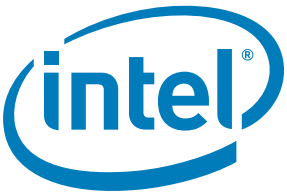


Intel® 5400 Chipset Memory Controller Hub (MCH)

Datasheet

November 2007

Document Number: 318610, Revision 001

Contents

Figures

Tables

Revision History

Revision Number	Description	Date
-001	Initial public release	November 2007

§

1 Introduction

The Intel® 5400 chipset is designed for systems based on the Intel processors Dual-Core Intel® Xeon® Processor 5100 series, Quad-Core Intel® Xeon® Processor 5300 series, Quad-Core Intel® Xeon® Processor 5400 series, and Dual-Core Intel® Xeon® Processor 5200 series, and supports 1066 MTS, 1333 MTS and 1600 MTS FSB interfaces. The Intel® 5400 chipset contains two main components: Memory Controller Hub (MCH) for the host bridge and the I/O controller hub for the I/O subsystem. The Intel® 5400 chipset uses the Intel® 631xESB/632xESB I/O Controller Hub.

The Intel® 5400 chipset is implemented in a 90 nm P1263 silicon process and packaged in a 1520 pin FCBGA package with pins on 1.067 mm (42 mil) centers. The overall package dimensions are 42.5 mm by 42.5 mm.

The Intel® 5400 chipset platform supports the Dual-Core Intel Xeon Processor 5100 series, Quad-Core Intel Xeon Processor 5300 series, Quad-Core Intel Xeon Processor 5400 series and Dual-Core Intel Xeon Processor 5200 series (266/333 MHz) in a 771-land, FC-LGA (Flip Chip Land Grid Array) package. This package uses the matching LGA771 socket. The surface mount, LGA771 socket supports Direct Socket Loading (DSL). The Dual-Core Intel Xeon Processor 5100 series, Quad-Core Intel Xeon Processor 5300 series, Quad-Core Intel Xeon Processor 5400 series, and Dual-Core Intel Xeon Processor 5200 series return a processor signature of 0x6Fyh, 0x06Fyh, 0x06Fyh and extended CPUID 0x1067yh, respectively, where y is the stepping number, when the CPUID instruction is executed with EAX=1.

Note: Unless otherwise specified, the term processor in this document refers jointly to the Dual-Core Intel Xeon Processor 5100 series, Quad-Core Intel Xeon Processor 5300 series, Quad-Core Intel Xeon Processor 5400 series, and Dual-Core Intel Xeon Processor 5200 series.

1.1 Terminology

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

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
ASF	Alert Standard Format
Asserted	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 is 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 is 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 allows 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.

Terminology	Description
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 Intel® 5400 MCH that acts as a temporary repository for system data in flight between the various ports: FSB's, FBD's, ESI, and PCI Express*.
Cfg, Config	Abbreviation for "Configuration".
Channel	In the MCH a FBD DRAM channel is the set of signals that connects to one set of FBD DIMMs. The MCH has up to four DRAM 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.
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 consistently 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.
CB	Intel® QuickData Technology Device
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 a 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.
DIMM Rank	That set of SDRAMs on one branch which provides the data packet
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

Terminology	Description
EHCI	Enhanced Host Controller Interface, specification for Universal Serial Bus
"ESB2"	Intel® 631xESB/632xESB I/O controller Hub (Enterprise South Bridge 2)
ESI	Enterprise South Bridge Interface
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.
GART	Graphics Aperture Remap Table. GART is a table in memory containing the page remap information used during AGP aperture address translations.
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.
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)

Terminology	Description
MCH	The Memory Controller Hub component that contains the processor interface, DRAM controller, PCI Express interface, and AGP interface. It communicates with the Intel 631xESB/632xESB I/O Controller Hub (formerly ESB2) over a proprietary interconnect called the Enterprise Southbridge 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
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
PCI Express 1.1 Compliant	Refers to compliance to the *PCI-Express Base Specification*, Revision 1.1
Peer to Peer	Transactions that occur between two devices below the PCI Express or ESI ports.
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 ESI. 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.

Terminology	Description
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).
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** is equivalent to **RAID-1**.
RASUM	Reliability, Availability, Serviceability, Usability, and Manageability, which are all important characteristics of workstations
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.
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.

Terminology	Description
SDR	Single Data Rate SDRAM.
SDRAM	Synchronous Dynamic Random Access Memory.
SEC/DED	Single-bit Error Correct / Double-symbol Error Detect
Intel® 5400 MCH	Intel® 5400 North Bridge Also referred to in general as MCH North Bridge or simply MCH.
Secondary PCI	The physical PCI interface that is a subset of the AGP bus driven directly by the MCH. It supports a subset of 32-bit, 66 MHz PCI 2.0 compliant components, but only at 1.5 V (not 3.3 V or 5 V).
Serial Presence Detect (SPD)	A two-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.
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® 5400 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)
Sticky Bits	Register bits, whose value remains unchanged when system is RESET
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.
System Bus	Processor-to-MCH interface. The system bus in this document refers to operation at 333/667/1333 MHz (Bus Clock/Address/Data). The system bus is not compatible with the P6 system bus.
Target	A device that responds to bus Transactions. The agent receiving a request packet is referred to as the Target for that Transaction.
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.
TPM	Trusted Platform Module
UHCI	Universal Host Controller Interface (UHCI) controllers that support USB full-speed and low-speed signaling
Upstream	See Terminology entry of "Inbound (IB)/Outbound (OB), AKA Upstream/DownStream, Northbound/Southbound, Upbound/Downbound"

1.2 Related Documents and Materials

Table 1-1. Related Documents

Document	Document Number/ Location
PCI-Express Base Specification, Revision 2.0	http://www.pcisig.com/specifications/pciexpress/
Low Pin Count Interface Specification, Revision 1.1 (LPC)	http://developer.intel.com/design/chipsets/industry/lpc.htm
System Management Bus Specification, Version 2.0 (SMBus)	http://www.smbus.org/specs/
PCI Local Bus Specification, Revision 2.3	http://www.pcisig.com/specifications/conventional/
PCI Power Management Specification, Revision 1.1	http://www.pcisig.com/specifications/conventional/pci_bus_power_management_interface
Advanced Configuration and Power Interface, Version 2.0 (ACPI)	http://www.acpi.info/
Enhanced Host Controller Interface Specification for Universal Serial Bus, Revision 0.96 (EHCI)	http://developer.intel.com/technology/usb/ehcispec.htm

1.3 Intel® 5400 Chipset Overview

In a Intel® 5400 chipset-based platform, the MCH provides two FSB processor interfaces, four fully buffered DIMM memory channels, nine x4 PCI Express bus interfaces configurable to form x8 or x16 ports, an Enterprise South Bridge Interface (ESI), and six SM Bus interfaces for system management, and DIMM Serial Presence Detect (SPD). The chipset device has a 24 MB Snoop Filter integrated into the FSB interface as well an Intel® I/O Acceleration Technology (Intel® I/OAT) compatible Intel® QuickData Technology Device. Figure 1-1, "Intel® 5400 Chipset System Block Diagram" shows an example block diagram of a Intel® 5400 chipset-based platform.

The Intel® 5400 chipset is designed for use in workstation and high-performance computing (HPC) systems based on the processors Dual-Core Intel Xeon Processor 5100 series, Quad-Core Intel Xeon Processor 5300 series, Quad-Core Intel® Xeon® Processor 5400 Series and Dual-Core Intel® Xeon® Processor 5200 Series. The Intel® 5400 chipset supports two processors on dual independent point to point system buses operating at 266 MHz (1066 MTS), 333 MHz (1333 MTS) or 400 MHz (1600 MTS); the peak bandwidth of the two processor busses is respectively, 17 GB/s Intel® 5400 chipset and 21 GB/s for outbound and 8 GB/s and 10 GB/s for inbound accesses. The MCH supports 38-bit addressability with a total 128 GB of physical memory.

The Dual-Core Intel Xeon Processor 5100 series (65 nm process) has a 2 x 4 MB shared cache and a 333 MHz (1333 MTS) system bus. The Quad-Core Intel® Xeon® Processor 5300 series (65 nm process) and Quad-Core Intel® Xeon® Processor 5400 Series (45 nm process) have respectively 2 x 4 MB and 2 x 6 MB shared L2 caches and a 333 MHz (1333 MTS) system bus. The Dual-Core Intel® Xeon® Processor 5200 Series (45 nm process) has a 6 MB shared cache and a 333 MHz (1333 MTS) system bus.

The MCH provides four channels of Fully Buffered DIMM (FBD) memory. Each channel can support up to 4 Dual Ranked FBD DDR2 DIMMs. The MCH can support up to 16 DIMMs or a maximum memory size of 128 GB physical memory. The read bandwidth for each FB-DIMM channel is 4.25 GB/s for DDR2 533 FB-DIMM memory which gives a total read bandwidth of 17 GB/s for four FB-DIMM channels. Thus this provides 8.5 GB/s of write memory bandwidth for four FB-DIMM channels. The read bandwidth for each FB-DIMM channel is 5.325 GB/s for DDR2 667 FB-DIMM memory which gives a total read bandwidth of 21.3 GB/s for four FB-DIMM channels. Thus this provides

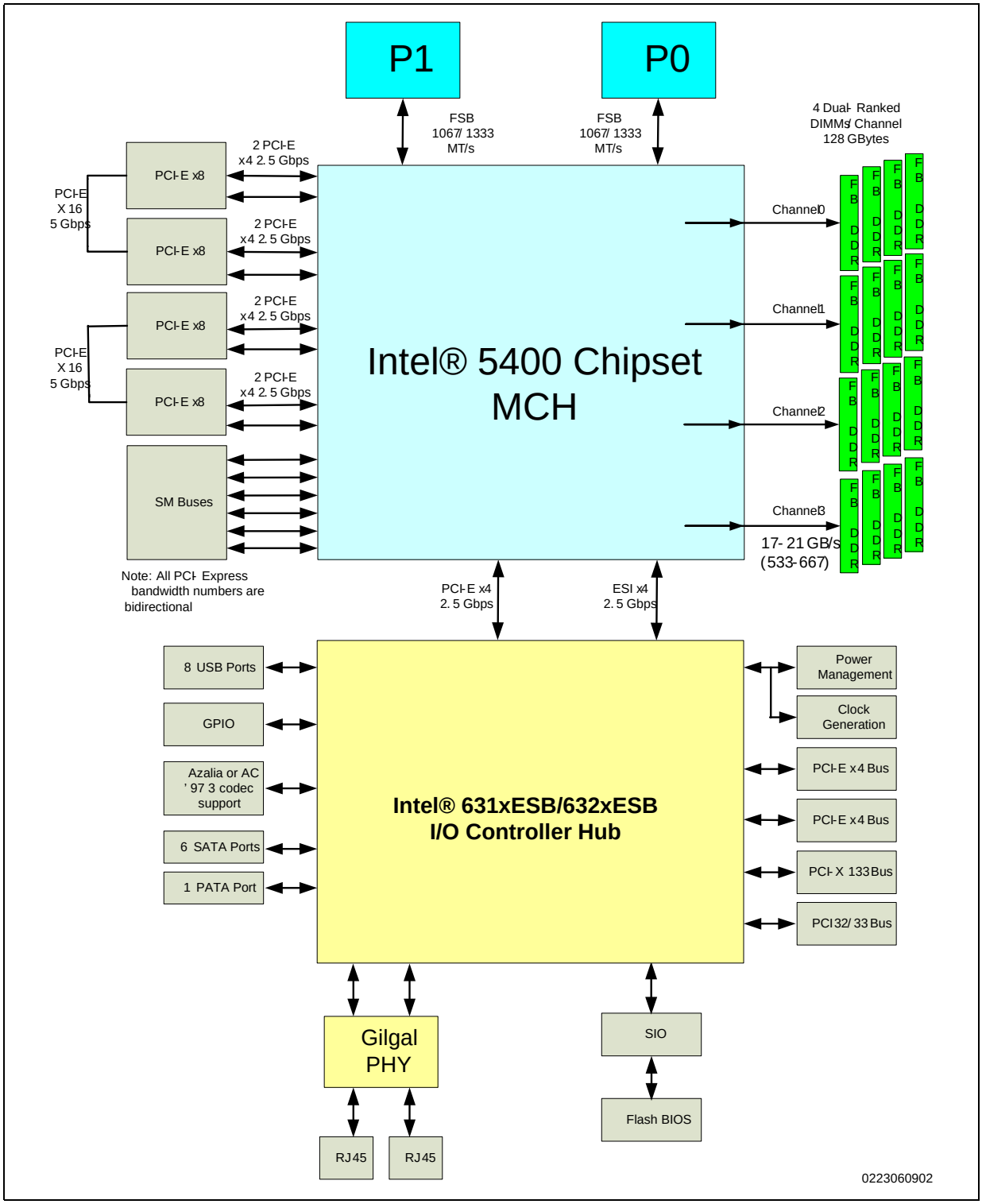
10.7 GB/s of write memory bandwidth for four FB-DIMM channels. The total bandwidth is based on read bandwidth therefore the total bandwidth is 17 GB/s for 533 and 21.3 GB/s for 667.

The MCH offers nine x4 PCI Express ports compliant to *PCI Express Base Specification 2.0*. Configured appropriately, the ports can be combined to form x8 or x16 ports. The ports have the added ability to operate up to speeds of 5 Gbps in x16 configuration for Graphics and HPC applications.

The Intel 631xESB/632xESB I/O Controller Hub integrates an Ultra ATA 100 controller, six Serial ATA host controller ports, one EHCI host controller, and four UHCI host controllers supporting eight external USB 2.0 ports, LPC interface controller, and a flash BIOS interface controller. Additionally the Intel 631xESB/632xESB I/O Controller Hub contains a PCI interface controller, Azalia / AC'97 digital controller, integrated LAN controller, an ASF controller and a ESI for communication with the MCH. 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. The Intel 631xESB/632xESB I/O Controller Hub elevates Serial ATA storage performance to the next level with Intel® RAID.

The ACPI compliant Intel 631xESB/632xESB I/O Controller Hub platform can support the Full-on, Stop Grant, Suspend to RAM, Suspend to Disk, and Soft-Off power management states. Through the use of the integrated LAN functions, the Intel 631xESB/632xESB I/O Controller Hub also supports Alert Standard Format for remote management.

Figure 1-1. Intel® 5400 Chipset System Block Diagram



§

2 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" at 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 will be appended to both 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

s/t/s Sustained Tri-state. This pin is driven to its inactive state prior to tri-stating.

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

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 1.1 V to 1.2 V.

CMOS CMOS buffers

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 "double-pumped").

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 2-1. 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.

Table 2-2 lists the reference terminology used for signal types.

Table 2-2. Buffer Signal Types

Buffer Direction	Description
I	Input signal
O	Output signal
A	Analog
I/O	Bidirectional (input/output) signal

2.1 Processor Front Side Bus Signals

2.1.1 Processor Front Side Bus 0

Signal Name	Type	Description
FSB0A_N[37:3]	I/O	**Processor 0 Address Bus:** FSB0A_N[37:3] connect to the processor address bus. During processor cycles, FSB0A_N[37:3] are inputs. The MCH drives FSB0A_N[37:3] during snoop cycles on behalf of ESI and Secondary PCI initiators. FSB0A_N[37:3] are transferred at 2X rate. Note that the address is inverted on the processor bus. The MCH drives the FSB0A_N[7] signal, which is then sampled by the processor and the MCH 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.
FSB0ADS_N	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]	I/O	**Processor 0 Address Strobe:** FSB0ADSTB_N[1:0] are source synchronous strobes used to transfer FSB0A_N[37:3] and FSB0REQ_N[4:0] at the 2X transfer rate. **StrobeAddress Bits** FSB0ADSTB0_NFSB0A_N[16:3], FSB0REQ_N[4:0], NFSB1_N[37:36] FSB0ADSTB1_NFSB0A_N[35:17]
FSB0AP_N[1:0]	I/O	**Processor 0 Address Parity:** FSB0AP_N[1:0] provide parity protection on the address bus
FSB0BINIT_N	I/O	**Processor 0 Bus Initialization:** This signal causes a reset of the bus state machines.

Signal Name	Type	Description
FSB0BNR_N	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_N	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	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.
FSB0BREQ_N[1:0]	I/O	**Processor 0 Bus Requests**: The MCH pulls the FSB0BREQ0_N 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]	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 FSB**0DBI_N**[3:0] signals.
FSB0DBI_N[3:0]	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_N3 FSB0D_N[63:48] FSB0DBI_N2 FSB0D_N[47:32] FSB0DBI_N1 FSB0D_N[31:16] FSB0DBI_N0 FSB0D_N[15:0]
FSB0DBSY_N	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	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_N[3:0]	I/O	**Processor 0 Data Bus Parity:** FSB0DP_N[3:0] provide parity protection on the data bus.
FSB0DRDY_N	I/O	**Processor 0 Data Ready:** This signal is asserted for each cycle that data is transferred.
FSB0STBP_N[3:0] FSB0STBN_N[3:0]	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. **StrobeData Bits** FSB0STBP3_N, FSB0STBN3_N FSB0D_N[63:48], FSB0DBI3_N FSB0STBP2_N, FSB0STBN2_N FSB0D_N[47:32], FSB0DBI2_N FSB0STBP1_N, FSB0STBN1_N FSB0D_N[31:16], FSB0DBI1_N FSB0STBP0_N, FSB0STBN0_N FSB0D_N[15:0], FSB0DBI0_N
FSB0HIT_N	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	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.

Signal Name	Type	Description
FSB0LOCK_N	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 FSB0BPRI_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	I/O	**Processor 0 Machine Check Error:** Machine check error
FSB0REQ_N[4:0]	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	O	**Processor 0 Reset:** The FSB0RESET_N pin is an output from the MCH. The MCH asserts FSB0RESET_N while RESETI_N (PCIRST_N from Intel 631xESB/632xESB I/O Controller Hub) is asserted and for approximately 1ms after RESETI_N is deasserted. The FSB0RESET_N allows the processors to begin execution in a known state.
FSB0RS_N[2:0]	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	O	**Processor 0 Response Status Parity:**
FSB0TRDY_N	O	**Processor Bus 0 Target Ready:** This signal indicates that the target of the processor transaction is able to enter the data transfer phase.
FSB0FSBVREF	Analog	**Processor 0 Voltage Reference:** Processor 0 voltage reference varies between 0.733 V and 0.800 V depending on the installed CPU. For Dual-Core Intel Xeon Processor 5100 series and Quad-Core Intel Xeon Processor 5300 series the voltage is 0.800 V and for Quad-Core Intel Xeon Processor 5400 series and Dual-Core Intel Xeon Processor 5200 series the voltage is 0.733 V.

2.1.2 Processor Front Side Bus 1

Signal Name	Type	Description
FSB1A_N[37:3]	I/O	**Processor 1 Address Bus:** FSB1A_N[37:3] connect to the processor address bus. During processor cycles, FSB1A_N[37:3] are inputs. The MCH drives FSB1A_N[37:3] during snoop cycles on behalf of ESI and Secondary PCI initiators. FSB1A_N[37:3] are transferred at 2X rate. Note that the address is inverted on the processor bus. ***Note:*** The MCH drives the FSB1A_N[7] signal, which is then sampled by the processor and the MCH on the active-to-inactive transition of FSB1RESET_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 FSB1CLKs.
FSB1ADS_N	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]	I/O	**Processor 1 Address Strobe:** FSB1ADSTB_N[1:0] are source synchronous strobes used to transfer FSB1A_N[37:3] and FSB1REQ_N[4:0] at the 2X transfer rate. **Strobe** — **Address Bits** FSB1ADSTB0_N — FSB1A_N[16:3], FSB1REQ_N[4:0] FSB1ADSTB1_N — FSB1A_N[37:17]
FSB1AP_N[1:0]	I/O	**Processor 1 Address Parity:** FSB0AP_N[1:0] provide parity protection on the address bus
FSB1BINIT_N	I/O	**Processor 1 Bus Initialization:** This signal causes a reset of the bus state machines.
FSB1BNR_N	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]	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.
FSB1BPRI_N	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]	I/O	**Processor 1 Bus Requests:** The MCH pulls the FSB1BREQ1_N & FSB1BREQ0_N signals low during RESETI_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]	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]	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** FSB1DBI3_N — FSB1D_N[63:48] FSB1DBI2_N — FSB1D_N[47:32] FSB1DBI1_N — FSB1D_N[31:16] FSB1DBI0_N — FSB1D_N[15:0]
FSB1DBSY_N	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	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.

Signal Name	Type	Description
FSB1DP_N[3:0]	I/O	**Processor 1 Data Bus Parity:** FSB1DP_N[3:0] provide parity protection on the data bus.
FSB1DRDY_N	I/O	**Processor 1 Data Ready:** This signal is asserted for each cycle that data is transferred.
FSB1STBP_N[3:0] FSB1STBN_N[3:0]	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. **StrobeData Bits** FSB1STBP3_N, FSB1STBN3_N FSB1D_N[63:48], FSB1DBI3_N FSB1STBP2_N, FSB1STBN2_N FSB1D_N[47:32], FSB1DBI2_N FSB1STBP1_N, FSB1STBN1_N FSB1D_N[31:16], FSB1DBI1_N FSB1STBP0_N, FSB1STBN0_N FSB1D_N[15:0], FSB1DBI0_N
FSB1HIT_N	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	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	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	I/O	**Processor 1 Machine Check Error:** Machine check error
FSB1REQ_N[4:0]	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	O	**Processor 1 Reset:** The FSB1RESET_N pin is an output from the MCH. The MCH asserts FSB1RESET_N while RESETI_N (PCIRST_N from Intel 631xESB/632xESB I/O Controller Hub) is asserted and for approximately 1ms after RESETI_N is deasserted. The FSB1RESET_N allows the processors to begin execution in a known state.
FSB1RS_N[2:0]	O	**Processor 1 Response Status Signals:** These signals indicates 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
FSB1RSP_N	O	Processor 1 Response Status Parity:
FSB1TRDY_N	O	**Processor Bus 1 Target Ready:** This signal indicates that the target of the processor transaction is able to enter the data transfer phase.
FSB1FSBVREF	Analog	**Processor 1 Voltage Reference:** Processor 1 voltage reference varies between 0.733 V and 0.800 V depending on the installed CPU. For Dual-Core Intel Xeon Processor 5100 series and Quad-Core Intel Xeon Processor 5300 series the voltage is 0.800 V and for Quad-Core Intel Xeon Processor 5400 series and Dual-Core Intel Xeon Processor 5200 series the voltage is 0.733 V.

2.2 Fully Buffered DIMM Memory Channels

The following reference and compensation signals are common to all FB-DIMM channels.

Signal Name	Type	Description
FB-DIMMICOMP	Analog	**FB-DIMM Impedance Compensation**

2.2.1 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
FB-DIMM01CLKN	Analog	**FB-DIMM Clock Negative:** Core Clock Negative Phase
FB-DIMM01CLKP	Analog	**FB-DIMM Clock Positive:** Core Clock Positive Phase

2.2.1.1 FB-DIMM Channel 0

Signal Name	Type	Description
FB-DIMM0NBIN[13:0]	I	**FB-DIMM Channel 0 Northbound Input Data Negative Phase: NOTE: FB-DIMM0NBIN[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FB-DIMM0NBIP[13:0]	I	**FB-DIMM Channel 0 Northbound Input Data Positive Phase: NOTE: FB-DIMM0NBIP[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FB-DIMM0SBON[9:0]	O	**FB-DIMM Channel 0 Southbound Output Negative Phase:**
FB-DIMM0SBOP[9:0]	O	**FB-DIMM Channel 0 Southbound Output Positive Phase:**

2.2.1.2 FB-DIMM Channel 1

Signal Name	Type	Description
FB-DIMM1NBIN[13:0]	I	**FB-DIMM Channel 1 Northbound Input Data Negative Phase: NOTE: FB-DIMM1NBIN[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FB-DIMM1NBIP[13:0]	I	**FB-DIMM Channel 1 Northbound Input Data Positive Phase: NOTE: FB-DIMM1NBIP[13] is not an active signal carrying signal but must be connected to properly terminate the FB DIMM component.**
FB-DIMM1SBON[9:0]	O	**FB-DIMM Channel 1 Southbound Output Negative Phase:**
FB-DIMM1SBOP[9:0]	O	**FB-DIMM Channel 1 Southbound Output Positive Phase:**

2.2.2 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
FB-DIMM23CLKN	Analog	**FB-DIMM Clock Negative:** Core Clock Negative Phase
FB-DIMM23CLKP	Analog	**FB-DIMM Clock Positive:** Core Clock Positive Phase
VCCAFB-DIMM23PLL	Power/ Other	VCC: Analog Voltage for the PLL
VCCAFB-DIMM23PLL18	Power/ Other	VCC: 1.8V supply for the PLLs. See also FB-DIMM Branch 0.

