bit	Attr	Default	Description
1	RW	0	Bypass: a one bit register used to bypass the chip for board testing.

Figure 10-7. Bypass Register Implementation



10.1.11 Device ID Register

This register contains the device identification code in the format shown in Table 10-5. Three fields are predefined as the version number (stepping number), the manufacturer's identification code, and a logical 1 field. The component identification field is sub-divided into 3 fields. The Product Segment field identifies if the component is intended for CPU, laptop, desktop, server, etc. Product Type further defines the

component within a segment by stating it to be a CPU, memory, chipset, etc. The last field is a sequential component number assignment. This value will be maintained as sequential as possible depending on when each component's request was satisfied in the corporate database.

Table 10-5. Intel® 7300 Chipset Chipset Device ID Codes

Device	Version	Component Identification Fields			Manufacturing ID	"1"	Entire Code (hex)
		Product Segment	Product Type	Component Number			
	4	6	5	5	11	1	32
MCH	0001	101000	01000	01011	00000001001	1	0x1A10B013

10.1.11.1 Device ID Register

JTAG encode: 0000010			
Bit	**Attr**	**Default**	**Description**
31:28	R	0000	Version: This number changes for each stepping including metal "dash" steppings. The most significant 2 bits are the stepping number: 00 A-step; 01 B-step, 10 C-step, and 11 D-step. The least significant 2 bits is the revision within a stepping.
27:22	R	101000	Product Segment: Number assigned that determines the market segment into which this component belongs. Since this format is new, the value for chipset is shown with others as an example. CPU (Legacy):000 000 CPU:100 000 Desktop:010 000 Laptop:001 000 Server (legacy):000 100 Server: 101 000 etc.
21:17	R	01000	Product Type: Number assigned to further define the component within the market segment. Since this format is new, the value for chipset is shown with others as an example. Test:00 000 CPU:10 010 Memory:00 100 Modem:00 101 Chipset:01 000 etc.
16:12	R	Listed in next column	Component Number: Sequential listing based on request to database. MCH:01011b
11:1	R	00000001001	Manufacturing ID: This number is assigned to Intel.
0	R	1	'1'

10.1.12 Boundary Scan Register

The following requirements apply to those interfaces that continue to support boundary scan (bscan) or the miscellaneous I/O signals.

- Each signal or clock pin (with the exception of the TAP specific pins TCK, TDI, TDO, TMS, & TRST#) will have an associated Boundary-Scan Register Cell. Differential Driver or Receiver Pin Pairs that cannot be used independently shall be considered a single pin (i.e. one Boundary-Scan Register Cell after the differential receiver).
- Internal Signals which control the direction of I/O pins shall also have associated Boundary- Scan Register Cells.
- Each Output pin (with the exception of TDO) shall be able to be driven to a tristate condition for HIGHZ test.

10.2 Intel® IBIST Support

Intel® Interconnect-Built-In-Self-Test (Intel® IBIST) ia an advanced method of detecting and diagnosing interconnect circuit faults. IBIST covers both DC and AC faults, and operates at the full bandwidth of the interface being tested. As bus speeds increase beyond 500 MHz additional testability features such as Boundary Scan (IEEE 1149.1) (a virtual bus interconnect test methodology) will become useless due to frequency and timing issues. Further, the board/system fault spectrum associated with high-speed system buses has expanded beyond simple opens/shorts due to limited tolerance to transmission-line loss, impedance discontinuities, return path discontinuities, ISI, crosstalk, power supply collapse, nonlinear driver effects, non optimum VOH, VOL levels, non-ideal termination, and non-centered VREF.

IBIST is an on die technology that tests the I/O buffer register, driver, package interconnect, circuit board interconnect, receiver package interconnect, receiver and receiver register. IBIST uses a built in pattern generator that operates off of internal or programmed patterns. The on die implementation of IBIST is customized to the interface being tested. Tailoring IBIST to the interface being tested allows optimization of IBIST to test the fault spectrum of the interface. This customization includes the programming of built-in bit pattern test vectors for the interface under test.

The on-die IBIST architecture is designed for use in system level bus testing to assist platform electrical verification and platform HVM. The goals of the IBIST technology are:

- Increase platform level test coverage on IBIST enabled high performance buses,
- Reduce platform test time
- Facilitate platform level test access and availability
- Enable advancements in platform management.

IBIST has significant time to market and reliability benefits from a validation perspective, as it provides:

- Reduces test pattern development time
- Reduces complex system configuration time, by enabling concurrent traffic without the bus protocol and driver limitations
- Reduces debug time, more deterministic way to isolate faults

The IBIST architecture is designed to detect complex high frequency faults, it also provides simple opens/shorts detection.