2.2.2.1 FB-DIMM Channel 2

Signal Name	Type	Description
FB-DIMM2NBIN[13:0]	I	**FB-DIMM Channel 2 Northbound Input Data Negative Phase**
FB-DIMM2NBIP[13:0]	I	**FB-DIMM Channel 2 Northbound Input Data Positive Phase**
FB-DIMM2SBON[9:0]	O	**FB-DIMM Channel 2 Southbound Output Negative Phase:**
FB-DIMM2SBOP[9:0]	O	**FB-DIMM Channel 2 Southbound Output Positive Phase:**

2.2.2.2 FB-DIMM Channel 3

Signal Name	Type	Description
FB-DIMM3NBIN[13:0]	I	**FB-DIMM Channel 3 Northbound Input Data Negative Phase**
FB-DIMM3NBIP[13:0]	I	**FB-DIMM Channel 3 Northbound Input Data Positive Phase**
FB-DIMM3SBON[9:0]	O	**FB-DIMM Channel 3 Southbound Output Negative Phase**
FB-DIMM3SBOP[9:0]	O	**FB-DIMM Channel 3 Southbound Output Positive Phase**

2.3 PCI Express* Signal List

2.3.1 PCI Express Common Signals

Signal Name	Type	Description
PE0CLKN	Analog	**PCI Express Common Clock Negative Phase related to PE0PLL.**
PE0CLKP	Analog	**PCI Express Common Clock Positive Phase related to PE0PLL.**
PE0ICOMPI	Analog	**PCI Express Impedance Compensation related to PE0PLL.**
PE1CLKN	Analog	**PCI Express Common Clock Negative Phase related to PE1PLL.**
PE1CLKP	Analog	**PCI Express Common Clock Positive Phase related to PE1PLL.**
PE1ICOMPI	Analog	**PCI Express Impedance Compensation related to PE1PLL.**
VCCRPE125	Power/Other	Digital I/O Supply.
VCCAPE0PLL125	Power/Other	VCC: 1.25V Analog voltage for PLL0
VCCAPE0PLL18	Power/Other	VCC: 1.8V Analog voltage for PLL0
VSSAPE0PLL	Power/Other	VSS: Analog ground for PLL0
VCCAPE1PLL125	Power/Other	VCC: 1.25V Analog voltage for PLL1
VCCAPE1PLL18	Power/Other	VCC: 1.8V Analog voltage for PLL1
VSSAPE1PLL	Power/Other	VSS: Analog ground for PLL1
VCCABGPE033	Power/Other	VCC: Bandgap Voltage 0

Signal Name	Type	Description
VSSABGPE0	Power/Other	VSS: Bandgap Ground 0
VCCABGPE133	Power/Other	VCC: Bandgap Voltage 1
VSSABGPE1	Power/Other	VSS: Bandgap Ground 1

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

PCI Express port 0 is a x4 port dedicated to provide the ESI link between the Intel® 5400 MCH and the Intel 631xESB/632xESB I/O Controller Hub.

Signal Name	Type	Description Reference
ESIRP[3:0]	I	**ESI port Positive Phase Inbound:** (Receive) Signals
ESIRN[3:0]	I	**ESI port Negative Phase Inbound:** (Receive) Signals
ESITP[3:0]	O	**ESI port Positive Phase Outbound:** (Transmit) Signals
ESITN[3:0]	O	**ESI port Negative Phase Outbound:** (Transmit) Signals

2.3.3 PCI Express Port 9, Stand Alone x4 Port

PCI Express port 9 is a x4 port and can only be configured as a stand alone x4 port.

Signal Name	Type	Description
PE9RP[3:0]	I	**PCI Express Port 9 Positive Phase Inbound:** (Receive) Signals
PE9RN[3:0]	I	**PCI Express Port 9 Negative Phase Inbound:** (Receive) Signals
PE9TP[3:0]	O	**PCI Express Port 9 Positive Phase Outbound:** (Transmit) Signals
PE9TN[3:0]	O	**PCI Express Port 9 Negative Phase Outbound:** (Transmit) Signals

2.3.4 PCI Express Port 1

PCI Express port 1 is a x4 port. PCI Express port 1 can be configured as a stand alone x4 port, combined with PCI Express port 2 to form a single PCI Express x8 port or combined with PCI Express ports 2-4 to form a single PCI Express x16 port for Graphics applications

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

2.3.5 PCI Express Port 2

PCI Express port 2 is a x4 port. PCI Express port 2 can be configured as a stand alone x4 port, combined with PCI Express port 1 to form a single PCI Express x8 port or combined with PCI Express ports 1, 3 and 4 to form a single PCI Express x16 port for Graphics applications.

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

2.3.6 PCI Express Port 3

PCI Express port 3 is a x4 port. PCI Express port 3 can be configured as a stand alone x4 port, combined with PCI Express port 4 to form a single PCI Express x8 port or combined with PCI Express ports 1, 2 and 4 to form a single PCI Express x16 port for Graphics applications.

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

2.3.7 PCI Express Port 4

PCI Express port 4 is a x4 port. PCI Express port 4 can be configured as a stand alone x4 port, combined with PCI Express port 3 to form a single PCI Express x8 port or combined with PCI Express ports 1-3 to form a single PCI Express x16 port for Graphics applications.

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

2.3.8 PCI Express Port 5

PCI Express port 5 is a x4 port. PCI Express port 5 can be configured as a stand alone x4 port, combined with PCI Express port 6 to form a single PCI Express x8 port or combined with PCI Express ports 6-8 to form a single PCI Express x16 port for Graphics applications

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

2.3.9 PCI Express Port 6

PCI Express port 6 is a x4 port. PCI Express port 6 can be configured as a stand alone x4 port, combined with PCI Express port 5 to form a single PCI Express x8 port or combined with PCI Express ports 5, 7and 8 to form a single PCI Express x16 port for Graphics applications.

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

2.3.10 PCI Express Port 7

PCI Express port 7 is a x4 port. PCI Express port 7 can be configured as a stand alone x4 port, combined with PCI Express port 8 to form a single PCI Express x8 port or combined with PCI Express ports 5, 6 and 8 to form a single PCI Express x16 port for Graphics applications.

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

2.3.11 PCI Express Port 8

PCI Express port 8 is a x4 port. PCI Express port 8 can be configured as a stand alone x4 port, combined with PCI Express port 7 to form a single PCI Express x8 port or combined with PCI Express ports 5-7 to form a single PCI Express x16 port for Graphics applications.

Signal Name	Type	Description
PE8RP[3:0]	I	**PCI Express Port 8 Positive Phase Inbound:** (Receive) Signals
PE8RN[3:0]	I	**PCI Express Port 8 Negative Phase Inbound:** (Receive) Signals
PE8TP[3:0]	O	**PCI Express Port 8 Positive Phase Outbound:** (Transmit) Signals
PE8TN[3:0]	O	**PCI Express Port 8 Negative Phase Outbound:** (Transmit) Signals

2.3.12 PCI Express Graphics Ports

In the Intel® 5400 MCH PCI Express ports 1, 2, 3 and 4 can be combined as shown below to form a high performance x16 graphics ports.

Signal Name	Type	Description
PE1RP[3:0]	I	PCI Express Graphics Port First x4, Positive Phase Inbound (Receive) Signals:
PE1RN[3:0]	I	PCI Express Graphics Port First x4, Negative Phase Inbound (Receive) Signals:
PE1TP[3:0]	O	PCI Express Graphics Port First x4, Positive Phase Outbound (Transmit) Signals:
PE1TN[3:0]	O	PCI Express Graphics Port First x4, Negative Phase Outbound (Transmit) Signals:
PE2RP[3:0]	I	PCI Express Graphics Port Second x4, Positive Phase Inbound (Receive) Signals:
PE2RN[3:0]	I	PCI Express Graphics Port Second x4, Negative Phase Inbound (Receive) Signals:
PE2TP[3:0]	O	PCI Express Graphics Port Second x4, Positive Phase Outbound (Transmit) Signals:
PE2TN[3:0]	O	PCI Express Graphics Port Second x4, Negative Phase Outbound (Transmit) Signals:
PE3RP[3:0]	I	PCI Express Graphics Port Third x4, Positive Phase Inbound (Receive) Signals:
PE3RN[3:0]	I	PCI Express Graphics Port Third x4, Negative Phase Inbound (Receive) Signals:
PE3TP[3:0]	O	PCI Express Graphics Port Third x4, Positive Phase Outbound (Transmit) Signals:
PE3TN[3:0]	O	PCI Express Graphics Port Third x4, Negative Phase Outbound (Transmit) Signals:
PE4RP[3:0]	I	PCI Express Graphics Port Fourth x4, Positive Phase Inbound (Receive) Signals:
PE4RN[3:0]	I	PCI Express Graphics Port Fourth x4, Negative Phase Inbound (Receive) Signals:
PE4TP[3:0]	O	PCI Express Graphics Port Fourth x4, Positive Phase Outbound (Transmit) Signals:
PE4TN[3:0]	O	PCI Express Graphics Port Fourth x4, Negative Phase Outbound (Transmit) Signals:

In the Intel® 5400 MCH PCI Express ports 5, 6, 7 and 8 can be combined as shown below to form a high performance x16 graphics ports.

Signal Name	Type	Description
PE5RP[3:0]	I	PCI Express Graphics Port First x4, Positive Phase Inbound (Receive) Signals:
PE5RN[3:0]	I	PCI Express Graphics Port First x4, Negative Phase Inbound (Receive) Signals:
PE5TP[3:0]	O	PCI Express Graphics Port First x4, Positive Phase Outbound (Transmit) Signals:
PE5TN[3:0]	O	PCI Express Graphics Port First x4, Negative Phase Outbound (Transmit) Signals:
PE6RP[3:0]	I	PCI Express Graphics Port Second x4, Positive Phase Inbound (Receive) Signals:
PE6RN[3:0]	I	PCI Express Graphics Port Second x4, Negative Phase Inbound (Receive) Signals:
PE6TP[3:0]	O	PCI Express Graphics Port Second x4, Positive Phase Outbound (Transmit) Signals:
PE6TN[3:0]	O	PCI Express Graphics Port Second x4, Negative Phase Outbound (Transmit) Signals:
PE7RP[3:0]	I	PCI Express Graphics Port Third x4, Positive Phase Inbound (Receive) Signals:
PE7RN[3:0]	I	PCI Express Graphics Port Third x4, Negative Phase Inbound (Receive) Signals:
PE7TP[3:0]	O	PCI Express Graphics Port Third x4, Positive Phase Outbound (Transmit) Signals:
PE7TN[3:0]	O	PCI Express Graphics Port Third x4, Negative Phase Outbound (Transmit) Signals:
PE8RP[3:0]	I	PCI Express Graphics Port Fourth x4, Positive Phase Inbound (Receive) Signals:
PE8RN[3:0]	I	PCI Express Graphics Port Fourth x4, Negative Phase Inbound (Receive) Signals:
PE8TP[3:0]	O	PCI Express Graphics Port Fourth x4, Positive Phase Outbound (Transmit) Signals:
PE8TN[3:0]	O	PCI Express Graphics Port Fourth x4, Negative Phase Outbound (Transmit) Signals:

2.4 System Management Bus Interfaces

There are six 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

Signal Name	Type	Description
CFGSMBCLK	I/O	**Slave SMB Clock:** System Management Bus Clock
CFGSMBDATA	I/O	**Slave SMB Data:** SMB Address/Data
GPIOSMBCLK	I/O	**PCI Express SMB Clock:** System Management Bus Clock
GPIOSMBDATA	I/O	**PCI SMB Data:** SMB Address/Data
SPD0SMBCLK	I/O	**FB-DIMM Channel 0 SMB Clock:** FB-DIMM Memory Serial Presents Detect 0, System Management Bus Clock
SPD0SMBDATA	I/O	**FB-DIMM Channel 0 SMB Data:** FB-DIMM Memory Serial Presents Detect 0, SMB Address/Data
SPD1SMBCLK	I/O	**FB-DIMM Channel 1 SMB Clock:** FB-DIMM Memory Serial Presents Detect 1, System Management Bus Clock
SPD1SMBDATA	I/O	**FB-DIMM Channel 1 SMB Data:** FB-DIMM Memory Serial Presents Detect 1, SMB Address/Data

Signal Name	Type	Description
SPD2SMBCLK	I/O	**FB-DIMM Channel 2 SMB Clock:** FB-DIMM Memory Serial Presents Detect 2, System Management Bus Clock
SPD2SMBDATA	I/O	**FB-DIMM Channel 2 SMB Data:** FB-DIMM Memory Serial Presents Detect 2, SMB Address/Data
SPD3SMBCLK	I/O	**FB-DIMM Channel 3 SMB Clock:** FB-DIMM Memory Serial Presents Detect 3, System Management Bus Clock
SPD3SMBDATA	I/O	**FB-DIMM Channel 3 SMB Data:** FB-DIMM Memory Serial Presents Detect 3, SMB Address/Data

2.5 Extended Debug Port Signal List

Signal Name	Type	Description
XDPCLK1X	I/O	Bus Clock Reference
XDPD_N[15:0]	I/O	**Data Bus**.
XDPSTBN_N XDPSTBP_N	I/O	**Data Bus Strobe Negative and Positive Phases.**
XDPRDYACK_N	I/O	Ready ACK
XDPRDYREQ_N	I/O	Bus Ready Request.
XDPVREF	Analog	XDP Voltage Reference

2.6 JTAG Bus Signal List

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

2.7 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.
ERR_N[2:0]	O	**Error Output:** Error output signal: ERR_N[0] = Correctable and recoverable error ERR_N[1] = Uncorrectable error from the MCH ERR_N[2] = Fatal error from the MCH
FSBDRVCRES	Analog	**Processor Bus Compensation**:
FSBODTCRES	Analog	**Processor Bus Compensation:**
FSBSLWCRES	Analog	**Processor Bus Slew Rate Compensation:**
FSBSLWCTRL	CMOS	**Processor Bus Slew Rate Control:**
PWRGOOD	I	**Power OK:** When asserted, this signal indicates that all power supplies are in specification.

Signal Name	Type	Description
PSEL[2:0]	I	**Processor Speed Select.** **Selects the clock speed for the FSB, Core and FB-DIMM interfaces. 000: 266 MHz; 001: 133 MHz; 011: 166; 100: 333; 110: 400, others reserved.**
RESETI_N	I	**MCH Reset:** This is the hard reset
PLLPWRDET	I	**PLL Power Detect:** Assert prior to PWRGOOD. Indicates power and master clocks are stable.
RESERVED	---	**Reserved Pin. Tie to GND.**
THERMALERT_N	O	Thermal Alert. (threshold reached)
THERMTRIP_N	O	Thermal Trip (catastrophic)
XOROUT	O	**XOR tree output**
VSCALEN	I	**Voltage Scale Enable**: At Power-on when de-asserted, allows the MCH to maintain VID at 1.25v
VID[6:1]	O	Voltage Identifier
VSSSEN	Analog	**Quiet VSS:** Quiet VSS for Thermal Sensor
VCCSEN	Analog	**Quiet VCC:** Quiet VCC for Thermal Sensor

2.8 Power and Ground Signals

Signal Name	Description
VCC	**VCC Supply:** This is the 1.25 V core voltage.
VSS	**Ground Return:** Common return for power supplies
VTT	**VTT Supply:** VTT is a 1.1/1.2V FSB supply
VCCAFB-DIMM	**Analog VCC for System Memory:** VCCAFB-DIMM is 1.25V for DDR2 power.
VCCAPE125	Analog VCC for PCI Express ports.
VCCACPLL	Analog VCC for the PLL
VCCAFB-DIMM01PLL	VCC: Analog Voltage for the PLL
VCCAFB-DIMM23PLL	Analog VCC for the PLL
VCCAFB-DIMM01PLL18	VCC: 1.8V supply for the PLLs. See also FB-DIMM Branch 1.
VCCSF	**VCC Snoop Filter Supply:** This is the 1.5 V core voltage on a separate plane for the snoop filter.
VCCDCPLL	Digital Voltage for the PLL
VCCMISC125	Misc 1.25V I/O Supply
VCCMISC33	Misc 3.3V I/O Supply

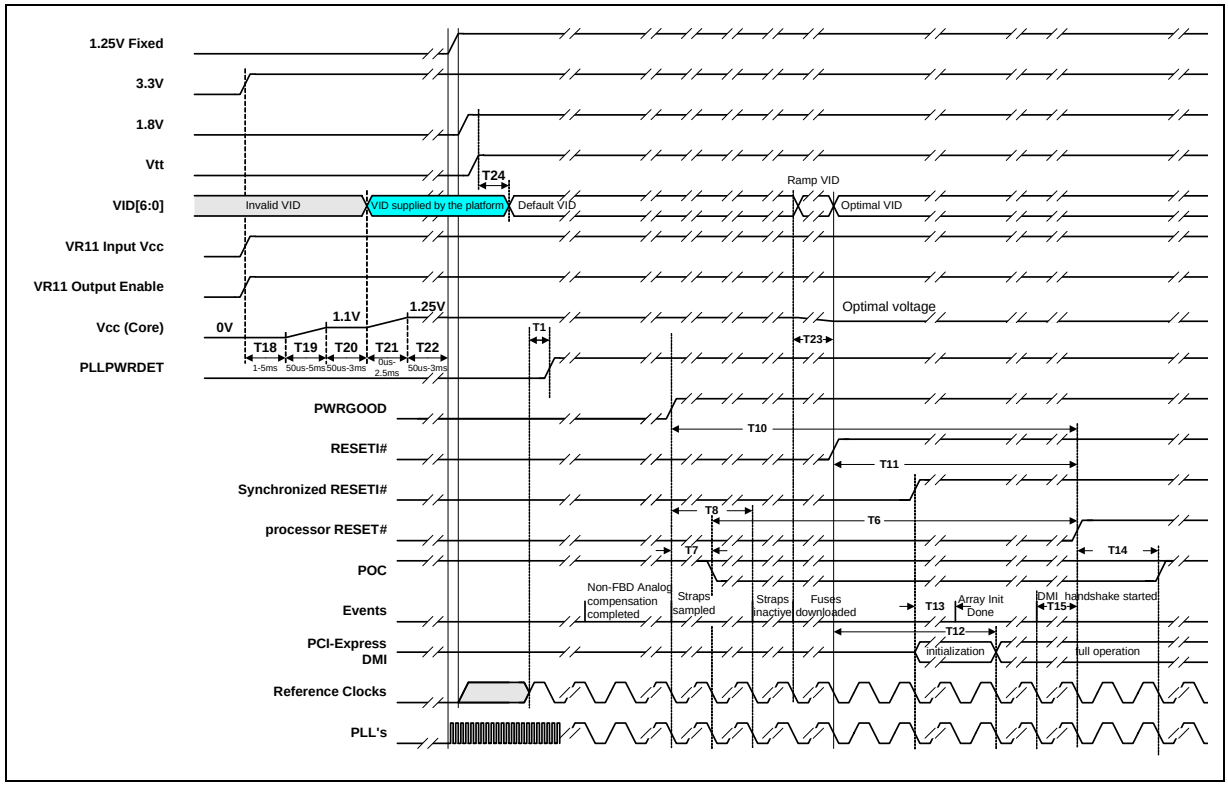
2.9 MCH Sequencing and Reset Requirements

2.9.1 Timing Diagrams

2.9.1.1 Power-Up

The power-up sequence is illustrated in Figure 2-1.

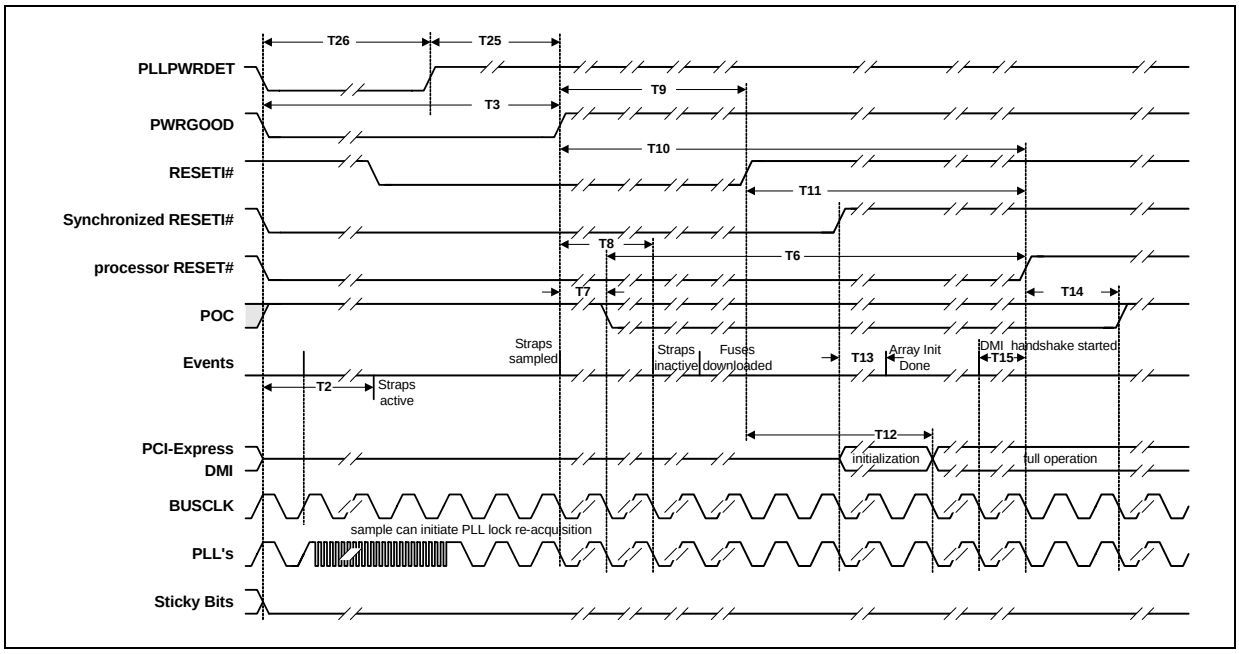
Figure 2-1. Power-Up



2.9.1.2 Power Good

The PWRGOOD reset sequence is illustrated in Figure 2-2.

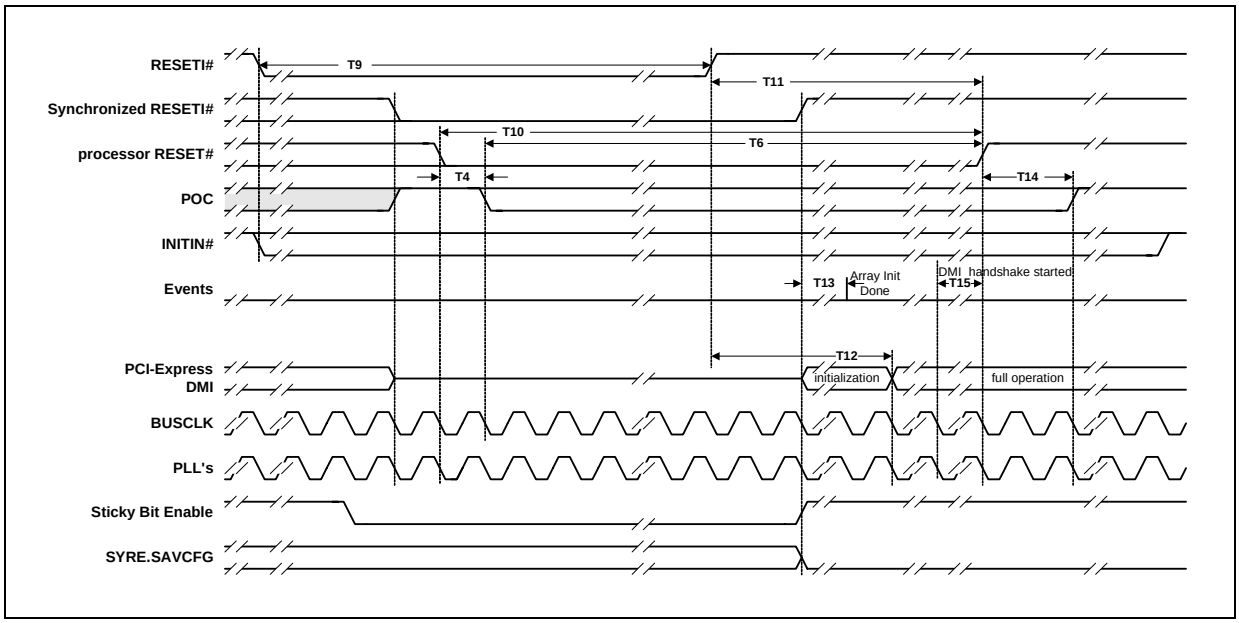
Figure 2-2. PWRGOOD



2.9.1.3 Hard Reset

The Hard Reset sequence is illustrated in Figure 2-3.

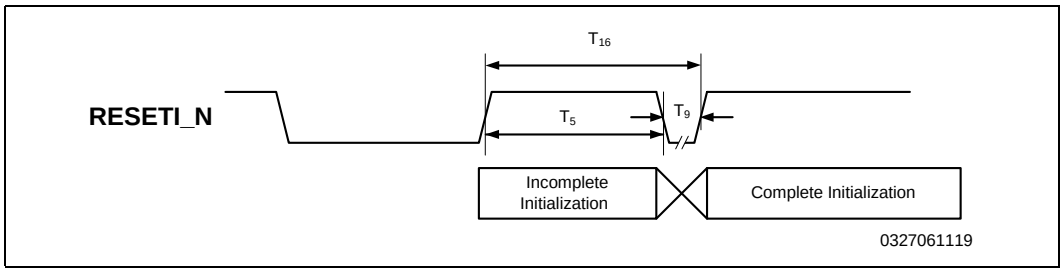
Figure 2-3. Hard Reset



2.9.1.4 RESETI_N Retriggering Limitations

Figure 2-4 shows the timing for a RESETI_N retrigger.

Figure 2-4. RESETI_N Retriggering Limitations



2.9.2 Reset Timing Requirements

Table 2-3 specifies the timings drawn in Figure 2-1, Figure 2-2, Figure 2-3, and Figure 2-4. Nominal clock frequencies are described. Specifications still hold for derated clock frequencies.

Table 2-3. Power Up and Hard Reset Timings (Sheet 1 of 2)

Timing	Description	Min	Max	Comments
T1	Power and master clocks stable to PLLPWRDET signal assertion	0 ms		
T2	PWRGOOD de-assertion to straps active		40 ns	
T3	PWRGOOD de-assertion	200 us		Minimum PWRGOOD de-assertion time while power and platform clocks are stable.
T4	POC after RESET# assertion delay	1 BUSCLK		
T5	Platform reset de-assertion to platform reset assertion	50 BUSCLK's		Minimum re-trigger time on RESETI# de-assertion.
T6	POC valid until RESET# de-assertion delay	480 us		CPU Core PLL lock acquisition time
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	RESETI# signal assertion during PWRGOOD / PWROK signal assertion	1 ms		This delay can be provided by the Intel 631xESB/632xESB I/O Controller Hub or by system logic
T10	RESET# assertion during processor PWRGOOD assertion	1 ms	10 ms	Processor EMTS specification.
T11	RESETI# signal de-assertion to processor RESET# signal de-assertion	480 us		486.0us for 1,333MHz FSB 607.5us for 1,066MHz FSB
T12	RESETI# signal de-assertion to completion of PCI Express initialization sequence		1,250,000 PECLK's	PCI Express clock is 100MHz
T13	Array Initialization duration		200 cycles	
T14	POC hold time after RESET# de-assertion	2 BUSCLK's	19 BUSCLK's	Processor EMTS specification
T15	Initiation of ESI reset sequence to processor RESET# signal de-assertion		10,000 PECLK's + T17	Intel 631xESB/632xESB I/O Controller Hub specification
T16	RESETI# re-trigger delay	T5 + T9		
T17	CPU_RESET_DONE capture timer	2,000 BUSCLK's		
T18	VR11 input to output delay (start ramp)	1 ms	5 ms	VR11 specification
T19	VR11 output transition (0 to VR11 default)	50 us	5 ms	VR11 specification
T20	VR11 default to sample VID's	50 us	3 ms	VR11 specification
T21	VR11 output transition from VR11 default to 1.25V	0 us	2.5 ms	VR11 specification
T22	VR11 1.25V to VR11 pwrgood output assert	50 us	3 ms	VR11 specification

Table 2-3. Power Up and Hard Reset Timings (Sheet 2 of 2)

Timing	Description	Min	Max	Comments
T23	Ramp from 1.25V to Optimal Voltage		45 us	15 steps at 5us each
T24	Vtt stable to VID valid	10 us		Change-over from platform-supplied VID to MCH-supplied VID may not commence until this delay has elapsed.
T25	PLLPWRDET assertion to PWRGOOD assertion	100 us	15 ms	min: P1263 PLL stabilization max: P1263 Fuse reliability
T26	PLLPWRDET de-assertion	100 us		Minimum PLLPWRDET de-assertion time while power and platform clocks are stable

Table 2-4 summarizes the Intel® 5400 MCH Initialization Timings.

Table 2-4. Critical MCH Initialization Timings

Sequence	Started by	Maximum Length	Covered by Timing parameter
FB-DIMM PLL lock	BRN[1:0]PLLCTRL.DISPLL deassertion	100 uSec	N/A
Intel® 5400 MCH Core, FSB, FB-DIMM PLL lock	Stable power and master clock	666,667 333 MHz cycles	T_1
Intel® 5400 MCH PCI Express PLL lock	Stable power and master clock	200,000 100 MHz cycles	
Array initialization	Synchronized RESETI_N Deassertion	200 cycles	T_{13}
Fuse download	PWRGOOD Assertion	333,333 333 MHz cycles	T_9

2.9.3 Miscellaneous Requirements and Limitations

- Power rails and stable BUSCLK, FB-DIMM{0/1}CLK, and PECLK master clocks remain within specifications through all but power-up reset.
- Frequencies (e.g. 266 MHz) described in this chapter are nominal. The Intel® 5400 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 or PCI agents.
- Hard Reset is not guaranteed to correct all illegal configurations or malfunctions. Software can configure sticky bits in the Intel® 5400 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.
- After cold boot initialization (DDRFRQ set, FB-DIMMRST de-assert, etc.), the FB-DIMM channels will be enabled for packet levelization (FB-DIMMST.STATE="Ready" or "RecoveryReady" state) upon completion of a hard reset. Software should inspect the FB-DIMMST.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 Intel® 5400 MCH cannot meet the SPD data $t_{SU,STO}$ timing specification. Since the Intel® 5400 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.

2.10 Platform Signal Routing Topology Diagrams

Figure 2-5. Simplest Power Good Distribution

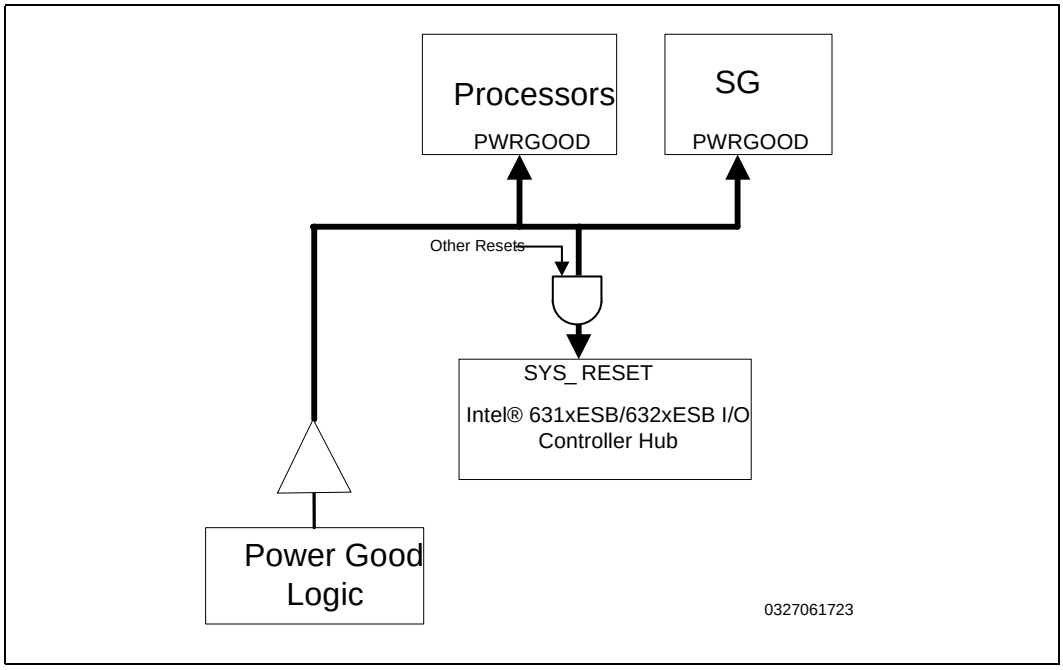


Figure 2-6. Basic System Reset Distribution

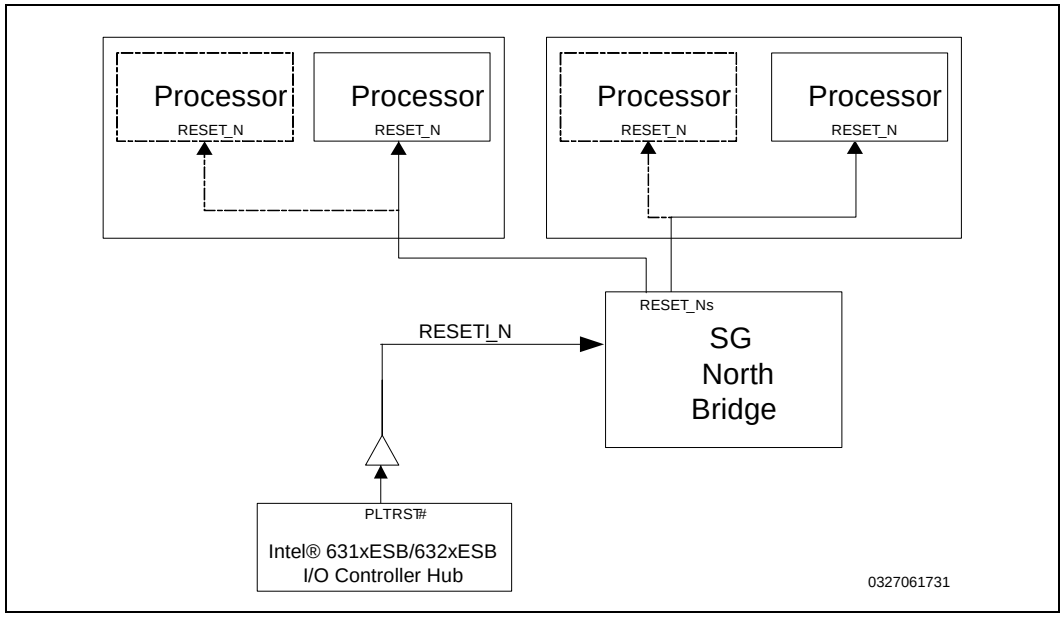
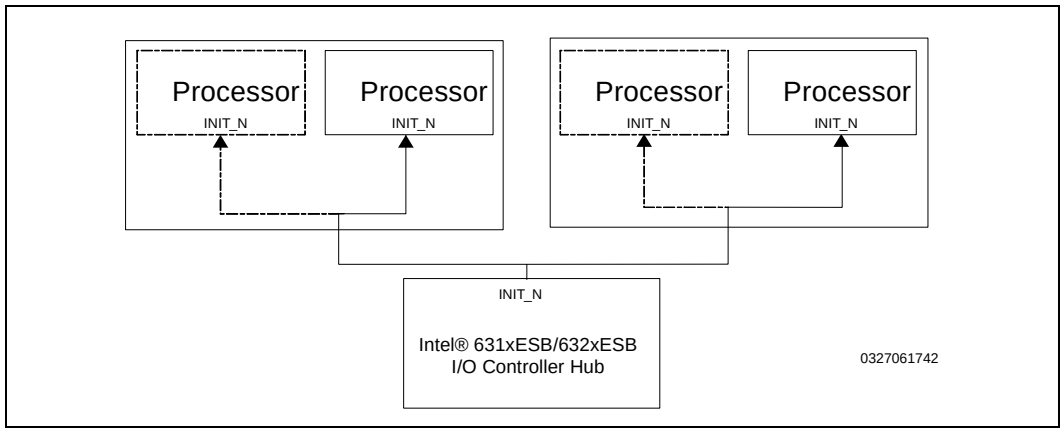


Figure 2-7. Basic INIT_N Distribution



2.10.1 Intel® 5400 MCH Customer Reference Platform (SRP) Reset Topology

Typical platform level reset implementation is described in the Quad-Core Intel® Xeon® processor 5300 series (1066 MHz) and the platform design guide.

2.11 Signals Used as Straps

2.11.1 Functional Straps

Signal Name	Type	Description
PSEL[2:0]	I	**Selects Clock Bus Speed.**

2.11.2 Strap Input Signals

Signal Name	Type	Description
PSEL[2:0]	I	**Selects Processor Bus Speed.**

§

3 Register Description

The Intel® 5400 chipset MCH contains 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 (control port 0, i.e., DRAM configuration, other chipset operating parameters, and optional features).

The MCH supports PCI configuration space accesses using the mechanism denoted as Configuration Mechanism 1 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 2.0*. The MCH supports registers in PCI Express extended space. All MCH registers in Intel® 5400 chipset appear on PCI Bus #0.

In addition, the MCH registers can be accessed by a memory mapped register access mechanism (as MMIO) and a PCI configuration access mechanism (only PCI space registers). 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, 32-bit or 64-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.

3.1 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**. These bits can only be read by software, writes have no effect. The value of the bits is determined by the hardware only.
RW	**Read / Write:** These bits can be read and written by software.
RWO	**Read / Write Once:** These bits can be read by software. After reset, these bits can only be written by software once, after which the bits becomes 'Read Only'.
RWL	**Read / Write Lock:** These bits can be read and written by software. Hardware can make these bits 'Read Only' via a separate configuration bit or other logic.
RW1C	**Read / Write 1 to Clear:** These bits can be read and cleared by software. Writing a '1' to a bit clears it, while writing a '0' to a bit has no effect.
RC	**Read Clear:** These bits can only be read by software, but a read causes the bits to be cleared. *NOTE: Use of this attribute type is deprecated, as reads with side-effects are harmful for debug.*
RCW	**Read Clear / Write:** These bits can be read and written by software, but a read causes the bits to be cleared. *NOTE: Use of this attribute type is deprecated, as reads with side-effects are harmful for debug.*

Term	Description
ROS	**RO Sticky:** These bits can only be read by software, writes have no effect. The value of the bits is determined by the hardware only. These bits are only reinitialized to their default value by a PWRGOOD reset.
RWS	**R / W Sticky:** These bits can be read and written by software. These bits are only reinitialized to their default value by a PWRGOOD reset.
RW1CS	**R / W1C Sticky:** These bits can be read and cleared by software. Writing a '1' to a bit clears it, while writing a '0' to a bit has no effect. These bits are only reinitialized to their default value by a PWRGOOD reset.
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.
RV	**Reserved:** These bits are reserved for future expansion and their value must not be modified by software. When writing these bits, software must preserve the value read. The bits are read-only must return '0' when read.

3.2 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® 5400 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. 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 fourteen PCI devices within a single physical component. The configuration registers for these devices are mapped as devices residing on PCI bus 0.

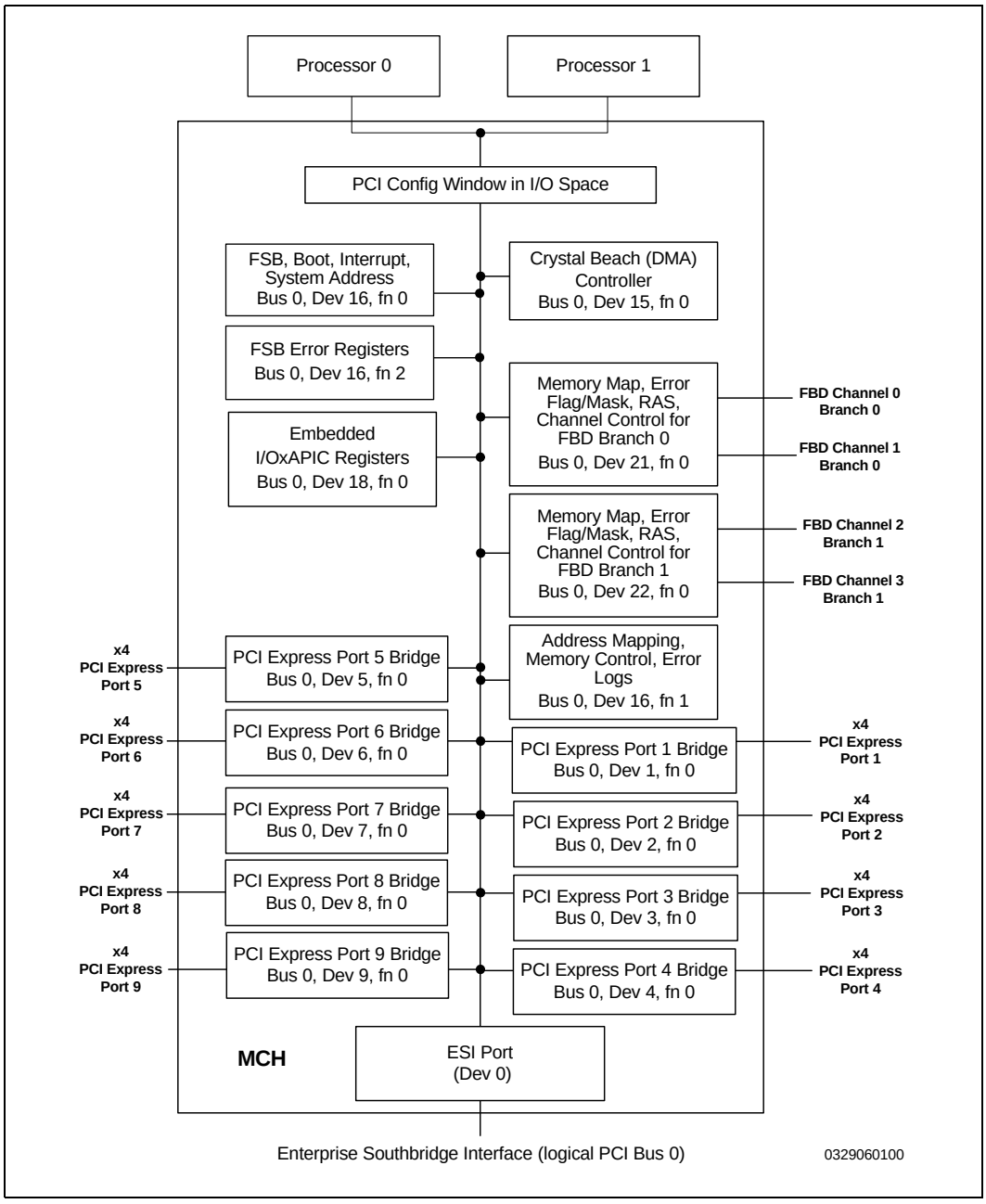
- **Device 0:** ESI Port 0. Logically, this appears as a PCI device that resides on PCI bus 0. Physically Device 0, Function 0 contains the PCI Express configuration registers for the ESI port, and other MCH specific registers.
- **Device 1:** PCI Express 1. Logically this appears as a PCI device residing on bus 0. Device 1, Function 0 is routed to the PCI Express configuration registers for PCI Express port 1. When PCI Express ports 1 and 2 are combined into a single x8 port, or when ports 1, 2, 3 and 4 are combined into a single x16 port, the combined port is controlled by port 1 registers. The other Devices 2-4, Function 0 (ports 2-4) configuration registers are inactive. PCI Express port 1 has a DID of 4021h.
- **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 1 and 2 are combined into a single x8 port, or when ports 1, 2, 3 and 4 are combined into a single x16 port, the combined port is controlled by port 1 registers. The other Devices 2-4, Function 0 (ports 2-4) configuration registers are inactive. PCI Express port 2 has a DID of 4022h.
- **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 3 and 4 are combined into a single x8 port, the combined port is controlled by port 3 registers, Device 4, Function 0 (port 4) configuration registers are inactive. When ports 1, 2, 3 and 4 are combined into a

single x16 port, the combined port is controlled by port 1 registers. The other Devices 2-4, Function 0 (ports 2-4) configuration registers are inactive. PCI Express port 3 has a DID of 4023h.

- **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 3 and 4 are combined into a single x8 port, Device 3, Function 0 contains the configuration registers and Device 4, Function 0 (port 4) configuration registers are inactive. When ports 1, 2, 3 and 4 are combined into a single x16 port, the combined port is controlled by port 1 registers. The other Devices 2-4, Function 0 (ports 2-4) configuration registers are inactive. PCI Express port 4 has a DID of 4024h.
- **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 5 and 6 are combined into a single x8 port, or when ports 5, 6, 7 and 8 are combined into a single x16 port, the combined port is controlled by port 5 registers. The other Devices 6-8, Function 0 (ports 6-8) configuration registers are inactive. PCI Express port 5 has a DID of 4025h.
- **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 5 and 6 are combined into a single x8 port, or when ports 5, 6, 7 and 8 are combined into a single x16 port, the combined port is controlled by port 5 registers. The other Devices 6-8, Function 0 (ports 6-8) configuration registers are inactive. PCI Express port 6 has a DID of 4026h.
- **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 7 and 8 are combined into a single x8 port, the combined port is controlled by port 7 registers, Device 8, Function 0 (port 8) configuration registers are inactive. When ports 5, 6, 7 and 8 are combined into a single x16 port, the combined port is controlled by port 5 registers. The other Devices 6-8, Function 0 (ports 6-8) configuration registers are inactive. PCI Express port 7 has a DID of 4027h.
- **Device 8:** PCI Express 8. Logically this appears as a PCI device residing on bus 0. Device 8, Function 0 contains the PCI Express configuration registers for PCI Express port 8. When PCI Express ports 7 and 8 are combined into a single x8 port, the combined port is controlled by port 7 registers, Device 8, Function 0 (port 8) configuration registers are inactive. When ports 5, 6, 7 and 8 are combined into a single x16 port, the combined port is controlled by port 5 registers. The other Devices 6-8, Function 0 (ports 6-8) configuration registers are inactive. PCI Express port 8 has a DID of 4028h.
- **Device 9:** PCI Express 9. Logically this appears as a PCI device residing on bus 0. Device 9, Function 0 contains the PCI Express configuration registers for PCI Express port 9. The port is designed for x4 port use only. PCI Express port 8 has a DID of 4029
- **Device 16**: Device 16, Function 0 is routed to the Frontside Bus (FSB) Controller, Boot, 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. The device DID of 4030h.
- **Device 18:** Device 18, Function 0 is routed to the IOxAPIC registers. Device DID of 4032h.
- **Device 21:** Device 21, Function 0, Memory Map, Error Flag/Mask, RAS and Channel Control registers for FB-DIMM branch 0. Device DID of 4035h.

- **Device 22:** Device 22, Function 0, Memory Map, Error Flag/Mask, RAS and Channel Control registers for FB-DIMM Branch 1. Device DID of 4036h.

Figure 3-1. Conceptual Intel® 5400 MCH PCI Configuration Diagram



3.3 Routing Configuration Accesses

Intel® 5400 MCH supports both PCI Type 0 and Type 1 configuration access mechanisms as defined in the *PCI-Express Base Specification*, Revision 2.0. PCI Revision 3.0 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.

3.3.1 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 Local Bus Specification,* Revision 2.3 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 Local Bus Specification,* Revision 2.3 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.

3.3.2 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.