The IBIST methodology is a comprehensive bus level testing solution, which covers:

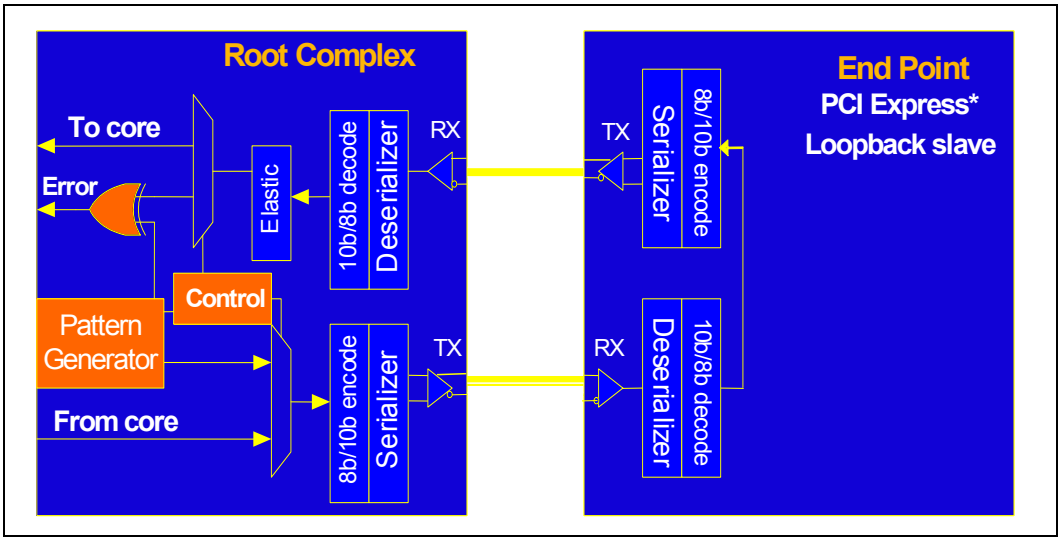
- Component-to-component
- Component-to-adapter
- Examination of entire high speed interconnect bus topologies (from I/O drivers to PCB traces characteristics to receivers and power delivery sub-system)

The MCH supports IBIST on all PCI Express and Fully Buffered DIMM interfaces. The MCH IBIST is controlled by IBIST registers.

10.2.1 PCI Express IBIST

The IBIST architecture utilizes the same component interconnect and timing paths as normal operation, output and input latches are inside the IBIST loop. Precise control of patterns is now possible due to the fact that IBIST operation is independent of system bus protocol. The IBIST architecture is located in the physical layer of the PCI Express Architecture. The Link and Transaction layers are isolated from the physical layer during IBIST testing.

Figure 10-8. PCI Express IBIST Architecture

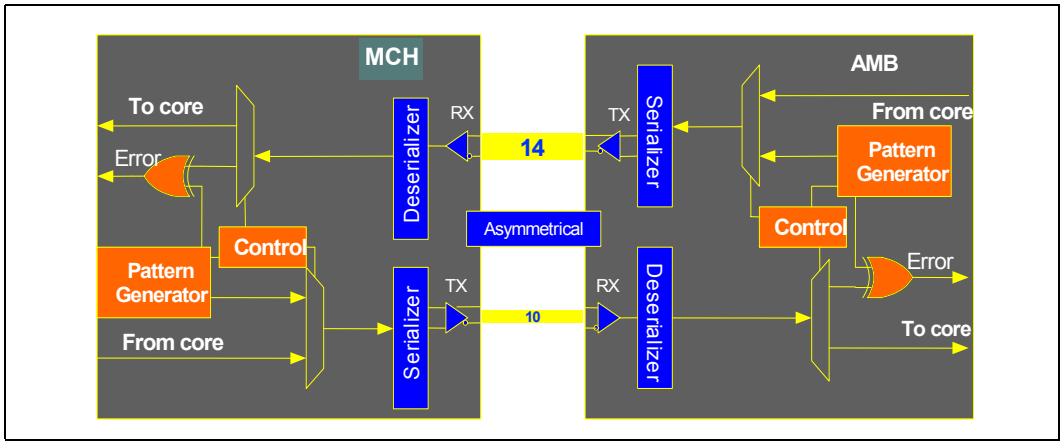


10.2.2 FB-DIMM IBIST

The FB-DIMM lane is a non-symmetrical link, MCH is a master in the southbound link and AMB is the master on the northbound link. The FB-DIMM uses IBIST to verify the integrity of the link between the host controller and the AMB. The host controller includes one IBIST engine (generator and checker), and the AMB includes two IBIST engines, one for northbound and one for southbound links. FB-DIMM IBIST supports both loopback and master to master topology.

- The MCH sends test pattern through the southbound FB-DIMM link, the AMB receives the test pattern and compares it against the expected pattern.
- AMB sends test pattern through the northbound FB-DIMM link, the MCH receives the test pattern and compares it against the expected pattern.

Figure 10-9. FB-DIMM IBIST Architecture



Please refer to the *Intel® IBIST (Interconnect Built-In Self Test) Functional Specification* for detail information on IBIST.

§