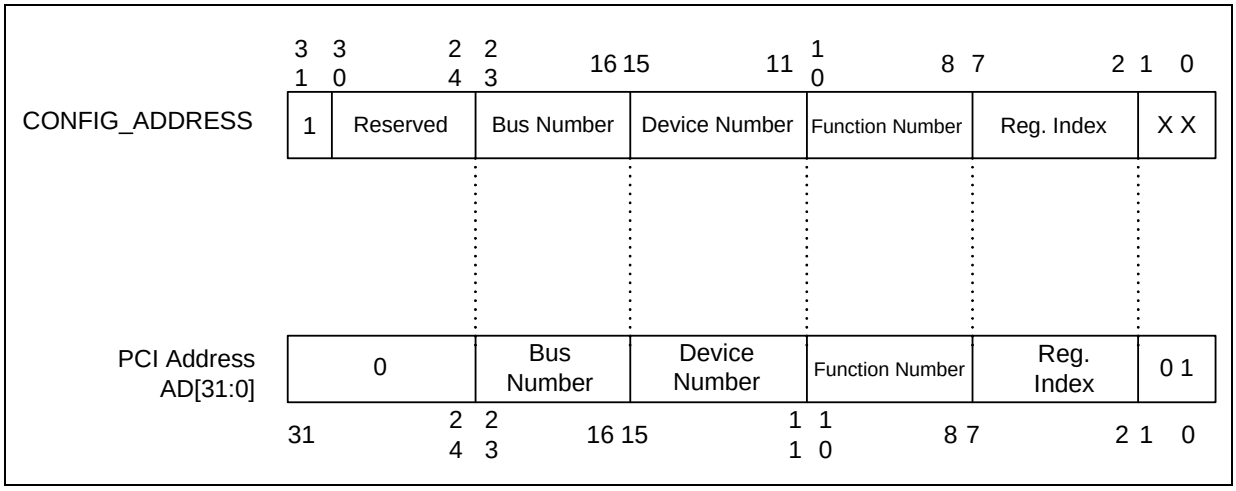
The ESI bridge entity within the MCH is hardwired as Device 0 on PCI Bus 0. The ESI bridge passes PCI south bridge configuration requests to the south bridge.

3.3.3 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 3-2. 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 3-2. Type 1 Configuration Address to PCI Address Mapping



3.4 Device Mapping

Each component in a Intel® 5400 MCH 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® 5400 chipset platform must support Type 0 configuration accesses. All MCH registers in the Intel® 5400 MCH appear on Bus #0.

The Intel® 5400 MCH configuration registers reside in the configuration space defined by Bus, Device, Function, Register address or in memory space such as Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d) and IOxAPIC. 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 3-1. 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 from ESI or PCI Express	Not permitted to access MCH or FB-DIMM DIMM regs and will be master aborted. Peer to Peer accesses targeting valid MMIO space will be forwarded through appropriate decoding.							
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[10:1]	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[10:1]	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 5-28)	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[1]		Cannot access	A[7:3] BE[7:4]	BE[7:0]	n/a

Notes:

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

3.4.1 Device Identification for Intel® 5400 MCH Component

All devices in the Intel® 5400 chipset MCH reside on Bus 0. The following table describes the root device ID for different MCH versions.

Table 3-2. Memory Control Hub ESI Device Identification

Component	Register Group	DID	Device	Function	Comment
Intel® 5400 chipset	Enterprise Southbridge Interface	4000h	0	0	

3.4.2 Special Device and Function Routing

All devices in the Intel® 5400 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 3-3. Functions Specially Handled by the MCH

Component	Register Group	DID	Device	Function	Comment
MCH	PCI Express Port 1	4021h	1	0	x16, x8, or x4 max link width.
MCH	PCI Express Port 2	4022h	2	0	x4 max link width
MCH	PCI Express Port 3	4023h	3	0	x8 or x4 max link width
MCH	PCI Express Port 4	4024h	4	0	x4 max link width
MCH	PCI Express Port 5	4025h	5	0	x16, x8, or x4 max link width
MCH	PCI Express Port 6	4026h	6	0	x4 max link width
MCH	PCI Express Port 7	4027h	7	0	x8 or x4 max link width
MCH	PCI Express Port 8	4028h	8	0	x4 max link width
MCH	PCI Express Port 9	4029h	9	0	x4 max link width
MCH	PCI Express Ports 4 - 1 & ESI port	402Dh	13	4 - 0	
MCH	PCI Express Ports 9 - 5	402Eh	14	4 - 0	
MCH	Processor Bus, Boot, Interrupt, System Address	4030h	16	0	
MCH	Address Mapping, Memory Control, Error Logs			1	
MCH	FSB Error Registers			2	
MCH	Coherency Engine, Data Manager & Snoop Filter.	4031	17	0	
MCH	IOXAPIC	4032h	18	0	
MCH	Memory Map, Error Flag/Mask, RAS, Channel Control for FB-DIMM Branch 0	4035h	21	0 - 1	
MCH	Memory Map, Error Flag/Mask, RAS, Channel Control for FB-DIMM Branch 1	4036h	22	0 - 1	

To comply with the *PCI Express Base Specification 2.0*, accesses to non-existent functions, registers, and bits will be master aborted. This behavior is defined in the following table:

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

Access to	Writes	Reads
Devices listed in Table 3-2, but to functions not listed	Have no effect	MCH returns all ones
Devices listed in Table 3-2, but to registers not listed in Section 3.8	Have no effect	MCH returns all zeroes
Reserved bits in registers	Software must read-modify-write to preserve the value	MCH returns all zeroes

3.5 I/O Mapped Registers

There are only two I/O addresses that affect Intel® 5400 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.

3.5.1 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 3-1.

Table 3-5. 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

3.5.2 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.

Table 3-6. 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**

3.6 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 3-7. 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

3.7 Detailed Configuration Space Maps

Table 3-8. Device 0, Function 0: PCI Express PCI space

DID		VID		00h	PEXSLOTCAP		80h
PCISTS		PCICMD		04h	PEXSLOTSTS	PEXSLOTCTRL	84h
CCR			RID	08h	PEXRTCAP	PEXRTCTRL	88h
BIST	HDR	PRI_LT	CLS	0Ch	PEXRTSTS		8Ch
				10h	PEXDEVCAP2		90h
				14h	PEXDEVSTS2	PEXDEVCTRL2	94h
				18h	PEXLNKCAP2		98h
				1Ch	PEXLNKSTS2	PEXLNKCTRL2	9Ch
				20h	PEXLNKCAP2		A0h
				24h			A4h
				28h			A8h
SID		SVID		2Ch			ACh
				30h	VT_BAR		B0h
			CAPPTR	34h		VTCTRL	B4h
				38h			B8h
		INTP	INTL	3Ch			BCh
				40h			C0h
SSCTRL	IOAPIC CTRL			44h			C4h
PEXCTRL				48h			C8h
				4Ch			CCh
PMCAP		PMCAPLST		50h			D0h
PMCSR				54h	ESICTRL		D4h
MSICTRL		MSICAPLST		58h	PEXGCTRL		D8h
MSIAR				5Ch			DCh
MSIDR				60h	PEXLWTCTRL		E0h
				64h			E4h
DEVPRES				68h			E8h
PEXCAP		PEXCAPLST		6Ch			ECh
PEXDEVCAP				70h			F0h
PEXDEVSTS		PEXDEVCTRL		74h			F4h
PEXLNKCAP				78h			F8h
PEXLNKSTS		PEXLNKCTRL		7Ch			FCh

Table 3-9. 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
		140h			1C0h
PEX_ERR_PIN_MASK	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
EMASK_PEX_UNIT		164h			1E4h
PEX_UNIT_STS		168h			1E8h
PEX_UNIT_SEV		16Ch			1ECh
PEX_UNIT_MASK		170h			1F0h
PEX_UNIT_RPTCTRL		174h			1F4h
		178h			1F8h
		17Ch			1FCh

Table 3-10. Device 1-9, Function 0: PCI Express PCI space

DID			VID	00h	PEXSLOTCAP		80h
PCISTS			PCICMD	04h	PEXSLOTSTS	PEXSLOTCTRL	84h
CCR			RID	08h	PEXRTCAP	PEXRTCTRL	88h
BIST	HDR	PRI_LT	CLS	0Ch	PEXRTSTS		8Ch
				10h	PEXDEVCAP2		90h
				14h	PEXDEVSTS2	PEXDEVCTRL2	94h
SEC_LT	SUBUSN	SBUSN	PBUSN	18h	PEXLNKCAP2		98h
SECSTS		IOLIM	IOBASE	1Ch	PEXLNKSTS2	PEXLNKCTRL2	9Ch
MLIM		MBASE		20h			A0h
PMLIM		PMBASE		24h			A4h
PMBU				28h			A8h
PMLU				2Ch			ACh
				30h		SCAPL	B0h
			CAPPTR	34h	SSID	SSVID	B4h
				38h			B8h
BCTRL		INTP	INTL	3Ch			BCh
				40h			C0h
SSCTRL				44h			C4h
PEXCTRL				48h			C8h
INTXSWZCTRL				4Ch			CCh
PMCAP		PMCAPLST		50h			D0h
PMCSR				54h			D4h
MSICTRL		MSICAPLST		58h			D8h
MSIAR				5Ch			DCh
MSIDR				60h			E0h
				64h			E4h
				68h			E8h
PEXCAP		PEXCAPLST		6Ch			ECh
PEXDEVCAP				70h			F0h
PEXDEVSTS		PEXDEVCTRL		74h			F4h
PEXLNKCAP				78h			F8h
PEXLNKSTS		PEXLNKCTRL		7Ch			FCh

Table 3-11. Device 1-9, 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
		140h			1C0h
PEX_ERR_PIN_MASK	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
EMASK_PEX_UNIT		164h			1E4h
PEX_UNIT_STS		168h			1E8h
PEX_UNIT_SEV		16Ch			1ECh
PEX_UNIT_MASK		170h			1F0h
PEX_UNIT_RPTCTRL		174h			1F4h
		178h			1F8h
		17Ch			1FCh

Table 3-12. Device 16, Function 0: Processor Bus, Boot, and Interrupt

DID		VID		00h	XTPR0	80h
				04h	XTPR1	84h
CCR			RID	08h	XTPR2	88h
	HDR			0Ch	XTPR3	8Ch
				10h	XTPR4	90h
				14h	XTPR5	94h
				18h	XTPR6	98h
				1Ch	XTPR7	9Ch
				20h	XTPR8	A0h
				24h	XTPR9	A4h
				28h	XTPR10	A8h
SID		SVID		2Ch	XTPR11	ACh
				30h	XTPR12	B0h
				34h	XTPR13	B4h
				38h	XTPR14	B8h
				3Ch	XTPR15	BCh
CPURSTCAPTMR		SYRE		40h	BOFL0	C0h
POC				44h	BOFL1	C4h
AMBASE				48h	BOFL2	C8h
				4Ch	BOFL3	CCh
AMR				50h	SPAD0	D0h
MAXDIMMPERCH	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
REDIRBUCKETS				68h	SPADS2	E8h
REDIRCTL		CPUKILL1	CPUKILL0	6Ch	SPADS3	ECh
				70h	PROCENABLE	F0h
				74h		F4h
				78h		F8h
				7Ch		FCh

Table 3-13. Device 16, Function 0: Processor Bus 0 Error registers

											100h
											104h
											108h
											10Ch
FSB0PMCH				FSB0PMDH							110h
FSB0PMR0											114h
FSB0PMR1											118h
											11Ch
FSB0PMEL0											120h
FSB0PMEL1											124h
FSB0PMEM0											128h
FSB0PMEM1											12Ch
FSB0PMEH0											130h
FSB0PMEH1											134h
											138h
											13Ch
											140h
											144h
											148h
											14Ch
											150h
											154h
											158h
											15Ch
											160h
											164h
											168h
											16Ch
											170h
											174h
											178h
											17Ch

NERR_NF_FSB				**NERR_FAT_FSB**				**FERR_NF_FSB**			180h
NRECFSB											184h
RECFSB											188h
NRECADDRL											18Ch
EMASK_FSB										**NRECADDRH**	190h
ERR1_FSB								**ERR0_FSB**			194h
MCERR_FSB								**ERR2_FSB**			198h
											19Ch
											1A0h
											1A4h
											1A8h
											1ACh
											1B0h
											1B4h
											1B8h
											1BCh
											1C0h
											1C4h
											1C8h
											1CCh
											1D0h
											1D4h
											1D8h
											1DCh
											1E0h
											1E4h
											1E8h
											1ECh
											1F0h
											1F4h
											1F8h
											1FCh

Table 3-14. Device 16, Function 0: Processor Bus 1 Error registers

		400h	**NERR_NF_FSB**	**NERR_FAT_FSB**	**FERR_NF_FSB**	**FERR_FAT_FSB**	480h
		404h	**NRECFSB**				484h
		408h	**RECFSB**				488h
		40Ch	**NRECADDRL**				48Ch
FSB0PMCH	FSB0PMDH	410h	**EMASK_FSB**			**NRECADDRH**	490h
FSB0PMR0		414h	**ERR1_FSB**		**ERR0_FSB**		494h
FSB0PMR1		418h	**MCERR_FSB**		**ERR2_FSB**		498h
		41Ch					49Ch
FSB0PMEL0		420h					4A0h
FSB0PMEL1		424h					4A4h
FSB0PMEM0		428h					4A8h
FSB0PMEM1		42Ch					4ACh
FSB0PMEH0		430h					4B0h
FSB0PMEH1		434h					4B4h
		438h					4B8h
		43Ch					4BCh
		440h					4C0h
		444h					4C4h
		448h					4C8h
		44Ch					4CCh
		450h					4D0h
		454h					4D4h
		458h					4D8h
		45Ch					4DCh
		460h					4E0h
		464h					4E4h
		468h					4E8h
		46Ch					4ECh
		470h					4F0h
		474h					4F4h
		478h					4F8h
		47Ch					4FCh

Table 3-15. Device 16, Function 1: Memory Branch Map, Control, Errors

DID		VID	00h		MIR0		80h
PCISTS		PCICMD	04h		MIR1		84h
CCR		RID	08h				88h
	HDR		0Ch		AMIR0		8Ch
			10h		AMIR1		90h
			14h				94h
			18h	FERR_FAT_FB-DIMM			98h
			1Ch	NERR_FAT_FB-DIMM			9Ch
			20h	FERR_NF_FB-DIMM			A0h
			24h	NERR_NF_FB-DIMM			A4h
			28h	EMASK_FB-DIMM			A8h
SID		SVID	2Ch	ERR0_FB-DIMM			ACh
			30h	ERR1_FB-DIMM			B0h
			34h	ERR2_FB-DIMM			B4h
			38h	MCERR_FB-DIMM			B8h
			3Ch		DIMMISO		BCh
MC			40h				C0h
MS			44h				C4h
DRTA			48h				C8h
DRTB			4Ch				CCh
ERRPER			50h				D0h
	DDRFRQ		54h				D4h
MCA			58h				D8h
MCCTRL			5Ch				DCh
			60h				E0h
	DRAMPDCTL		64h				E4h
			68h				E8h
	TOLM		6Ch				ECh
			70h				F0h
			74h				F4h
			78h				F8h
			7Ch				FCh

Table 3-16. Device 16, Function 1: Memory Registers

				100h
				104h
				108h
				10Ch
				110h
				114h
				118h
				11Ch
				120h
				124h
				128h
				12Ch
				130h
				134h
				138h
				13Ch
				140h
				144h
				148h
				14Ch
				150h
				154h
				158h
				15Ch
FB-DIMMTOHOSTGRCFG0				160h
FB-DIMMTOHOSTGRCFG1				164h
HOSTTOFB-DIMMGRCFG				168h
GRFB-DIMMTO-HOSTDBLCFG	**GRBUBBLECFG**	**GRHOST-FULLCFG**	**GRFB-DIMMVLDCFG**	16Ch
				170h
				174h
				178h
				17Ch

				180h
				184h
				188h
				18Ch
				190h
				194h
				198h
				19Ch
				1A0h
				1A4h
				1A8h
				1ACh
				1B0h
				1B4h
				1B8h
				1BCh
				1C0h
				1C4h
				1C8h
				1CCh
				1D0h
				1D4h
				1D8h
				1DCh
				1E0h
				1E4h
				1E8h
				1ECh
				1F0h
				1F4h
				1F8h
				1FCh

Table 3-17. Device 16, Function 1: Memory Registers

MEM0EINJMSK0							200h
MEM1EINJMSK1					MEM0EINJMSK1		204h
MEM1EINJMSK0							208h
							20Ch
							210h
							214h
							218h
							21Ch
							220h
							224h
							228h
							22Ch
							230h
							234h
							238h
							23Ch
							240h
							244h
							248h
							24Ch
							250h
							254h
							258h
							25Ch
							260h
							264h
							268h
							26Ch
							270h
							274h
							278h
							27Ch

							280h
							284h
							288h
							28Ch
							290h
							294h
							298h
							29Ch
							2A0h
							2A4h
							2A8h
							2ACh
							2B0h
							2B4h
							2B8h
							2BCh
							2C0h
							2C4h
							2C8h
							2CCh
							2D0h
							2D4h
							2D8h
							2DCh
							2E0h
							2E4h
							2E8h
							2ECh
							2F0h
							2F4h
							2F8h
							2FCh

Table 3-18. 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	NRECSF0			B0h
				34h	NRECSF1			B4h
				38h	RECSF0			B8h
				3Ch	RECSF1			BCh
FERR_Global				40h	FERR_NF_INT2	FERR_NF_INT	FERR_FAT_INT2	FERR_FAT_INT C0h
NERR_Global				44h	NERR_NF_INT2	NERR_NF_INT	NERR_FAT_INT2	NERR_FAT_INT C4h
			ERRFRZ	48h	NRECINT			C8h
				4Ch	RECINT			CCh
				50h	EMASK_INT			D0h
				54h	ERR0_INT			D4h
				58h	ERR1_INT			D8h
				5Ch	ERR2_INT			DCh
				60h	MCERR_INT			E0h
				64h				E4h
				68h				E8h
				6Ch				ECh
				70h	NERR_NF_THR	NERR_FAT_THR	FERR_NF_THR	FERR_FAT_THR F0h
				74h	EMASK_THR			F4h
				78h	ERR1_THR		ERR0_THR	F8h
				7Ch	MCERR_THR		ERR2_THR	FCh

Table 3-19. Device 16, Function 3: Memory Throttling

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	B0h
		34h	B4h
		38h	B8h
		3Ch	BCh
THRTCTRLA		40h	C0h
		44h	C4h
		48h	C8h
		4Ch	CCh
		50h	D0h
		54h	D4h
		58h	D8h
		5Ch	DCh
		60h	THRTLOWA E0h
THRTCTRL / THRTHI		64h	THRTLOWB E4h
	THRTSTS	68h	THRTLOWC E8h
		6Ch	THRTLOWD ECh
		70h	THRTMIDA F0h
		74h	THRTMIDB F4h
		78h	THRTMIDC F8h
		7Ch	THRTMIDD FCh

Table 3-20. Device 16, Function 4: On-Die Throttling

DID				VID				00h
PCISTS				PCICMD				04h
CCR							RID	08h
	HDR							0Ch
								10h
								14h
								18h
								1Ch
								20h
								24h
								28h
SID				SVID				2Ch
								30h
								34h
								38h
								3Ch
								40h
								44h
								48h
								4Ch
								50h
								54h
								58h
								5Ch
								60h
								64h
TSTHRPEX		TSTHRFSB			TSCTRL			68h
TSTHRHI					TSTHRLO			6Ch
TSFSC				CTHINT				70h
CTCTRL		CTCOOL					CTSTS	74h
								78h
								7Ch

								80h
								84h
								88h
								8Ch
								90h
								94h
								98h
								9Ch
								A0h
								A4h
								A8h
								ACh
								B0h
								B4h
								B8h
								BCh
								C0h
								C4h
								C8h
								CCh
								D0h
								D4h
								D8h
								DCh
								E0h
								E4h
								E8h
								ECh
								F0h
								F4h
								F8h
								FCh

Table 3-21. MCH Device 18 Function 0 I/OxAPIC Configuration Map

DID		VID		00h			80h
PCISTS		PCICMD		04h			84h
CCR			RID	08h			88h
	HDR		CLS	0Ch			8Ch
MBAR				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
				3Ch			BCh
		ABAR		40h			C0h
				44h			C4h
				48h			C8h
				4Ch			CCh
				50h			D0h
				54h			D4h
				58h			D8h
				5Ch			DCh
				60h			E0h
				64h			E4h
				68h			E8h
PMCAP				6Ch			ECh
PMCSR				70h			F0h
				74h			F4h
				78h			F8h
				7Ch			FCh

Table 3-22. IOxAPIC Direct Memory Mapped Registers

	INDEX	00h		80h
		04h		84h
		08h		88h
		0Ch		8Ch
WINDOW		10h		90h
		14h		94h
		18h		98h
		1Ch		9Ch
		20h		A0h
		24h		A4h
		28h		A8h
		2Ch		ACh
		30h		B0h
		34h		B4h
		38h		B8h
		3Ch		BCh
	EOI	40h		C0h
		44h		C4h
		48h		C8h
		4Ch		CCh
		50h		D0h
		54h		D4h
		58h		D8h
		5Ch		DCh
		60h		E0h
		64h		E4h
		68h		E8h
		6Ch		ECh
		70h		F0h
		74h		F4h
		78h		F8h
		7Ch		FCh

Table 3-23. Device 21, 22, Function 0: FB-DIMM DIMM Map, Control, RAS

DID		VID		00h	MTR1	MTR0	80h
PCISTS		PCICMD		04h	MTR3	MTR2	84h
CCR			RID	08h			88h
	HDR			0Ch			8Ch
				10h	DMIR0		90h
				14h	DMIR1		94h
				18h	DMIR2		98h
				1Ch	DMIR3		9Ch
				20h	DMIR4		A0h
				24h			A4h
				28h			A8h
SID		SVID		2Ch			ACh
				30h	DIMMPAIRCNT1	DIMMPAIRCNT0	B0h
				34h	DIMMPAIRCNT3	DIMMPAIRCNT2	B4h
				38h			B8h
				3Ch			BCh
SPCPS	SPCPC			40h	BADRAMA		C0h
FB-DIMMICMD1	FB-DIMMICMD0	FB-DIMMLVL1	FB-DIMMLVL0	44h	BADRAMB		C4h
FB-DIMMST		FB-DIMMCHCFG0		48h			C8h
FB-DIMMHPC		FB-DIMMCHCFG1		4Ch	BADCNTA		CCh
FB-DIMMRST				50h			D0h
				54h			D4h
FB-DIMMISTS1		FB-DIMMISTS0		58h			D8h
NBTRH0		NBTRL0		5Ch			DCh
NBTRH1		NBTRL1		60h	CERRCNTA0		E0h
AMBPRESENT1		AMBPRESENT0		64h	CERRCNTB0		E4h
				68h			E8h
				6Ch			ECh
				70h	CERRCNTA1		F0h
				74h	CERRCNTB1		F4h
				78h			F8h
				7Ch			FCh

Table 3-24. Device 21, 22, Function 1: Memory Branch Map, Control, Errors

DID	VID	00h			80h
		04h			84h
CCR	RID	08h			88h
HDR		0Ch			8Ch
		10h	NRECFB-DIMMIDLE		90h
		14h	RECFB-DIMMIDLE		94h
		18h			98h
		1Ch			9Ch
		20h			A0h
		24h			A4h
		28h			A8h
SID	SVID	2Ch			ACh
		30h			B0h
		34h			B4h
		38h			B8h
		3Ch	NRECMEMA		BCh
RECFB-DIMMRPA		40h	NRECMEMB		C0h
RECFB-DIMMRPB		44h	NRECFB-DIMMA		C4h
RECFB-DIMMRPC		48h	NRECFB-DIMMB		C8h
RECFB-DIMMRPD		4Ch	NRECFB-DIMMC		CCh
RECFB-DIMMRPE		50h	NRECFB-DIMMD		D0h
	RECFB-DIMMRPF	54h	NRECFB-DIMME		D4h
		58h	NRECFB-DIMMF		D8h
		5Ch	REDMEMA		DCh
		60h		RECMEMA	E0h
		64h	RECMEMB		E4h
		68h	RECFB-DIMMA		E8h
		6Ch	RECFB-DIMMB		ECh
VALIDLOG		70h	RECFB-DIMMC		F0h
NRECFGLOG		74h	RECFB-DIMMD		F4h
RECFGLOG		78h	RECFB-DIMME		F8h
REDMEMB		7Ch	RECFB-DIMMF		FCh

3.8 Register Definitions

3.8.1 PCI Standard Registers

These registers appear in every function for every device.

3.8.1.1 VID - Vendor Identification Register

The VID Register contains the vendor identification number. This 16-bit register, combined with the Device Identification Register uniquely identifies the manufacturer of the function with in the MCH. Writes to this register have no effect.

Device: 0, 1 - 9
Function: 0
Offset: 00h

Device: 16
Function: 0 - 4
Offset: 00h

Device: 17,18
Function: 0
Offset: 00h

Device: 21, 22
Function: 0 - 1
Offset: 00h

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

3.8.1.2 DID - Device Identification Register

This 16-bit register combined with the Vendor Identification register uniquely identifies the Function with in the MCH. Writes to this register have no effect. See Table 3-3 for the DID of each MCH function.

Device: 0, 1 - 9
Function: 0
Offset: 02h

Device: 15
Function: 0
Offset: 02h

Device: 16
Function: 0 - 2
Offset: 02h

Device: 17,18
Function: 0
Offset: 02h

Device: 21, 22
Function: 0 - 1
Offset: 02h

Bit	Attr	Default	Description
15:0	RWO	*See Table 3-3	Device Identification Number Identifies each function of the MCH

3.8.1.3 RID - Revision Identification Register

This register contains the revision number of the MCH. 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. Previously, a new value for RID was assigned for Intel chipsets for every stepping. There is a 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, for instance, to prevent Windows software 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 drivers updates and minimize changes to the OS image for minor optimizations to the silicon for yield improvement, or feature enhancement reasons that do not negatively impact the OS driver functionality.

By default the SRID is returned when the RID is read at offset 08h. The SRID value reflects the actual product stepping.

Device: 0, 1 - 9
Function: 0
Offset: 08h

Device: 16
Function: 0 - 4
Offset: 08h

Device: 17, 18
Function: 0
Offset: 08h

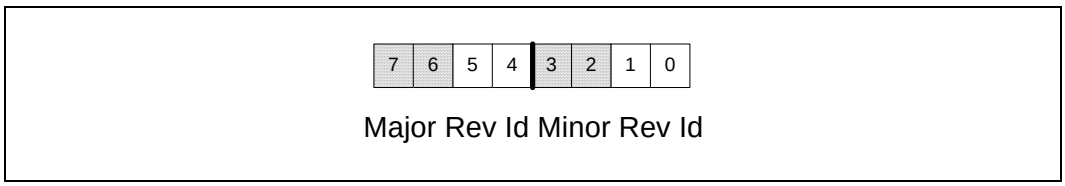
Device: 21, 22
Function: 0 - 1
Offset: 08h

Bit	Attr	Default	Description
7:4	RO	0h	Major Revision Steppings which require all masks to be regenerated. 0000: A stepping for SG with SF 0001: B stepping for SG with SF 0010: C stepping for SG with SF Others: *Reserved*
3:0	RO	0h	Minor Revision Incremented for each stepping which does not modify all masks. Reset for each major revision. 0x0: M0 stepping 0x1: M1 stepping Others: *Reserved* Note: The Metal steppings indicated are a subset of the Major revision. For e.g. an A stepping with M0 as minor revision typically means A0.

3.8.1.3.1 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 3-3.

Figure 3-3. MCH implementation of SRID Registers



3.8.1.4 CCR - Class Code Register

This register contains the Class Code for the device. Writes to this register have no effect. The CCR for the Intel® QuickData Technology, device 15 and the IOxAPIC, device 18 are defined separately in there respective sections.

Device: 0, 1 - 9
Function: 0
Offset: 09h

Device: 16
Function: 0 - 4
Offset: 09h

Device: 17
Function: 0
Offset: 09h

Device: 21, 22
Function: 0 - 1
Offset: 09h

Bit	Attr	Default	Description
23:16	RO	06h	**Base Class**. This field indicates the general device category. For the MCH, this field is hardwired to 06h, indicating it is a "Bridge Device".
15:8	RO	if (DEV1-9) {04h} else {00h}	Sub-Class. This field qualifies the Base Class, providing a more detailed specification of the device function. For PCI Express Devices 1 - 9 default is 04h, indicating "PCI to PCI Bridge" For all other Devices: 0, 13, 14, 16, 17, 21, 22 default is 00h, indicating "Host Bridge".
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.

3.8.1.5 HDR - Header Type Register

This register identifies the header layout of the configuration space.

Device: 0, 1 - 9
Function: 0
Offset: 0Eh

Device: 16
Function: 0
Offset: 0Eh

Device: 17,18
Function: 0
Offset: 0Eh

Device: 21, 22
Function: 0 - 1
Offset: 0Eh

Bit	Attr	Default	Description
7	RO	if(DEV16) {1h} elsif DEV21-22){1h} else{0h} endif	Multi-function Device. Selects whether this is a multi-function device, that may have alternative configuration layouts. This bit is hardwired to '0' for devices for the MCH with the exception of devices 16,21 and 22 which it are set to '1'.
6:0	RO	if (DEV1-9) {01h} else {00h} endif	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 -9 default is 01h, indicating "PCI to PCI Bridge" For all other Devices: 0, 13 - 18,21,22 default is 00h, indicating a conventional type 00h PCI header

3.8.1.6 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.

Device: 0
Function: 0
Offset: 2Ch

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

Device: 17,18
Function: 0
Offset: 2Ch

Device: 21, 22
Function: 0 - 1
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 the MCH will write to all of them.

3.8.1.7 SID - Subsystem Identity

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

Device: 0
Function: 0
Offset: 2Eh

Device: 15
Function: 0
Offset: 2Eh

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

Device: 17,18
Function: 0
Offset: 2Eh

Device: 21, 22
Function: 0 - 1
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.

3.8.2 Address Mapping Registers

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

Table 3-29. 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 - 9) PMBASE/PMLIM (devices 1 - 9) PMBU/PMBL (devices 1 - 9) IOBASE/IOLIM (devices 1 - 9) SBUSN,SUBUSN (devices 1 - 9) BCTRL, HECBASE, PCICMD (devices 1 - 9)
ESI	Subtractive decode[1] (device 0)

Notes:

1. 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 MCH allows programmable memory attributes on 13 Legacy memory segments of various sizes in the 640 Kilobytes to 1 Megabytes address range. Seven Programmable Attribute Map (PAM) Registers are used to support these features.

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

3.8.2.1 PAM0 - Programmable Attribute Map Register 0

This register controls the read, write, and shadowing attributes of the BIOS area which extends from 0F 0000h - 0F FFFFh.

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 MCH 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 MCH 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

3.8.2.2 PAM1 - Programmable Attribute Map Register 1

This register controls the read, write, and shadowing attributes of the BIOS areas which extend from 0C 0000h-0C 7FFFh.

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
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

3.8.2.3 PAM2 - Programmable Attribute Map Register 2

This register controls the read, write, and shadowing attributes of the BIOS areas which extend from 0C 8000h -0C FFFF h.

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

3.8.2.4 PAM3 - Programmable Attribute Map Register 3

This register controls the read, write, and shadowing attributes of the BIOS areas which extend from 0D 0000h - 0D 7FFFh.

Device: 16
Function: 0
Offset: 5Ch

Bit	Attr	Default	Description
7:6	RV	00	Reserved
5:4	RW	00	**ESIENABLE3: 0D 4000h - 0D 7FFFh Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0D 4000h -0D 7FFFh. 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: 0D 0000h - 0D 3FFFh Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0D 0000h -0D 3FFFh. 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

3.8.2.5 PAM4 - Programmable Attribute Map Registers 4

This register controls the read, write, and shadowing attributes of the BIOS areas which extend from 0D 8000h - 0D FFFFh.

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

3.8.2.6 PAM5 - Programmable Attribute Map Register 5

This register controls the read, write, and shadowing attributes of the BIOS areas which extend from 0E 0000h -0E 7FFFh.

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

3.8.2.7 PAM6 - Programmable Attribute Map Register 6

This register controls the read, write, and shadowing attributes of the BIOS areas which extend from 0E 8000h -0E FFFFh.

Device: 16
Function: 0
Offset: 5Fh

Bit	Attr	Default	Description
7:6	RV	00	Reserved
5:4	RW	00	**ESIENABLE6: 0EC000-0DFFFF Attribute Register** This field controls the steering of read and write cycles that address the BIOS area from 0EC000-0DFFFF. 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	**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

3.8.2.8 SMRAMC - System Management RAM Control Register

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 MCH supports only the SMM space between A 0000h and B FFFFh, this field is hardwired to 010.

3.8.2.9 EXSMRC - Extended System Management RAM Control Register

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	RV	0	Reserved
6	RO	0	**MDAP: MDA Present** Since the MCH does not support MDA, this bit has no meaning.
5	RW	0	APICDIS: APIC Memory Range Disable When set to '1', the MCH forwards accesses to the IOAPIC regions to the appropriate interface, as specified by the memory and PCI configuration registers. When this bit is clear, the MCH will send cycles between 0_FEC0_0000 and 0_FEC7_FFFF to ESI, accesses between 0_FEC8_0000 and 0_FEC8_0FFF will be sent to PEX1, between 0_FEC8_1000 and 0_FEC8_1FFF will be sent to PEX2, between 0_FEC8_2000 and 0_FEC8_2FFF will be sent to PEX3, between 0_FEC8_3000 and 0_FEC8_3FFF will be sent to PEX4, between 0_FEC8_4000 and 0_FEC8_4FFF will be sent to PEX5, between 0_FEC8_5000 and 0_FEC8_5FFF will be sent to PEX6, between 0_FEC8_6000 and 0_FEC8_6FFF will be sent to PEX7, between 0_FEC8_7000 and 0_FEC8_7FFF will be sent to PEX8, between 0_FEC8_8000 and 0_FEC8_8FFF will be sent to PEX9
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: 512 kB 01: 1 MB 10: 2 MB 11: 4 MB
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.

3.8.2.10 EXSMRTOP - Extended System Management RAM Top Register

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 4.3.3: ESMMTOP-TSEG_SZ <= Address < ESMMTOP TSEG_SZ can be 512 KB, 1 MB, 2 MB, or 4 MB, 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. This field defaults to point to the same address as 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.

3.8.2.11 EXSMRAMC - Expansion System Management RAM Control Register

Device: 16
Function: 2
Offset: 60h

Bit	Attr	Default	Description
7	RW1C	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. The MCH will set this bit if any In-Bound access from I/O device targeting SMM range that gets routed to the ESI port (master abort). Refer to Section 4.4.3 for details. The MCH 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.

3.8.2.12 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 256 MB range of addresses resides for this particular host bridge. This register defaults to the same address as the default value for TOLM.
11:10	RV	0	Reserved.
9:3	RV	00h	Reserved
2:1	RW	00	**LENGTH:** This field describes the length of this region - Extended Configuration MMCFG Space Region/Buses Decoded 00: 256 MB (Buses 0-255). Bits 23:12 are decoded in the PCI Express MMCFG Base Address field. 01: 128 MB (Buses 0-127). Bits 23:11 are decoded in the PCI Express MMCFG Base Address field. 10: 64 MB (Buses 0-63). Bits 23:10 are decoded in the PCI Express MMCFG Base Address field. 11: Reserved. Behavior is undefined if set to this value.
0	RW	1	**HECBASEEN: HECBASE Enable** 0: This register is disabled. Memory reads and writes transactions proceed as if there were no such register. Bits 31:1 of this register are R/W with no functionality behind them. 1: This register is enabled. Memory reads and writes transactions whose address bits 39:28 match bits 23:12 (plus extra bit(s) per Length field [bits 2:1] setting) will be translated to PCI Express configuration reads and writes within MCH.

3.8.3 AMB Memory Mapped Registers

The MCH supports four FB-DIMM channels. The MCH supports up to 16 FB-DIMM DIMMs (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 functions per AMB component with 256B of register space per function.

The MCH supports memory mapped register regions for software to access individual AMB configuration registers. Memory mapped access to AMB register regions are converted by the MCH to FB-DIMM channel command encodings subject to AMBPRESENT register settings (see Section 3.9.23.12). This region is relocatable by programming the AMBASE register. Software is required to program the AMR for the size of AMB register regions. The size of this region is 128 KB. It is mapped to each AMB addressing slot in 2 KB blocks. If the corresponding AMBPRESENT bit is not set, then MCH will not send configuration transaction to that AMB addressing slot.

To support SMBus and JTAG access using traditional PCI configuration mechanism, the MCH 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 23, function 0, is mapped to the AMB's configuration registers selected by AMBSELECT register (see Section 3.8.3.3, "AMBSELECT: AMB Switching Window Select Register").

Access to bus 0, device 23, function 0, is limited to SMBus and JTAG only, accesses from other sources such as FSB to this function will be mastered aborted by the MCH as non-existent PCI function.

3.8.3.1 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 MCH.
39:17	RW	007F00h	**AMBASE:** This marks the 128 KB 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 MCH uses only 38-bit addressable space, hence only bits 39:17 are valid. The default base address is at: 0xFE00_0000. This field could be relocated by software.
16:0	RV	0h	Reserved

3.8.3.2 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 MCH, the value is 128 KB: 2 KB per AMB for a total of 16 AMB per channel, 32 KB per FB-DIMM channel for a total of four channels.

3.8.3.3 AMBSELECT: AMB Switching Window Select Register

When SMBus/JTAG is accessing bus 0, device 23, 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 FB-DIMM channel being accessed via bus 0, device 23, function 0 for SMBus and JTAG only.
6:3	RW	0h	**AMB_Select:** Specify the FB-DIMM channel being accessed via bus 0, device 23, function 0 for SMBus and JTAG only
2:0	RW	0h	**Function_Select:** Specify the FB-DIMM channel being accessed via bus 0, device 23, function 0 for SMBus and JTAG only

3.8.3.4 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 FB-DIMM channels that MCH supports.This register is the same for all different MCH SKUs. Software should read MCH's DID to determine the actual MCH platform SKU, e.g. there might be only two channels for MCHVS SKU. MIRs needs to be programmed appropriately.

3.8.3.5 MAXDIMMPERCH - Maximum DIMM PER Channel Number Register

This register controls the maximum number of AMB DIMMs per FB-DIMM channel that MCH supports for AMB configuration register access. This register applies only to DIMM modules in the FB-DIMM 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	04h	Maximum_number_DIMM_per_channel: Set by hardware to indicate the maximum number of FB-DIMM DIMM AMBs per channel that the MCH supports.

3.8.3.6 Map to AMB Registers

In Table 3-30, each 2 KB range is mapped to individual AMB registers by address translation of MCH. The address of this relocatable register area is specified in the AMBASE register. Configuration transactions targeting these ranges are converted to FB-DIMM commands by the MCH and sent to the FB-DIMM channel subject to AMBPRESENT register settings.

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

Table 3-30. Register Offsets in AMB Memory Mapped Registers Region (Sheet 1 of 2)

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

Table 3-30. Register Offsets in AMB Memory Mapped Registers Region (Sheet 2 of 2)

7FFh-0h	map to channel_0, AMB_0 registers
147FFh-14000h	map to channel_2, AMB_8 registers
14FFFh-14800h	map to channel_2, AMB_9 registers
157FFh-15000h	map to channel_2, AMB_A registers
15FFFh-15800h	map to channel_2, AMB_B registers
167FFh-16000h	map to channel_2, AMB_C registers
16FFFh-16800h	map to channel_2, AMB_D registers
177FFh-17000h	map to channel_2, AMB_E registers
17FFFh-17800h	map to channel_2, AMB_F registers
187FFh-18000h	map to channel_3, AMB_0 registers
18FFFh-18800h	map to channel_3, AMB_1 registers
197FFh-19000h	map to channel_3, AMB_2 registers
19FFFh-19800h	map to channel_3, AMB_3 registers
1A7FFh-1A000h	map to channel_3, AMB_4 registers
1AFFFh-1A800h	map to channel_3, AMB_5 registers
1B7FFh-1B000h	map to channel_3, AMB_6 registers
1BFFFh-1B800h	map to channel_3, AMB_7 registers
1C7FFh-1C000h	map to channel_3, AMB_8 registers
1CFFFh-1C800h	map to channel_3, AMB_9 registers
1D7FFh-1D000h	map to channel_3, AMB_A registers
1DFFFh-1D800h	map to channel_3, AMB_B registers
1E7FFh-1E000h	map to channel_3, AMB_C registers
1EFFFh-1E800h	map to channel_3, AMB_D registers
1F7FFh-1F000h	map to channel_3, AMB_E registers
1FFFFh-1F800h	map to channel_3, AMB_F registers

3.8.4 Interrupt Redirection Registers

3.8.4.1 REDIRCTL - Redirection Control Register

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

Device: 16
Function: 0
Offset: 6Eh

Bit	Attr	Default	Description
15:14	RV	0	**Reserved**
13	RV	0	Reserved
12	RV	0	**Reserved**
11:8	RW	0h	BUCKET2: First priority number not in BUCKET0, BUCKET1, or BUCKET2. Must be programmed with a larger value than BUCKET1. A suggested value is Ch.
7:4	RW	0h	BUCKET1: First priority number not in BUCKET0 or BUCKET1. Must be programmed with a larger value than **BUCKET0.** A suggested value is 8h.
3:0	RW	0h	**BUCKET0:** First priority number not in **BUCKET0**. A suggested value is 0h.

3.8.4.2 REDIRBUCKETS - Redirection Bucket Number Register

This register allows software to read the current hardware bucket number assigned to each XTPR register

Device: 16 Function: 0 Offset: 68h			
Bit	**Attr**	**Default**	**Description**
31:30	RO	0	**BUCKET15:** Redirection bucket number for XTPR[15].
29:28	RO	0	**BUCKET14:** Redirection bucket number for XTPR[14].
27:26	RO	0	**BUCKET13:** Redirection bucket number for XTPR[13].
25:24	RO	0	**BUCKET12:** Redirection bucket number for XTPR[12].
23:22	RO	0	**BUCKET11:** Redirection bucket number for XTPR[11].
21:20	RO	0	**BUCKET10:** Redirection bucket number for XTPR[10].
19:18	RO	0	**BUCKET9:** Redirection bucket number for XTPR[9].
17:16	RO	0	**BUCKET8:** Redirection bucket number for XTPR[8].
15:14	RO	0	**BUCKET7:** Redirection bucket number for XTPR[7].
13:12	RO	0	**BUCKET6:** Redirection bucket number for XTPR[6].
11:10	RO	0	**BUCKET5:** Redirection bucket number for XTPR[5].
9:8	RO	0	**BUCKET4:** Redirection bucket number for XTPR[4].
7:6	RO	0	**BUCKET3:** Redirection bucket number for XTPR[3].
5:4	RO	0	**BUCKET2:** Redirection bucket number for XTPR[2].
3:2	RO	0	**BUCKET1:** Redirection bucket number for XTPR[1].
1:0	RO	0	**BUCKET0:** Redirection bucket number for XTPR[0].

3.8.5 Boot and Reset Registers

3.8.5.1 SYRE - System Reset Register

This register controls MCH 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, ESI, processor bus).

There is no “SOFT RESET” bit in this register. That function is invoked through the ESI. There are no CORE:FB-DIMM gear ratio definitions in this register. Those are located in the DDRFRQ register.

Device: 16 Function: 0 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
15	RV	0	Reserved

Device: 16 Function: 0 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
14	RW	0	**CPURST: Processor Reset** If set, the MCH will assert processor RESET# on both buses as soon as the MCH has no pending transactions. The chipset will then deassert RESET# following the timing rules described in the Reset Chapter. The MCH does not have any mechanism to drain transactions before effecting the CPU RESET#. It is the responsibility of software to ensure that the system is quiet before sending the configuration write (last command) to set this field in the MCH in order to drive the CPU RESET# signal. Any violation of this usage pattern would render the system unstable and potentially catastrophic.
13	RV	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.
12:11	RV	0	Reserved1
10	ROS	0	**S3: S3 Sleep State** The MCH sets this bit when it sends an Ack-S3 message to the ESI port. The MCH clears this bit after it has placed appropriate FB-DIMM channels into the FB-DIMM.Calibrate state in response to deassertion of the RESETI# signal.
9	RW	0	**ROR: Processor Reset on Refresh** If set, the MCH will assert processor RESET# on both busses when a refresh cycle completes.
8	RWS	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

3.8.5.2 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
11:0	RWS	7FFh	**DCRT: ESI CPU Reset Done Ack** Determinism Timer This field provides the determinism timer threshold for the Intel® 5400 MCH 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® 5400 MCH. Maximum value is 4095 core clocks A default of 2047 clocks (7FFh) is used.

3.8.5.3 POC - Power-On Configuration Register

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.

The MCH drives the contents of this register on A[35:4]_N 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 POC bits do not affect MCH operation except for driving A[35:4]_N.

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

There are other power-on configuration bits in the SYRE register.

Device: 16
Function: 0
Offset: 44h

Bit	Attr	Default	Description
31:28	RV	0h	Reserved
27	RV	0	Reserved
26:12	RV	0h	Reserved

Device: 16 Function: 0 Offset: 44h			
Bit	**Attr**	**Default**	**Description**
11	RV	0h	Reserved
10:0	RV	0h	Reserved

3.8.5.4 SPAD[3:0] - Scratch Pad Registers

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.

3.8.5.5 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	RWS	00000000h	Scratch Pad value. These sticky bits have no effect on the hardware.

3.8.5.6 BOFL[3:0] - Boot Flag Register

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 that gets a non-zero value 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 I/O addresses.

Device: 16 Function: 0 Offset: C0h, C4h, C8h, CCh			
Bit	**Attr**	**Default**	**Description**
31:0	RCW	A5A5A5A5h	**SemaVal: Semaphore Value** Can be written to any value. Value is cleared when there is a read.

3.8.6 Control and Interrupt Registers

3.8.6.1 PROCENABLE: Processor Enable Global Control

The two FSBEN bits are used to enable or disable frontside bus arbitration. When frontside bus arbitration is disabled the processor is effectively disabled.

Device: 16 Function: 0 Offset: F0h			
Bit	**Attr**	**Default**	**Description**
31:5	RV	0f8h	*Reserved.*
4:3	RWS	11	FSBEN: FSB1 and FSB0 Enable The field is defined as the following: 00: reserved 01: FSB1 is disabled. FSB0 is enabled. 10: FSB1 is enabled. FSB0 is disabled. 11: FSB1 is enabled. FSB0 is enabled. (default) Hard-reset is needed after changing value in this register.
2	RWS	0	Snoop Filter Bypass 0: SF is enabled 1: SF is disabled Note: A hardcoded disabled default prevents writes to this registers from changing abilities.
1:0	RV	0h	*Reserved.*

3.8.6.2 FSBS[1:0] - Processor Bus Status Register

This register holds status from the Processor Busses.

Device: 16 Function: 0 Offset: 7Ch, 74h			
Bit	**Attr**	**Default**	**Description**
31:2	RV	0h	*Reserved.*
1	RO	0	2SOCKET: 2 Sockets present on this FSB Set when Intel® 5400 MCH has seen Ab[22] asserted, indicating there are more than 1 processors present on this FSB.
0	RO	0	2CORE: 2 Cores present Set when Intel® 5400 MCH has seen Ab[30] asserted, indicating there is more than 1 core in a processor. Note: Mixing single core with dual-core processors will be recognized as dual-core processor on this FSB.

3.8.6.3 XTPR[7:0] - External Task Priority Register

These registers control redirectable interrupt priority for xAPIC agents connected to the MCH. Up to four agents on each bus are supported. These agents may be two dual core processors each with two threads or four single core processors. 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.

Each register contains the following fields:

1. Agent priority (Task Priority)
2. APIC enable bit (TPR Enable)
3. Logical APIC ID (LOGID)
4. Processor physical APIC ID (PHYSID)

The XTPR registers are modified by a front side bus xTPR_Update transaction. In addition, the XTPR registers can be modified by software.

Table 3-31. XTPR Index

Index	Value
3	0 for FSB0, 1 for FSB1
2	Ab[29]
1	Ab[30] OR Ab[22]
0	Ab[21]

These registers are used for lowest priority delivery through interrupt redirection by the chipset. Whenever this register is updated, the "CLUSTER" bit in the register is also updated.

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

Bit	Attr	Default	Description
31	if (XTPR0) {RW} else {RV} endif	0	**CLUSTER: Global Cluster Mode (XTPR[0] only)** Used in interrupt redirection for lowest priority delivery. Updated by every xTPR_Update transaction on either bus (Aa[3]). 0: flat 1: cluster
30:24	RV	00h	*Reserved.*
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.
22:20	RV	0h	*Reserved.*
19:16	RW	0h	**PRIORITY: Task Priority** The processor with the lowest enabled value will be assigned the redirectable interrupt. This field is updated with Ab[27:24] of the xTPR_Update transaction.
15:8	RW	0h	**PHYSID: Physical APIC ID** The physical ID of the APIC agent associated with the XTPR 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 XTPR entry. This field is updated with Aa[11:4] of the xTPR_Update transaction.

3.8.7 PCI Express Device Configuration Registers

This section describes the registers associated with the PCI Express Interface.

The PCI Express register structure is exposed to the operating system and requires a separate device per port. Ports 1-9 will be assigned devices 1 through 9 while Port 0 is the ESI interconnect to the Intel 631xESB/632xESB I/O Controller Hub. The PCI Express ports determine at reset the maximum width of the devices to which they are connected through link training. All ports will be made visible to OS even if unconnected. If Ports are combined to form larger widths (e.g. x8 or x16 from a x4 link), then the unused ports will Master Abort (reads return all ones, writes dropped) any accesses to it.

Table 3-32. When will a Intel® 5400 Chipset PCI Express Device be Accessible?

PCI Express Port	Device	x16	x8	Registers may be accessed if:
9	9	Dedicated x4 Not combinable		Port 9 is connected to a x4 Intel 631xESB/632xESB I/O Controller Hub port and cannot be combined with any other device.
8	8	High Performance Graphics Port	Possible combination	Port 8 is connected to a x4 device.
7	7			Port 7 is connected to a x4 or x8 device.
6	6		Possible combination	Port 6 is connected to a x4 device.
5	5			Port 5 is connected to a x4 or x8 or x16 device.
4	4	High Performance Graphics Port	Possible combination	Port 4 is connected to a x4 device.
3	3			Port 3 is connected to a x4 or x8 device.
2	2		Possible combination	Port 2 is connected to a x4 device.
1	1			Port 1 is connected to a x4 or x8 or x16 device.
0	0	ESI - Not combinable		Port0 is connected to a x4 Intel 631xESB/632xESB I/O Controller Hub port through ESI and cannot be combined with any other port.

Figure 3-4, "PCI Express Configuration Space" 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 the Intel® 5400 MCH the supported capabilities are:
  - — Message Signalled Interrupts
  - — 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 MCH 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 MCH, working register sets exist for the Standard Controller and Power Management capabilities.

Figure 3-4. PCI Express Configuration Space

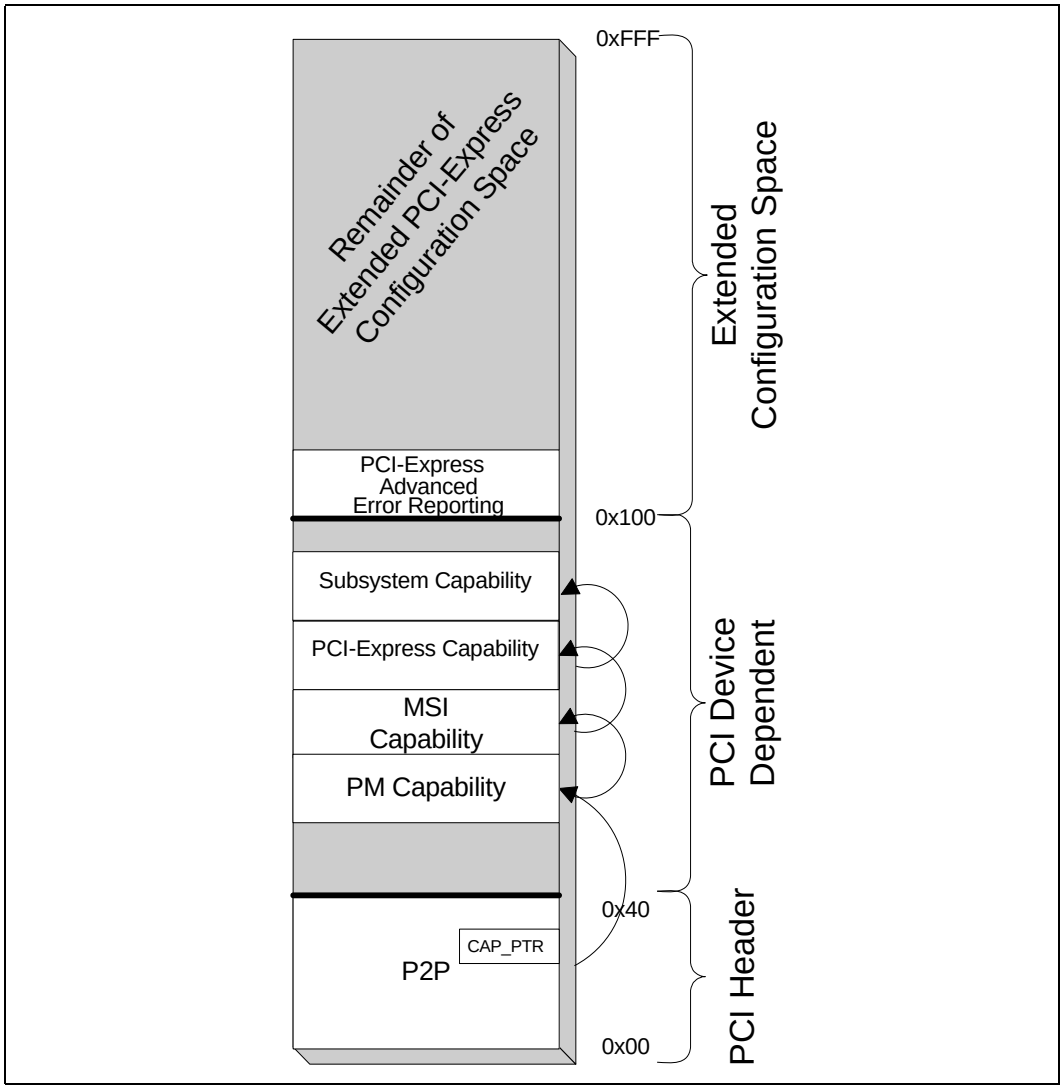


Figure 3-4 shows the configuration register offset addresses for each of the PCI Express ports as defined in the *PCI-Express Base Specification*, Revision 2.0. It is also compatible with the standard PCI 2.3 capability structure and comprises of a linked list where each capability has a pointer to the next capability in the list. For PCI Express extended capabilities, the first structure is required to start at 0x100 offset.

3.8.8 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 MCH. Unless otherwise specified, the registers are enumerated as a vector [9:1] mapping to each of the nine PCI Express ports uniquely while the ESI port is referred by index 0.

3.8.8.1 PCICMD[9:0]- Command Register

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

Device: 9 - 0 Function: 0 Offset: 04h			
Bit	**Attr**	**Default**	**Description**
15:11	RV	0h	*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® 5400 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® 5400 MCH. Refer to the INTP register in Section 3.8.8.27, "INTP[9:0] - Interrupt Pin Register" on page 111 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** his field handles the reporting of fatal and non-fatal errors by enabling the error pins ERR[2:0]. 1: The Intel® 5400 chipset is enabled to send fatal/non-fatal errors. 0: The Intel® 5400 chipset is disabled from generating fatal/non-fatal errors. The errors are also enabled by the PEXDEVCTRL register in Section 3.8.11.4 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[1] forwarded from the PCI Express interface. This bit does not affect the transmission of forwarded ERR_COR messages. Refer to the Intel® 5400 chipset MCH 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: 9 - 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. The MCH 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 When the BME is disabled, the MCH 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. 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. This corner case is handled by the BME filtration in the Intel® 5400 chipset MCH's PCI Express port using the above rules since BME is reset. However, in general, it is illegal for a an I/O device to issue inbound requests until the CPURESET# has been deasserted to prevent any possible malfunction in the Intel® 5400 chipset MCH logic.
1	if (port 9-1) {RW} elseif (port 0) {RO} endif	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 non-prefetchable and prefetchable memory ranges (high/low) or targets a memory-mapped location within the bridge itself 1: Enables the Memory and Prefetchable 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 (MCH). 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	if (port 9-1) {RW} elseif (port 0) {RO} endif	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:

1. 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 3.8.8.28

3.8.8.2 PCISTS[9:0] - Status Register

The PCISTS 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 the MCH.

Device: 9 - 0
Function: 0
Offset: 06h

Bit	Attr	Default	Description
15	RW1C	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 (inbound/outbound direction). The detected parity error maps to B1, F6, M2 and M4 (uncorrectable data error from FSB, Memory or internal sources) of the Intel® 5400 chipset MCH.
14	RW1C	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	RW1C	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: MCH never sets to 1. The MCH root port is never a requester except for interrupts. (MSIs) generated for errors and HP/PM/linkstatus events. MSIs are not expected to receive a UR/CA response. 0: No Master Abort is generated
12	RW1C	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. 1: MCH never sets to 1. The MCH root port is never a requester except for interrupts (MSIs) generated for errors and HP/PM/linkstatus events. MSIs are not expected to receive a UR/CA response. 0: No Target Abort is generated
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	RW1C	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 if the root port generates a poisoned request or responds with a poisoned completions. The MCH root port will never set this bit. The root port is never the requester of a request.
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)

Device: 9 - 0 Function: 0 Offset: 06h			
Bit	**Attr**	**Default**	**Description**
3	RO	0	**INTxSTAT: INTx Status** Indicates that an INTx interrupt message is pending internally in the PCI Express port. The INTx status bit should be rescinded when all the relevant events via RAS errors/HP/PM internal to the port that requires legacy interrupts are cleared by software.
2:0	RV	0h	*Reserved.* (by PCI SIG)

3.8.8.3 CLS[9:0] - 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 the MCH.

Device: 9 - 0 Function: 0 Offset: 0Ch			
Bit	**Attr**	**Default**	**Description**
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 MCH.

3.8.8.4 PRI_LT[9:0] - Primary Latency Timer

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

Device: 9 - 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.

3.8.8.5 BIST[9:0] - 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 the Intel® 5400 chipset MCH.

Device: 9 - 0 Function: 0 Offset: 0Fh			
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	BIST_TST: BIST Tests Not supported. Hardwired to 00h

3.8.8.6 BAR0[9:0] - 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 MCH. The base address register 0 is not supported/defined in the PCI Express port of the MCH.

3.8.8.7 BAR1[9:0] - Base Address Register 1

The base address register 1 is not supported/defined in the MCH.

3.8.8.8 EXP_ROM[0]: Expansion ROM Registers

The ESI port (device 0, function 0) does not implement any Base address registers in the Intel® 5400 chipset MCH 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.

3.8.8.9 PBUSN[9:1] - Primary Bus Number

This register identifies the bus number on the on the primary side (MCH) of the PCI Express port.

Device: 9 - 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 consistently 0, these bits are read only and are hardwired to 0.

3.8.8.10 SBUSN[9:1] - 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: 9 - 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 busses connected to PCI Express. Since both bus 0, device 1 and the PCI to PCI bridge on the other end are considered by configuration software to be PCI-PCI bridges, this bus number will consistently correspond to the bus number assigned to the PCI Express port

3.8.8.11 SUBUSN[9:1] - 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: 9 - 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.

3.8.8.12 SEC_LT[9:1] - Secondary Latency Timer

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

Device: 9 - 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.

3.8.8.13 IOBASE[9:1] - I/O Base Register

The I/O Base and I/O Limit registers (see Section 3.8.8.14) 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 4 KB boundary while the top of the region specified by IO_LIMIT will be one less than a 4 KB multiple. Refer to Section 4.5.1 and Section 4.5.3 in the Platform Specification.

Device: 9 - 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 MCH does not support 32 bit addressing, so these bits are hardwired to 0.

3.8.8.14 IOLIM[9:1] - 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 of this register are programmable. For the purpose of address decode, address bits A[11:0] of the I/O limit register is treated as FFFh.

Device: 9 - 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 MCH does not support 32 bit I/O addressing, so these bits are hardwired to 0.

3.8.8.15 SECSTS[9:1] - 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 the MCH.

Device: 9 - 1
Function: 0
Offset: 1Eh

Bit	Attr	Default	Description
15	RW1C	0	SDPE: Detected Parity Error This bit is set by the Intel® 5400 chipset MCH 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)**. **BCTRL.PRSPEN register)**. This corresponds to IO4 as defined in Table 5-36, "Intel® 5400 Chipset Error List" on page 393 Refer to Table 3-33, "MCH PCISTS and SECSTS Master/Data Parity Error RAS Handling".
14	RW1C	0	SRSE: Received System Error This bit is set by the MCH when it receives a ERR_FATAL or ERR_NONFATAL message. Section 3.8.8.28. (Note that BCTRL.BCSERRE is not a gating item for the recording of this error on the secondary side). This corresponds but is not limited to IO11 and IO1 as defined in Table 5-36, "Intel® 5400 Chipset Error List" on page 393Refer to Table 3-33, "MCH PCISTS and SECSTS Master/Data Parity Error RAS Handling".
13	RW1C	0	S**RMAS: Received Master Abort Status** This bit is set when the Secondary Side for Type 1 Configuration Space header Device (for requests initiated by the Type 1 header device itself) receives a Completion with Unsupported Request Completion Status. Note: The MCH root port device is never a requester of an outbound request.
12	RW1C	0	SRTAS: Received Target Abort Status This bit is set when the Secondary Side for Type 1 Configuration Space header device (for request initiated by the Type 1 header device itself) receives a Completion with Unsupported Request Completion Status. Note: The MCH root port device is never a requester of an outbound request.

Device: 9 - 1 Function: 0 Offset: 1Eh			
Bit	**Attr**	**Default**	**Description**
11	RW1C	0	**SSTAS: Signaled Target Abort** This bit is set when the Secondary Side for Type 1 Configuration Space header device (for requests initiated by the Type 1 header device itself) completes a Request using Completer Abort Completion Status. Note: MCH root port device will never respond to a requests with a Completer Abort Completion Status.
10:9	RO	00	SDEVT: DEVSEL# Timing Not applicable to PCI Express. Hardwired to 0
8	RW1C	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 3.8.8.28 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 3-33 for details on the data parity error handling matrix in the Intel® 5400 chipset MCH. The MCH root port will never set this bit. The root port is never the requester of a request.
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 3-33. MCH PCISTS and SECSTS Master/Data Parity Error RAS Handling

Register Name	OB Transaction with Data (e.g. outbound Memory Writes)	OB Compl with Data (e.g. Inbound Memory/IO NP requests)	IB Transaction with Data (e.g. inbound memory writes)	IB Compl with Data (e.g. outbound Memory/IO NP requests)
PCISTS[15].DPE	yes	yes	no	no
PCISTS[8].MDPERR (gated by PCICMD.PERRE)	no	no	no	no
SECSTS[15].SDPE	no	no	yes	yes
SECSTS[8].SMDPERR (gated by BCTRL.PRSPEN)	no	no	no	no

3.8.8.16 MBASE[9:1] - Memory Base

The Memory Base and Memory Limit registers define a memory mapped I/O non-prefetchable address range (32-bit addresses) and the MCH 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 4.3.9, Section 4.4.2 and Section 4.4.3 for further details on address mapping.

Device: 9 - 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)

3.8.8.17 MLIM[9:1]: 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 MCH hardware to enforce prevention of overlap and operations of the system in the case of overlap are not guaranteed.

Device: 9 - 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)

3.8.8.18 PMBASE[9:1] - 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 F FFFFh. 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: 9 - 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.

3.8.8.19 PMLIM[9:1] - 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 F FFFFh.

Device: 9 - 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.

3.8.8.20 PMBU[9:1] - 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: 9 - 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.

3.8.8.21 PMLU[9:1] - Prefetchable Memory Limit (Upper 32 bits)

Device: 9 - 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.

3.8.8.22 IOB[9:1] - I/O Base Register (Upper 16 bits)

Not used since MCH does not support upper 16-bit I/O addressing.

3.8.8.23 IOL[9:1] - I/O Limit Register (Upper 16 bits)

Not used since MCH does not support upper 16-bit I/O addressing.

3.8.8.24 CAPPTR[9:0]- 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: 9 - 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

3.8.8.25 RBAR[9:1] - ROM Base Address Register

Not implemented in MCH, since the MCH is a virtual PCI-PCI bridge.

3.8.8.26 INTL[9:0] - Interrupt Line Register

The Interrupt Line register is used to communicate interrupt line routing information between the initialization code and the device driver. The MCH does not have a dedicated interrupt line. This register RO and is provided for backwards compatibility.

Device: 9 - 0 Function: 0 Offset: 3Ch			
Bit	**Attr**	**Default**	**Description**
7:0	RO	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 MCH since the PCI Express port does not have interrupt lines.

3.8.8.27 INTP[9:0] - 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: 9 - 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.

3.8.8.28 BCTRL[9:1] - Bridge Control Register

This register provides extensions to the PCICMD 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 MCH, e.g. VGA compatible address range mapping.

Device: 9 - 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.

Device: 9 - 1
Function: 0
Offset: 3Eh

Bit	Attr	Default	Description
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. When this SBUSRESET bit is cleared after the MCH enters the "hot-reset" state, MCH 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, MCH 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.
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 1 KB 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	I**SAEN: ISA Enable** Modifies the response by the Intel® 5400 chipset MCH 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® 5400 chipset MCH will not forward to PCI Express any I/O transactions addressing the last 768 bytes in each 1 KB block even if the addresses are within the range defined by the IOBASE and IOLIM registers. See Section 4.5.2. 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.

Device: 9 - 1 Function: 0 Offset: 3Eh			
Bit	**Attr**	**Default**	**Description**
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 MCH does not ever set SECSTS.MDPE status bit

3.8.8.29 IOAPICCTRL: IOAPIC Control Register

This register provides control for the interrupt handling behavior of the IOxAPIC, as well as the visibility of the integrated IOxAPIC PCI configuration space.

Device: 0 Function: 0 Offset: 46h			
Bit	**Attr**	**Default**	**Description**
7:3	RV	0h	***Reserved***.
2	RW	0	UNLABARA: **Unlock ABAR Address:** 1: When this bit is set, the X/Y/Z fields of the ABAR register are writable. 0: When this bit is clear, the X/Y/Z fields of the ABAR register are read only.
1	RV	0	Reserved
0	RW	0	DISIROUTE: Disable PCI INTx Routing to IOH: 1: When this bit is set, INTx messages received from the PCI Express ports of the IOH are not routed to ESI - they are either converted into MSI via the integrated I/OxAPIC (if the I/OxAPIC mask bit is clear in the appropriate entries) or cause no further action (when mask bit is set). 0: When this bit is clear, INTx messages are routed to ESI, if the corresponding Redirection Table Entry within the IOxAPIC has been masked,

3.8.8.30 SSCTRL[9:0]: Stop and Scream Control Register

This register provides the ability for the Intel® 5400 chipset MCH to filter poisoned TLP in the outbound direction (writes, read completions) and prevent the data from corrupting the end point.

Device: 9 - 1, 0 Function: 0 Offset: 47h			
Bit	**Attr**	**Default**	**Description**
7:2	RV	0h	***Reserved***.
1	RV	0h	***Reserved***.
0	RW	0	SSEN: Stop and Scream Enable 0: Disable Stop and Scream (normal/default operation) 1: Enables the link to perform Stop and Scream functionality . When this bit is set, if the Intel® 5400 chipset MCH encounters a poisoned data traversing the outbound path towards the PCI Express/ESI Port, the IOU cluster stops the data, sets the “link_down condition” internally and places the link in “hot_reset”. Inbound/outbound paths are blocked and all outstanding requests will be master aborted including Memory, I/O and configuration requests.

3.8.8.31 PEXCTRL[9:0]: PCI Express Control Register

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

Device: 9 - 0
Function: 0
Offset: 48h

Bit	Attr	Default	Description
31:28	RW	0h	Reserved
27-22	RV	0	Reserved
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 • IO{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
20:14	RV	0h	*Reserved*
13	RW	0	LNKSTS_IDO: PEXLNKSTS Interrupt Disable Override This bit allow BIOS to override the interrupt control in the PEXLNKCTRL register. 1: Override interrupt control and disable PEXLNKSTS interrupts. 0: Allow PEXLNKSTS interrupts to be controlled by PEXLNKCTRL registers.
12	RW	1	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 the MCH to wait for all coalescable data before sending the transaction as opposed to forwarding as much as possible. However if there are insufficient data credits, MCH will not issue a completion with the maximum payload as indicated by the completion combining limit. 0: Normal operation 1: wait to coalesce data
10	RW	1	**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.
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	RV	0	Reserved
7	RV	1	Reserved

Device: 9 - 0 Function: 0 Offset: 48h			
Bit	**Attr**	**Default**	**Description**
6:3	RWS	0000	Reserved
2	RV	1	Reserved
1	if (port 0} {RO} else {RW} endif	0	DIS_APIC_EOI; Disable APIC EOI The Intel® 5400 chipset MCH 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. In general, EOI should be disabled for active ports that have a non-IO APIC controller attached to them for performance considerations.
0	RW	0	**DISINBPDS: Disable Inband Presence Detect** This register field will be used by the LTSSM to disable the “inband presence detect” mechanism using the State Transactions described in Table 4-4 of the *PCI-Express Base Specification*, Revision 2.0. This field can be used by software for controlling the presence detect of a card using the PEXSLOTSTS.PDS bit. see Section 0: Enable inband presence detect 1: Disable inband presence detect

3.8.8.32 PEXGCTRL - PCI Express Global Control Register

This 32-bit global register in the MCH implements chipset specific operations for generalized control of all PCI Express events and activity such as Power Management. There is only one register for all PCI Express ports that controls related I/O operations.

Device: 0 Function: 0 Offset: D8h			
Bit	**Attr**	**Default**	**Description**
31:8	RV	0h	Reserved.
7:3	RW	1000	**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 ESI port before it back pressures the inbound PCIE ports. 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 is illegal and may cause a system
2	RV	0	Reserved
1	RWS	0	PME_TURN_OFF: Send PME Turn Off Message When set, the MCH will issue a PME Turn Off Message to all enabled PCI Express ports excluding the ESI port.3 The MCH will clear this bit once the Message is sent.
0	RW1C	0	**PME_TO_ACK: Received PME Time Out Acknowledge Message** The MCH 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 ESI2 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 MCH ESI port, it will be Master Aborted.

3.8.8.33 INTXSWZCTRL[9:1]: PCI Express Interrupt Swizzle Control Register

Device: 9-1 Function: 0 Offset: 4Fh			
Bit	**Attr**	**Default**	**Description**
7:2	RO	0h	Reserved
1:0	RWO	00	INTxSWZ: INTx Swizzle The encoding below defines the target Intx type to which the incoming Intx message is mapped to for that port. (4 combinations using the Barber-pole stride mechanism) 00: INTA=>INTA, INTB=>INTB, INTC=>INTC, INTD=>INTD (default 1:1) 01: INTA=>INTB, INTB=>INTC, INTC=>INTD, INTD=>INTA 10: INTA=>INTC, INTB=>INTD, INTC=>INTA, INTD=>INTB 11: INTA=>INTD, INTB=>INTA, INTC=>INTB, INTD=>INTC

This register provides software the ability to swizzle the legacy interrupts (intx) from each port and remap them to a different interrupt type (IntA,B,C,D) for the purposes of interrupt rebalancing to optimize system performance. This swizzling only applies to inbound intx messages that arrive at the various ports (including ESI). The default setting is to have one-to-one map of the same interrupt types i.e (INTA => INTA, etc.). BIOS can program this register during boot time (before enabling interrupts) to swizzle the intx types for the various ports within the combinations described in this register. MCH will use the transformed intx messages from the various ports and track them using the bit vector as a wired-or logic for sending assert/deassert intx messages on the ESI.

3.8.8.34 PEXLWTCTRL: PCI Express Link Width and Training Control Register

This register provides the ability for software to configure the width of PCI Express ports 1-8. MCH 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 3.8.10.5, "DEVPRES: Device Present Control Register" , provides the primary method for BIOS to enforce a specific setting for a given platform.

The following sequence is required by BIOS to determine the number of ports and the maximum port widths for PCI Express ports in IOU0/IOU1:

1. BIOS must first determine the active PCI Express ports by programming the DEVPRES register. Table 3-34, "IOU0 or IOU1 PCI Express Port and Width Selection" shows how the DEVPRES register values determine the port width. The DEVPRES register value takes higher precedence over the PEXLWTCTRL register when a conflict arises. PCI Express port devices that are "not present" will not be shutdown.
2. If ports 1,3,5, and 7 are active, BIOS will need to ensure that the port width register fields are correctly programmed in the PEXLWTCTRL register before initiating link training.
3. BIOS will initiate link training.

Device: 0 Function: 0 Offset: E0h			
Bit	**Attr**	**Default**	**Description**
31:28	RV	0h	Reserved.

Device: 0
Function: 0
Offset: E0h

Bit	Attr	Default	Description
27:26	RW	01	PEXWID7: IOU1 Port 7 Maximum Width This field is used to determine the port width of IOU1 port 7 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 7 is "not present", this field has no effect. Refer to Table 3-34 to determine when HW will use this field to determine the width of port 7. 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	10	PEXWID5: IOU1 Port 5 Maximum Width This field is used to determine the port width of IOU1 port 5 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 5 is "not present", this field has no effect. Refer to Table 3-34 to determine when HW will use this field to determine the width of port 5. 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 10: x16 maximum port width Others: Reserved
23:17	RV	0h	Reserved.
16	RWL	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 5, 6, 7, and 8 if present as determined by Section 3.8.10.5, "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	0h	Reserved.
11:10	RW	01	PEXWID3: IOU0 Port 3 Maximum Width This field is used to determine the port width of IOU0 port 3 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 3 is "not present", this field has no effect. Refer to Table 3-34, "IOU0 or IOU1 PCI Express Port and Width Selection" to determine when HW will use this field to determine the width of port 3. 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	RW	10	PEXWID1: IOU0 Port 1 Maximum Width This field is used to determine the port width of IOU0 port 1 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 1 is "not present", this field has no effect. Refer to Table 3-34, "IOU0 or IOU1 PCI Express Port and Width Selection" to determine when HW will use this field to determine the width of port 1. 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 10: x16 maximum port width Others: Reserved
7:1	RV	0h	Reserved.

Device: 0 Function: 0 Offset: E0h			
Bit	**Attr**	**Default**	**Description**
27:26	RW	01	PEXWID7: IOU1 Port 7 Maximum Width This field is used to determine the port width of IOU1 port 7 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 7 is "not present", this field has no effect. Refer to Table 3-34 to determine when HW will use this field to determine the width of port 7. 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	10	PEXWID5: IOU1 Port 5 Maximum Width This field is used to determine the port width of IOU1 port 5 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 5 is "not present", this field has no effect. Refer to Table 3-34 to determine when HW will use this field to determine the width of port 5. 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 10: x16 maximum port width Others: Reserved
23:17	RV	0h	Reserved.
16	RWL	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 5, 6, 7, and 8 if present as determined by Section 3.8.10.5, "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	0h	Reserved.
11:10	RW	01	PEXWID3: IOU0 Port 3 Maximum Width This field is used to determine the port width of IOU0 port 3 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 3 is "not present", this field has no effect. Refer to Table 3-34, "IOU0 or IOU1 PCI Express Port and Width Selection" to determine when HW will use this field to determine the width of port 3. 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	RW	10	PEXWID1: IOU0 Port 1 Maximum Width This field is used to determine the port width of IOU0 port 1 before link training is initiated when the port is present as determined by the DEVPRES register. If device port 1 is "not present", this field has no effect. Refer to Table 3-34, "IOU0 or IOU1 PCI Express Port and Width Selection" to determine when HW will use this field to determine the width of port 1. 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 10: x16 maximum port width Others: Reserved
7:1	RV	0h	Reserved.

Device: 0 Function: 0 Offset: E0h			
Bit	**Attr**	**Default**	**Description**
0	RWL	0	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, 3, and 4 if present as determined by Section 3.8.10.5, "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.

Table 3-34. IOU0 or IOU1 PCI Express Port and Width Selection

DEVPRES[4:1] (or DEVPRES[8:5])	Port 1 (or port 5)	Port 2 (or port 6)	Port 3 (or port 7)	Port 4 (or port 8)
0000b	Not Present	Not Present	Not Present	Not Present
0001b	x16[1],x8,x4 [2]	Not Present	Not Present	Not Present
1101b	x8,x4 [2]	Not Present	x4	x4
1001b	x8,x4 [2]	Not Present	Not Present	x4
0101b	x8,x4 [2]	Not Present	x8,x4 [3]	Not Present
0111b	x4	x4	x8,x4 [3]	Not Present
0100b	Not Present	Not Present	x8,x4 [3]	Not Present
0110b	Not Present	x4	x8,x4 [3]	Not Present
0010b	Not Present	x4	Not Present	Not Present
1111b	x4	x4	x4	x4
1110b	Not Present	x4	x4	x4
0011b	x4	x4	Not Present	Not Present
1010b	Not Present	x4	Not Present	x4
1011b	x4	x4	Not Present	x4
0010b	Not Present	x4	Not Present	Not Present
1100b	Not Present	Not Present	x4	x4
1000b	Not Present	Not Present	Not Present	x4

Notes:
1. The maximum capable port width can only be X16, x8 or x4 regardless of the PEXLWTCTRL registers values. If the value of the PEXLWTCTRL.PEXWID1 (or PEXLWTCTRL.PEXWID5) indicates a port width larger than allowed in the table, hardware will set the maximum capable port width to maximum width allowed.
2. The PEXLWTCTRL.PEXWID1 field (or PEXLWTCTRL.PEXWID5 field) determines the port's maximum capable width. If the value of the PEXLWTCTRL.PEXWID1 (or PEXLWTCTRL.PEXWID5) indicates a port width larger than allowed in the table, hardware will set the maximum capable port width to maximum width allowed.
3. The PEXLWTCTRL.PEXWID3 field (or PEXLWTCTRL.PEXWID7 field) determines the port's maximum capable width.

3.8.9 PCI Express Power Management Capability Structure

The Intel® 5400 chipset MCH PCI Express port provides basic power management capabilities to handle PM events for compatibility. The PCI Express ports can be placed in a pseudo D3 hot state but it does have real power savings and works as if it were in the D0 mode.

3.8.9.1 PMCAPLST[9:0] - Power Management Capability List Register

The PM Capabilities Register defines the capability ID and next pointer. The following PM registers /capabilities are added for software compliance.

Device: 9 - 0
Function: 0
Offset: 50h

Bit	Attr	Default	Description
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.

3.8.9.2 PMCAP[9:0] - Power Management Capabilities Register

The PM Capabilities Register defines the power management capabilities for the PCI Express and ESI port. The following PM registers /capabilities are added for software compliance.

Device: 9 - 0
Function: 0
Offset: 52h

Bit	Attr	Default	Description
15:11	RO	11001	PMES: PME Support Identifies power states in the MCH 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 MCH)** XX1XXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D2 **(Not supported by MCH)** X1XXXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D3 hot **(Supported by MCH)** 1XXXXb - Assert_PMEGPE/Deassert PMEGPE can be sent from D3 cold **(Not supported by MCH)**
10	RO	0	D2S: D2 Support The MCH does not support power management state D2.
9	RO	0	D1S: D1 Support The MCH does not support power management state D1.
8:6	RO	0h	AUXCUR: AUX Current
5	RO	0	DSI: Device Specific Initialization
4	RV	0	*Reserved.*

Device: 9 - 0 Function: 0 Offset: 52h			
Bit	**Attr**	**Default**	**Description**
3	RO	0	PMECLK: PME Clock This field is hardwired to 0h as it does not apply to PCI Express.
2:0	RO	011	VER: Version This field is set to 3h as version number that indicates this function complies with revision 1.2 of the *PCI Power Management Interface Specification* as required from the*PCI-Express Base Specification*, Revision 2.0.

3.8.9.3 PMCSR[9:0] - Power Management Control and Status Register

This register provides status and control information for PM events in the PCI Express port of the MCH.

Device: 9 - 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® 5400 chipset MCH 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.
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	RW1CS	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® 5400 chipset 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	RWS	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® 5400 chipset MCH since it does not generate PME events internally
7:4	RV	0h	*Reserved.*
3	RWO	1	NSR: No Soft Reset This bit when '1' indicates that a device transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. The Configuration context is preserved. A transition from D3host to D0 by a system or bus segment reset will return the device state to D0 Uninitialized with only PME context preserved.
2	RO	0	Reserved for PCI Express

Device: 9 - 0 Function: 0 Offset: 54h			
Bit	**Attr**	**Default**	**Description**
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 (reserved), then the power state will not change from D0 or D3_hot. A write of 01b or 10b to this field will have no effect. The PCIE/ESI port behavior as if BME=MSE=IOSE=0 when in non-D0 device state. Type0 Cfg, messages, completions, and IOxAPIC memory space (0xFECx_xxxx) are unaffected by device states of the PCIE/ESI port. Inbound CFG are will receive a UR response.

3.8.10 PCI Express Message Signaled 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.

3.8.10.1 MSICAPLST[9:0] - MSI Capability List Register

Device: 9 - 0 Function: 0 Offset: 58h			
Bit	**Attr**	**Default**	**Description**
15:8	RO	6Ch	NXTPTR: Next Ptr This field is set to 6Ch for the next capability list (PCI Express capability structure - PEXCAP) in the chain.
7:0	RO	05h	CAPID: Capability ID Assigned by PCI-SIG for message signaling capability.

3.8.10.2 MSICTRL[9:0] - MSI Message Control Register

Device: 9 - 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).

Device: 9 - 0 Function: 0 Offset: 5Ah			
Bit	**Attr**	**Default**	**Description**
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. 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® 5400 chipset MCH GNB will only use the LSB of the MMEN (as a power of 2) to decode up to 2 messages.
3:1	RWO	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® 5400 chipset MCH is designed to handle MSIs for different events • HP/PM/Link Sts events • RAS Error events
0	RW	0	MSIEN: MSI Enable The software sets this bit to select legacy interrupts or transmit MSI messages. The device is prohibited from using its INTx legacy interrupts. 0: Disables MSI from being generated. 1: Enables the Intel® 5400 chipset MCH to use MSI messages to request context specific service through register bits for events such as PM, RAS.

3.8.10.3 MSIAR[9:0] - 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: 9 - 0 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	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 IA32 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

3.8.10.4 MSIDR[9:0] - MSI Data Register

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

Device: 9 - 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 consistently 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® 5400 chipset MCH to provide context sensitive interrupt information for different events that require attention from the processor. e.g Power Management and RAS error events. Depending on the number of Messages enabled by the processor in Section 3.8.10.2, and Table 3-35 illustrates the breakdown.

Table 3-35. IV Handling and Processing by MCH

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

Notes:

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

3.8.10.5 DEVPRES: Device Present Control Register

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

Writes to the DEVPRES register after it has been locked-down shall not modify the value of any RWL bits. BIOS is expected to lock the DEVPRES register by writing 00h to the MSB before handing off control to the OS.

When a device is marked "not present", all accesses from the processor to that device shall be routed to ESI to be master aborted by the I/O Hub (exactly as if the Device were not implemented in the MCH). However, any registers such as Base Address Registers (BAR's) or Memory Access Enable associated with a device shall not be cleared when a device is marked "not present".

BIOS shall 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 1, 3, 5 or 7 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 "PEXLWTCTRL: PCI Express Link Width and Training Control Register" and Table 3-34 for more details.

For Devices 1-9, 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 18 (IOxAPIC) requires some special exceptions. Accesses to the memory regions defined by MBAR or ABAR are sent to the IOxAPIC independent of the state of bit 18 where its routing is subject to APICDIS bit. SMBus and JTAG accesses to the Configuration Space of Devices marked "not present" shall be allowed by default.

Device: 0
Function: 0
Offset: 68h

Bit	Attr	Default	Description
31	RWO	1	Unlocked Status
30:24	RV	00h	Reserved
23	RV	0	Reserved
22	RWL	1	FB-DIMM 1 Device
21	RO	1	FB-DIMM 0 Device
20:19	RV	00	Reserved
18	RW	1	IOxAPIC Device Note: Replaces IOxAPIC PCI CFG hide bi
17	RO	1	CE Device
16	RO	1	FSB and Addr Decode Device
15	RV	1	Reserved
14:10	RV	0h	Reserved
9	RWL	1	PCIe Port 9 Device
8	RWL	1	PCIe Port 8 Device
7	RWL	1	PCIe Port 7 Device
6	RWL	1	PCIe Port 6 Device
5	RWL	1	PCIe Port 5 Device
4	RWL	1	PCIe Port 4 Device

Device: 0 Function: 0 Offset: 68h			
Bit	**Attr**	**Default**	**Description**
3	RWL	1	PCIe Port 3 Device
2	RWL	1	PCIe Port 2 Device
1	RWL	1	PCIe Port 1 Device
0	RO	1	ESI Port Device

3.8.11 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.

3.8.11.1 PEXCAPLST[9:0]- 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: 9 - 0 Function: 0 Offset: 6Ch			
Bit	**Attr**	**Default**	**Description**
15:8	RO	00h	NXTPTR: Next Ptr This field is set to NULL pointer to terminate the PCI capability list.
7:0	RO	10h	CAPID: Capability ID Provides the PCI Express capability ID assigned by PCI-SIG.

3.8.11.2 PEXCAP[9:0] - PCI Express Capabilities Register

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

Device: 9 - 0 Function: 0 Offset: 6Eh			
Bit	**Attr**	**Default**	**Description**
15	RV	0h	*Reserved.*
14	RO	0	TCSRS: TCS Route Support This bit when set indicates that the PCI Express root port supports routing of Trusted Configuration Requests. Root ports that support routing of trusted Configuration Requests must also set this bit.
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.

Device: 9 - 0 Function: 0 Offset: 6Eh			
Bit	**Attr**	**Default**	**Description**
8	if (port 9-1) {RWO} elsif (port 0) {RO} endif	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 always be 0b since it is required for boot. Rest of the PCI_Express ports which are slotted, BIOS or software can set this field to enable the slots.
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® 5400 chipset MCH.
3:0	RWO	0001	VERS: Capability Version This field identifies the version of the PCI Express capability structure. Set to 0001 by PCI SIG.

3.8.11.3 PEXDEVCAP[9:0] - PCI Express Device Capabilities Register

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

Device: 9 - 0 Function: 0 Offset: 70h			
Bit	**Attr**	**Default**	**Description**
31:28	RV	0h	*Reserved.*
28	RO	0	FLR Capable: This bit indicates that this device is capable of initiating FLRs.
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® 5400 chipset MCH 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® 5400 chipset MCH as it is a Root complex. Hardwired to 00h.
17:16	RV	0h	Reserved
15	RO	1	RBER: Role-Based Error Reporting MCH supports *PCI-Express Base Specification*, Revision 2.0 Error Reporting.
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 Intel® 5400 chipset MCH 1: *Reserved*
13	RO	0	AIPD: Attention Indicator Present This bit when set indicates that an Attention Indicator is implemented. 0: AIPD is disabled in Intel® 5400 chipset MCH 1: *Reserved*

Device: 9 - 0 Function: 0 Offset: 70h			
Bit	**Attr**	**Default**	**Description**
12	RO	0	ABPD: Attention Button Present This bit when set indicates that an Attention Button is implemented. 0: ABPD is disabled in Intel® 5400 chipset MCH 1: *Reserved*
11:9	RO	000	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 For other than endpoints, this field must be set to 000b.
8:6	RO	000	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 For other than endpoints, this field must be set to 000b.
5	RO	0	ETFS: Extended Tag Field Supported This field indicates the maximum supported size of the Tag field. 0: In the Intel® 5400 chipset MCH, 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 Intel® 5400 chipset 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

3.8.11.4 PEXDEVCTRL[9:0] - PCI Express Device Control Register

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

Device: 9 - 0
Function: 0
Offset: 74h

Bit	Attr	Default	Description
15	RV	0h	*Reserved.*
14:12	RO	000	**MRRS: Max_Read_Request_Size** This field sets maximum Read Request size generated by the Intel® 5400 chipset MCH. The PCI Express port must not generate read requests with size exceeding the set value. 000: 128B max read request size 001: 256B max read request size 010: 512B max read request size 011: 1024B max read request size 100: 2048B max read request size 101: 4096B max read request size 110: Reserved 111: Reserved The MCH will not generate read requests larger than 64B in general on the outbound side due to the internal Micro-architecture (CPU initiated, Intel® QuickData Technology device 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 Intel® 5400 chipset MCH never sets or modifies the “No snoop bit” in the received TLP even if ENNOSNP is enabled. For outbound traffic, the Intel® 5400 chipset MCH does not need to snoop.
10	RWS	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 3.8.9.3.
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. Intel® 5400 chipset 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 Intel® 5400 chipset MCH does not use this field (Root complex) and is hardwired to 0.

Device: 9 - 0
Function: 0
Offset: 74h

Bit	Attr	Default	Description
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 Intel® 5400 chipset 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) 010: 512B max payload size (unsupported) 011: 1024B max payload size (unsupported) 100: 2048B max payload size (unsupported) 101: 4096B max payload size (unsupported) others: Reserved Note that MCH supports 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.
4	RO	0	ENRORD: Enable Relaxed Ordering Intel® 5400 chipset 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 3.8.11.12.
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

3.8.11.5 PEXDEVSTS[9:0] - PCI Express Device Status Register

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

Device: 9 - 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.
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

3.8.11.6 PEXLNKCAP[9:0] - PCI Express Link Capabilities Register

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

Device: 9 - 0
Function: 0
Offset: 78h

Bit	Attr	Default	Description
31:24	RO	Dev ? 0 : 0h 1 : 4h 2 : 5h 3 : 6h 4 : 7h 5 : 8h 6 : 9h 7 : Ah 8 : Bh 9 : Ch	**PN: Port Number** This field indicates the PCI Express port number for the link. port 0- device number of 0 (ESI) with link number of 0 port 1- device number of 1 with link number of 4 port 2 - device number of 2 with link number of 5 port 3 - device number of 3 with link number of 6 port 4 - device number of 4 with link number of 7 port 5- device number of 5 with link number of 8 port 6- device number of 6 with link number of 9 port 7- device number of 7 with link number of 10 port 8- device number of 8 with link number of 11 port 9- device number of 9 with link number of 12
23:22	RV	0h	*Reserved.*
21	RO	Dev ? 0 : 0 1	**LBNC: Link Bandwidth Notification Capability** MCH supports Link Bandwidth Changed status and interrupts mechanisms. This capability is required for all Root Port and Switch downstream ports supporting links wider than x1 and/or multiple link speeds. Note: Not supported by Port 0. If speed and width change notification is defeatured, this field will not be updated by hardware to reflect a no notification state
20	RO	1	**DLLAEN: Data Link Layer Active Reporting Capable** 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** MCH is capable of detecting and reporting a surprise link down error condition.
18	RO	0h	**Reserved.**
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 The Intel® 5400 chipset MCH does not support L1 acceptable latency and is set to the maximum value for safety

Device: 9 - 0 Function: 0 Offset: 78h			
Bit	**Attr**	**Default**	**Description**
14:12	RO	100h	**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
11:10	RWO	11	**ASPMCTRL: Active State Power Management (ASPM) Control** This field indicates the level of active state power management supported on the given PCI Express port. 00: Disabled 01: L0s Entry Supported 10: Reserved 11: L0s and L1 Supported
9:4	RO	04h	**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 010000: x16 Others - *Reserved* After link training is initiated, the number of PCI Express ports and their Maximum Link Width is lock by hardware. The value in this field may change before link training.

3:0	RO	If (revision 2.0 {1h}
		elsif (port 0) {1h}
		else {2h}
		endif
			SLS: Supported Link Speed
			This field indicates the maximum Link speed of the given PCI Express port.
			0001: 2.5 Gb/s Link speeds supported
			0010: 5 Gb/s and 2.5Gb/s Link speeds supported
			Others - *Reserved*

Table 3-36. Maximum Link Width Default Value for Different PCI Express Ports

Device/Port	Maximum Link Width	Value
0,2,4,6,8,9	x4	000100
3,7	x8	001000
	x4	000100
1,5	x16	010000
	x8	001000
	x4	000100

Table 3-36 shows various combining options for PCI Express ports. When ports combine, the control registers for the combined port revert to the lower numbered port. Thus when ports 1and 2 are combined, the combined x8 port is accessed through port 1 control registers.

3.8.11.7 PEXLNKCTRL[9:0] - PCI Express Link Control Register

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

Device: 9 - 0
Function: 0
Offset: 7Ch

Bit	Attr	Default	Description
15:12	RV	0h	Reserved.
11	if (port0) {RO} else {RW} endif	0	LABIE: Link Autonomous Bandwidth Interrupt Enable This bit when '1' enables the generation of an interrupt to t indicate that the Link Autonomous Bandwidth Status bit has been set.
10	if (port0) {RO} else {RW} endif	0	LBMIE: Link Bandwidth Management Interrupt Enable This bit when '1' enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been set.
9	RW	0	HAWD: Hardware Autonomous Width Disable This bit when '1' disables the hardware for changing the link width for reasons other than attempting to correct unreliable link operation by reducing link width
8	RO	0	ENCKPRM: Enable Clock Power Management This bit when '1' indicates that the component tolerates the removal of any reference clock(s) via the "clock request" CLKREQ# mechanism when the link is in L1 or L2/L3 Ready link state.
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	RW	0	**RLNK: Retrain Link** This bit, when set, initiates link retraining in the given PCI Express port. It consistently 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 Software should wait a minimum of 2ms to make sure the link has entered the electrical idle state before clearing this bit.
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 Intel® 5400 chipset MCH supports only 64B read completion boundary and is hardwired to 0.

Device: 9 - 0 Function: 0 Offset: 7Ch			
Bit	**Attr**	**Default**	**Description**
2	RV	0	*Reserved.*
1:0	RW	00	ASTPMCTRL: 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: L1 Entry Supported 11: L0s and L1 Supported

3.8.11.8 PEXLNKSTS[9:0] - 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: 9 - 0 Function: 0 Offset: 7Eh			
Bit	**Attr**	**Default**	**Description**
15	RWC	0	LABS: Link Autonomous Bandwidth Status This bit is set by hardware to 1b to indicate that hardware has autonomously changed link speed or width, without the port transitioning through DL_Down Status, for reasons other than to attempt to correct unreliable link operation. This bit must be set when the upstream component receives eight consecutive TS1 or TS2 ordered sets with the Autonomous Change bit set.
14	RWC	0	LBMS: Link Bandwidth Management Status This bit is set by hardware to 1b to indicate that either of the following has occurred without the port transitioning through DL_Down status: 1. A link retraining initiated by a write of 1b to the Retrain Link bit has completed 2. Hardware has autonomously changed link speed or width to attempt to correct unreliable link operation, either through an LTSSM timeout or a higher level process
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 Intel® 5400 chipset MCH initializes this bit to '1' because the expected state of the platform is to have one clock source shared between the Intel® 5400 chipset 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.

Device: 9 - 0 Function: 0 Offset: 7Eh			
Bit	**Attr**	**Default**	**Description**
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 3.8.11.7, "PEXLNKCTRL[9:0] - PCI Express Link Control Register" 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	**Undefined by *PCI-Express Base Specification***, Revision 2.0 This field was formerly the Training Error status bit in PCIe v1.0a.
9:4	RO	000100	**NLNKWD: Negotiated Link Width**[1] This field indicates the negotiated width of the given PCI Express link after training is completed. Only x1, x2, x4, x8, and x16 link width negotiations are possible in the MCH. Refer to Table 3-37, "Negotiated Link Width For Different PCI Express Ports After Training" for the port and link width assignment after training is completed.
3:0	RO	1h	CLNKSPD: Current Link Speed This field indicates the negotiated Link speed of the given PCI Express Link: 0001 - 2.5 Gb/s PCI Express Link 0010 - 5 Gb/s PCI Express Link Others - *Reserved*

Notes:

1. 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 3-37. Negotiated Link Width For Different PCI Express Ports After Training

Device/Port	Negotiated Link Width	Value
1,2,3,4,5,6,7,8,9	x1	000001
1 - 9	x2	000010
1 - 9	x4	000100[1]
1,3,5,7	x8	001000[1]
1,5	x16	010000[1]

Notes:

1. Ports that do not train report 000000.

3.8.11.9 PEXSLOTCAP[9:0] - PCI Express Slot Capabilities Register

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

Device: 9 - 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 Intel® 5400 chipset MCH.
18	RO	0h	CCNC: Command Complete Not Capable MCH is capable of command complete interrupt.
17	RWO	0h	EMIP: Electromechanical Interlock Present This bit when set is indicates that an Electromechanical Interlock is implemented on the chassis for this slot and that lock is controlled by bit 11 in Slot Control Register. BIOS Note: This capability is not set if the Electromechanical Interlock control is connected to main slot power control.
16:15	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	RV	0h	Reserved
5	RV	0h	*Reserved*
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 BIOS programs this field with a 1 for CEM/SIOM FFs and a 0 for Express cable.
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 BIOS programs this field with a 1 for CEM/SIOM FFs.
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 BIOS programs this field with a 0 for SIOM/EXpress cable and with either 0 or 1 for CEM depending on system design.

Device: 9 - 0 Function: 0 Offset: 80h			
Bit	**Attr**	**Default**	**Description**
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 BIOS programs this with a 1 for CEM/SIOM FFs and a 0 for Express cable.
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 BIOS programs this with a 1 for CEM/SIOM FFs.

3.8.11.10 PEXSLOTCTRL[9:0] - PCI Express Slot Control Register

The Slot Control register identifies the PCI Express specific slot control specific parameters for operations such as Power Management. Software issues a command to a 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: 9 - 0 Function: 0 Offset: 84h			
Bit	**Attr**	**Default**	**Description**
15:13	RV	0h	*Reserved.*
12	RW	0	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	RW	0	EMIC: Electromechanical Interlock Control When software writes either a 1 to this bit, MCH pulses the EMIL pin per PCI-Express Server/Workstation Module Electromechanical Spec Rev 0.5a. Write of 0 has no effect. This bit always returns a 0 when read. If electromechanical lock is not implemented, then either a write of 1 or 0 to this register has no effect.
10	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

Device: 9 - 0
Function: 0
Offset: 84h

Bit	Attr	Default	Description
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 register is written, the event is signaled via the virtual pins of MCH over a dedicated SMBus port. MCH does not generated the Power_Indicator_On/Off/Blink messages on PCI Express when this field is written to by software.
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 register is written, the event is signaled via the virtual pins of the IOH over a dedicated SMBus port. IOH does not generated the Attention_Indicator_On/Off/Blink messages on PCI Express when this field is written to by software.
5:3	RV	0h	Reserved
2	RW	0h	**MRLINTEN: MRL Sensor Changed Enable** This bit enables the generation of wake messages via a MRL Sensor changed event. 0: Disables generation of wake messages when an MRL Sensor changed event happens. 1: Enables generation of wake messages when an MRL Sensor changed event happens.
1	RW	0h	**PWRINTEN:** Power Fault Detected Enable This bit enables the generation of wake messages via a power fault event. 0: Disables generation of wake messages when a power fault event happens. 1: Enables generation of wake messages when a power fault event happens.
0	RW	0h	**ATNINTEN: Attention Button Pressed Enable** This bit enables the generation of wake messages via an attention button pressed event. 0: Disables generation of wake messages when the attention button is pressed. 1: Enables generation of wake messages when the attention button is pressed.

3.8.11.11 PEXSLOTSTS[9:0] - PCI Express Slot Status Register

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

Device: 9 - 0
Function: 0
Offset: 86h

Bit	Attr	Default	Description
15:9	RV	0h	*Reserved.*
8	RWC	0	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.
7	RO	0	EMLS: Electromechanical Latch Status A read to this register returns the current state of the Electromechanical Interlock (the EMILS pin) which has the defined encodings as: 0: Electromechanical Interlock Disengaged 1: Electromechanical Interlock Engaged
6	RO	1h	**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
5	RO	0h	**MRLSS:** MRL Sensor State This bit reports the status of an MRL sensor if it is implemented. 0: MRL Closed 1: MRL Open
4	RV	0h	Reserved
3	RWC	0h	**PRSINT: Presence Detect Changed** This bit is set by the Intel® 5400 chipset MCH when a Presence Detect Changed event is detected. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding to this bit inactive if the FF/system does not support out-of-band presence detect.
2	RWC	0h	**MRLSC:** MRL Sensor Changed This bit is set by the Intel® 5400 chipset MCH when an MRL Sensor Changed event is detected. It is subsequently cleared by software after the field has been read and processed. On-board logic per slot must set the VPP signal corresponding to this bit inactive if the FF/system does not support out-of-band presence detect.
1	RWC	0h	**PWRINT:** Power Fault Detected This bit is set by the Intel® 5400 chipset 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. 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.
0	RWC	0h	**ABP:** Attention Button **Pressed** This bit is set by the Intel® 5400 chipset 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 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 3.8.11.10, "PEXSLOTCTRL[9:0] - PCI Express Slot Control Register" and HPINTEN are enabled. Software writes to clear these bits and 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.

Whereas 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.

3.8.11.12 PEXRTCTRL[9:0] - PCI Express Root Control Register

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

Device: 9 - 0
Function: 0
Offset: 88h

Bit	Attr	Default	Description
15:5	RV	0h	*Reserved.*
4	RO	0h	CRSSVE: CRS software visibility enable This bit, when set, enables the Root Port to return Configuration Retry Status (CRS) Completion status to software. 1: Enable software to received a CRS status. This allows software to make the decision to re-issue the configuration request or move on and re-issue the request at a later time. 0: Disable software from receiving a CRS response. MCH will wait re-issue the configuration request until it receives a response other than CRS. MCH does not support software visible configuration retry status.
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 3.8.11.14, 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.

Device: 9 - 0 Function: 0 Offset: 88h			
Bit	**Attr**	**Default**	**Description**
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.

3.8.11.13 PEXRTCAP[9:0]: PCI Express Root Capabilities Register

The PCI Express Root Capability register specifies features that are supported in the root complex port.

Device: 9-0 Function: 0 Offset: 8Ah			
Bit	**Attr**	**Default**	**Description**
15:1	RV	0h	*Reserved.*
0	RO	0h	**CRSSV: CRS Software Visibility** This bit, when set, indicates that the Root Port is capable of returning Configuration Retry Status (CRS) on completions to software. MCH does not supports this capability.

3.8.11.14 PEXRTSTS[9:0] - PCI Express Root Status Register

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

Device: 9 - 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 Intel® 5400 chipset 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**[1] 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:

1. 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[2:7] 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 3.8.10.2. Refer to the PM interrupt flow in Power Management Chapter.

3.8.11.15 PEXDEVCAP2[9:0]: PCI Express Device Capabilities 2 Register

This register is an extension of the PCI Express Device Capabilities register identifies device specific information for the port.

Device: 9-0
Function: 0
Offset: 90h

Bit	Attr	Default	Description
31:5	RV	0h	*Reserved.*
4	RO	1	CTDS: Completion Timeout Disable Support A value of 1b indicates support for the completion Timeout Disable Mechanism. Support of completion timeout disable is optional for Root Ports. MCH supports completions timeout disable.
3:0	RO	0111	**CTRS: Completion Timeout Range Supported** This field indicates device support for the optional Completion Timeout programmability mechanism. This mechanism allows system software to modify the Completion Timeout value. This field is applicable only to Root Ports, Endpoints that issue requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of request issues on PCI Express. For all other devices this field is reserved and maybe be hardwired to 0000b. Four time values ranges are defined: Range A: 50 µs to 10 ms Range B: 10 ms to 250 ms Range C: 250 ms to 4 s Range D: 4 s to 64 s Bits ares set according to table below to show timeout value ranges supported. 0000b: Completions Timeout programming not supported -- values is fixed by implementation in the range 50 µs to 50 ms. 0001b: Range A 0010b: Range B 0011b: Range A & B 0110b: Range B & C 0111b: Range A, B, & C 1111b: Range A, B, C & D All other values are reserved.

3.8.11.16 PEXDEVCTRL2[9:0]: PCI Express Device Control 2 Register

This register is an extension of the PCI Express Device Control register that controls PCI Express specific capabilities parameters associated with this port.

Device: 9-0
Function: 0
Offset: 94h

Bit	Attr	Default	Description
15:5	RV	0h	*Reserved.*
4	RW	0	CTPE: Completion timeout Disable 1: disable the completions timeout mechanism for all NP tx. 0: completion timeout is enabled for all NP tx
3:0	RW	0000	**CTV: Completion Timeout Value** In devices that support completion timeout programmability, this field allows system software to modify the completion timeout range. The following encodings and corresponding timeout ranges are defined: 0000b: 50 µs to 50 ms 0001b: 50 µs to 100 µs 0010b: 1 ms to 10 ms 0101b: 16 ms to 55 ms 0110b: 65 ms to 210 ms 1001b: 260 ms to 900 ms 1010b: 1 s to 3.5s Note: It is highly recommended that the completion timeout value not be less then 10 ms or greater. A small completion timeout value may result in premature completion timeout for slower responding devices.

3.8.11.17 PEXDEVSTS2[9:0]: PCI Express Device Status 2 Register

This register is an extension of the PCI Express Device Status Register.

Device: 9-0
Function: 0
Offset: 96h

Bit	Attr	Default	Description
15:0	RV	0h	*Reserved.*

3.8.11.18 PEXLNKCAP2[9:0]: PCI Express Link Capabilities 2 Register

This register is an extension of the PCI Express Link Capability register.

Device: 9-0
Function: 0
Offset: 98h

Bit	Attr	Default	Description
31:0	RV	0h	*Reserved.*

3.8.11.19 PEXLNKCTRL2[9:0]: PCI Express Link Control 2 Register

This register is an extension of the PCI Express Link Control register that controls the PCI Express Link specific parameters.

Device: 9-0
Function: 0
Offset: 9Ch

Bit	Attr	Default	Description
15:7	RV	0h	*Reserved.*
6	RWO	0h	SEL_DE: Selectable De-emphasis When the link is operating at 5 Gb/s speed, this bit selects the level of de-emphasis level of the upstream component. 1: -3.5dB 0: -6 dB When the Link is operating at 2.5 Gb/s speed, the setting of this bit has no effect.
5	RW	0	LBCIEN: Hardware Autonomous Speed Disable This bit disables the component to autonomously direct changes in link speed for reasons other than attempting to correct unreliable link operation. ***Note:*** Initial transition to the highest support common link speed is not blocked by this control bit.
4	RWS	0	FEC: Enter Compliance This bit allows software to force a port's transmitter to enter Compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1 and then initiating a hot reset on the link.
3:0	if (port 0) {RO} else {RW} endif	If (revision 2.0) {1h} elsif (port 0) {1h} else {2h} endif	TLS: Target Link Speed For Downstream ports, this field sets the upper limit on link operational speed by restricting advertised by the upstream component in its training sequence. Defined encodings are: 0001b: 2.5Gb/s Target Link Speed 0010b: 5Gb/s Target Link Speed All other encodings are reserved. If the value is written to this field that does not correspond to a speed included in the Supported link speed field, the behavior is undefined. The default value of this field is the highest link speed supported by the component (as reported in the Supported Link Speeds field of the Link Capabilities Register) unless the corresponding platform / form factor requires a different default value. For both Upstream and Downstream ports, this field is used to set the target compliance mode speed when software is using the Enter Compliance bit to force a link into compliance mode.

3.8.11.20 PEXLNKSTS2[9:0]: PCI Express Link Status 2 Register

This register is reserved for future use.

Device: 9-0 Function: 0 Offset: 9Eh			
Bit	**Attr**	**Default**	**Description**
15:1	RV	0h	*Reserved.*
0	RO	0h	CDL: Current De-emphasis Level When the Link is operating at 5GT/s speed, this reflects the level of de-emphasis. Encodings: 1: -3.5 dB 0: -6 dB When the Link is operating at 2.5GT/s speed, this bit is 0b

3.8.11.21 SCAPLST[9:1]: Subsystem ID Capability List Register

Device: 9-1 Function: 0 Offset: B0h			
Bit	**Attr**	**Default**	**Description**
15:8	RO	00h	NXTPTR: Next Ptr This field is set to NULL pointer to terminate the PCI capability list.
7:0	RO	0Dh	CAPID: Capability ID Assigned by PCI-SIG for Subsystem ID capability.

3.8.11.22 SSVID[9:1]: Subsystem Vendor Identification Register

Device: 9-1 Function: 0 Offset: B4h			
Bit	**Attr**	**Default**	**Description**
15:0	RWO	8086h	Assigned by PCI-SIG for vendor ID.

3.8.11.23 SSID[9:1]: Subsystem Identification Register

Device: 9-1 Function: 0 Offset: B6h			
Bit	**Attr**	**Default**	**Description**
15:0	RWO	8086h	Assigned by subsystem vendor to uniquely identify the subsystem.

3.8.11.24 VT_BAR[0]: VT-d Base Address Register

This is the base address register for the virtualization technology cluster's memory mapped register set. The size of this space is 64 K region and it must reside below 4 GB. The register set for each Intel® QuickData Technology Device-remapping hardware unit in MCH is placed at a size aligned to 4 KB memory-mapped region within this space.

Device: 0
Function: 0
Offset: B0h

Bit	Attr	Default	Description
31:16	RW	FE71h	VTBAR: VT-d Chipset Base Address Register Provides a aligned 64 K address space for MCH registers relating to the Intel VT-d.
15:1	RV	0	Reserved.
0	RW-LO	0	ENBVTBAR: Enable VT-d chipset Base Address Register Enables VT_BAR register.

3.8.11.25 VTCTRL: Virtualization Control Register

This register allows BIOS to program the Intel VT-d host physical address width supported by the CPU and guest physical address width supported by the Intel VT-d address translation engine.

Device: 0
Function: 0
Offset: B4h

Bit	Attr	Default	Description
15:8	RV	0h	Reserved
7:4	RW	4h	VTHPAL: VTd HPA Limit Represents the host processor addressing limit 0000: 2^36 (i.e. bits 35:0) 0001: 2^37 (i.e. bits 36:0) 0010: 2^38 (i.e. bits 37:0) 0011: 2^39 (i.e. bits 38:0) 0100: 2^40 (i.e. bits 39:0) Others: Reserved When Intel VT-d translation is enabled, all host addresses during page walks that go beyond the limit specified in this register will be aborted by MCH

Device: 0 Function: 0 Offset: B4h			
Bit	**Attr**	**Default**	**Description**
3:0	RW	Ch	VTGPAL: VTd GPA Limit Represents the guest addressing limit 0000: 2^36 (i.e. bits 35:0) 0001: 2^37 (i.e. bits 36:0) 0010: 2^38 (i.e. bits 37:0) 0011: 2^39 (i.e. bits 38:0) 0100: 2^40 (i.e. bits 39:0) 0101: 2^41 (i.e. bits 40:0) 0110: 2^42 (i.e. bits 41:0) 0111: 2^43 (i.e. bits 42:0) 1000: 2^44 (i.e. bits 43:0) 1001: 2^45 (i.e. bits 44:0) 1010: 2^46 (i.e. bits 45:0) 1011: 2^47 (i.e. bits 46:0) 1100: 2^48 (i.e. bits 47:0) Others: Reserved When Intel VT-d translation is enabled, all incoming guest addresses from PCI Express that go beyond the limit specified in this register will be aborted by MCH and a UR response returned (with the associated AER logging/reporting). This register is not used when translation is not enabled.

3.8.11.26 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**
31:15	RV	0h	*Reserved.*
14	RW	1	DL23R: Override L23 Ready - Recommend setting this bit to 1. 0: Wait for PME_Enter_L23 on all PCIe ports 1: Do not wait for PME_Enter_L23 on all PCIe ports
13:12	RV	0	Reserved
11	RWC	0	**PTE: PME_TO_Ack Time Expired** 0: Default mode where the Intel® 5400 chipset MCH hardware broadcasts PME_turn_off message through Intel 631xESB/632xESB I/O Controller Hub 1: Signal that time expiration has occurred when the PTOV field described below crosses the threshold in the Intel® 5400 chipset MCH.
10:9	RW	0h	**PTOV: PME_TO_Ack Time Out Value** 00: 1 ms (default) 01: 10 ms 10: 50 ms 11: *Reserved* This register field provides the timer limit for the Intel® 5400 chipset MCH to keep track of the elapsed time from sending "PME_Turn_off" to receiving a "PME_TO_Ack". The test mode assumes a zero delay for the round trip.
8:4	RV	0h	*Reserved.*

Device: 0 Function: 0 Offset: D4h			
Bit	**Attr**	**Default**	**Description**
3:0	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 16). 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

3.8.12 PCI Express Advanced Error Reporting Capability

3.8.12.1 PEXENHCAP[9: 0] - PCI Express Enhanced Capability Header

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

Device: 9 - 0 Function: 0 Offset: 100h			
Bit	**Attr**	**Default**	**Description**
31:20	RO	000h	**NCAPOFF: Next Capability Offset** This field points to the next Capability in extended configuration space.
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

3.8.12.2 UNCERRSTS[9:1] - Uncorrectable Error Status

This register identifies uncorrectable errors detected for the PCI Express Port. If an error occurs and is unmasked in the detect register (EMSAK_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 register gets recorded.These error log registers are described starting from Section 3.8.12.24.

Device: 9 - 1 Function: 0 Offset: 104h			
Bit	**Attr**	**Default**	**Description**
31:22	RV	0h	Reserved
21	RV	0	**Reserved. ACS Violation Status (Unsupported Feature)**
20	RW1CS	0	IO2Err: Received an Unsupported Request

Device: 9 - 1 Function: 0 Offset: 104h			
Bit	**Attr**	**Default**	**Description**
19	RV	0	Reserved
18	RW1CS	0	IO9Err: Malformed TLP Status
17	RW1CS	0	IO10Err: Receiver Buffer Overflow Status
16	RW1CS	0	IO8Err: Unexpected Completion Status
15	RW1CS	0	IO7Err: Completer Abort Status
14	RW1CS	0	IO6Err: Completion Time-out Status
13	RW1CS	0	IO5Err: Flow Control Protocol Error Status
12	RW1CS	0	IO4Err: Poisoned TLP Status
11:6	RV	0h	Reserved
5	RWS	0	IO19Err: Surprise Link Down Error Status
4	RW1CS	0	IO0Err: Data Link Protocol Error Status
3:0	RV	0h	Reserved

3.8.12.3 UNCERRSTS[0] - Uncorrectable Error Status For ESI Port

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 get recorded. These error log registers are described starting from Section 3.8.12.24.

Device: 0 Function: 0 Offset: 104h			
Bit	**Attr**	**Default**	**Description**
31:22	RV	0h	Reserved
21	RV	0	**Reserved. ACS Violation Status (Unsupported Feature)**
20	RW1CS	0	IO2Err: Received an Unsupported Request
19	RV	0	Reserved
18	RW1CS	0	IO9Err: Malformed TLP Status
17	RW1CS	0	IO10Err: Receiver Buffer Overflow Status
16	RW1CS	0	IO8Err: Unexpected Completion Status
15	RW1CS	0	IO7Err: Completer Abort Status
14	RW1CS	0	IO6Err: Completion Time-out Status
13	RW1CS	0	IO5Err: Flow Control Protocol Error Status
12	RW1CS	0	IO4Err: Poisoned TLP Status
11:6	RV	0h	Reserved
5	RWS	0	IOErr: Surprise Link Down Error Status
4	RW1CS	0	IO0Err: Data Link Protocol Error Status
3:0	RV	0h	Reserved

3.8.12.4 UNCERRMSK[9:1] - Uncorrectable Error Mask

This register masks uncorrectable errors from the UNCERRSTS[9:1] register from being signaled.

Device: 9 - 1
Function: 0
Offset: 108h

Bit	Attr	Default	Description
31:22	RV	0h	Reserved
21	RV	0	**Reserved. ACS Violation Status (Unsupported Feature)**
20	RWS	0	IO2Msk: Received an Unsupported Request
19	RV	0	Reserved
18	RWS	0	IO9Msk: Malformed TLP Status
17	RWS	0	IO10Msk: Receiver Buffer Overflow Mask
16	RWS	0	IO8Msk: Unexpected Completion Mask
15	RWS	0	IO7Msk: Completer Abort Status
14	RWS	0	IO6Msk: Completion Time-out Mask
13	RWS	0	IO5Msk: Flow Control Protocol Error Mask
12	RWS	0	IO4Msk: Poisoned TLP Mask
11:6	RV	0h	Reserved
5	RWS	0	IO19Msk: Surprise Link Down Error Mask
4	RWS	0	IO0Msk: Data Link Layer Protocol Error Mask
3:0	RV	000	Reserved

3.8.12.5 UNCERRMSK[0] - Uncorrectable Error Mask For ESI Port

This register masks uncorrectable errors from the UNCERRSTS[0] register (ESI port) from being signaled.

Device: 0
Function: 0
Offset: 108h

Bit	Attr	Default	Description
31:22	RV	0h	Reserved
21	RV	0	**Reserved. ACS Violation Status (Unsupported Feature)**
20	RWS	0	IO2Msk: Received an Unsupported Request
19	RV	0	Reserved
18	RWS	0	IO9Msk: Malformed TLP Status
17	RWS	0	IO10Msk: Receiver Buffer Overflow Mask
16	RWS	0	IO8Msk: Unexpected Completion Mask
15	RWS	0	IO7Msk: Completer Abort Status
14	RWS	0	IO6Msk: Completion Time-out Mask
13	RWS	0	IO5Msk: Flow Control Protocol Error Mask
12	RWS	0	IO4Msk: Poisoned TLP Mask
11:5	RV	0h	Reserved
5	RWS	0	IO19Msk: Surprise Link Down Error Mask

Device: 0 Function: 0 Offset: 108h			
Bit	**Attr**	**Default**	**Description**
4	RWS	0	IO0Msk: Data Link Layer Protocol Error Mask
3:0	RV	000	Reserved

3.8.12.6 UNCERRSEV[9:1] - 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.

Device: 9 - 1 Function: 0 Offset: 10Ch			
Bit	**Attr**	**Default**	**Description**
31:22	RV	0h	Reserved
21	RV	0	**Reserved. ACS Violation Status (Unsupported Feature)**
20	RWS	0	IO2Severity: Received an Unsupported Request
19	RV	0	Reserved
18	RWS	1	IO9Severity: Malformed TLP Severity
17	RWS	1	IO10Severity: Receiver Buffer Overflow Severity
16	RWS	0	IO8Severity: Unexpected Completion Severity
15	RWS	0	IO7Severity: Completer Abort Status
14	RWS	0	IO6Severity: Completion Time-out Severity
13	RWS	1	IO5Severity: Flow Control Protocol Error Severity
12	RWS	0	IO4Severity: Poisoned TLP Severity
11:6	RV	0h	Reserved
5	RWS	0	IO19Severity: Surprise Link Down Severity
4	RWS	1	IO0Severity: Data Link Protocol Error Severity (See Figure 3-17 in *PCI-Express Base Specification*, Revision 2.0)
3:0	RV	000	Reserved

3.8.12.7 UNCERRSEV[0] - Uncorrectable Error Severity For ESI Port

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 corresponding 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.

Device: 0
Function: 0
Offset: 10Ch

Bit	Attr	Default	Description
31:22	RV	0h	Reserved
21	RV	0	**Reserved. ACS Violation Status (Unsupported Feature)**
20	RWS	0	IO2Severity: Received an Unsupported Request
19	RV	0	Reserved
18	RWS	1	IO9Severity: Malformed TLP Severity
17	RWS	1	IO10Severity: Receiver Buffer Overflow Severity
16	RWS	0	IO8Severity: Unexpected Completion Severity
15	RWS	0	IO7Severity: Completer Abort Status
14	RWS	0	IO6Severity: Completion Time-out Severity
13	RWS	1	IO5Severity: Flow Control Protocol Error Severity
12	RWS	0	IO4Severity: Poisoned TLP Severity
11:6	RV	0h	Reserved
5	RWS	0	IO19 Severity: Surprise Link Down Severity
4	RWS	1	IO0Severity: Data Link Protocol Error Severity (See Figure 3-17 in *PCI-Express Base Specification*, Revision 2.0)
3:0	RV	000	Reserved

3.8.12.8 CORERRSTS[9:0] - Correctable Error Status

This register identifies which unmasked correctable error has been detected. The error is directed to the respective device correctable error bit in the PEX_NF_COR_FERR, PEX_NF_COR_NERR registers (If the error is unmasked in the CORERRMSK register defined in Section 3.8.12.9). These registers are discussed starting from Section 3.8.12.25.

Device: 9 - 0
Function: 0
Offset: 110h

Bit	Attr	Default	Description
31:14	RV	0h	Reserved
13	RW1CS	0	**IO20Err: Advisory Non-Fatal Error Status**
12	RW1CS	0	IO16Err: Replay Timer Time-out Status
11:9	RV	0h	Reserved
8	RW1CS	0	IO15Err: Replay_Num Rollover Status
7	RW1CS	0	IO14Err: Bad DLLP Status
6	RW1CS	0	IO13Err: Bad TLP Status

Device: 9 - 0 Function: 0 Offset: 110h			
Bit	**Attr**	**Default**	**Description**
5:1	RV	0h	Reserved
0	RW1CS	0	IO12Err: Receiver Error Status

3.8.12.9 CORERRMSK[9:0] - Correctable Error Mask

This register masks correctable errors from being signalled. They are still logged in the CORERRSTS register.

Device: 9 - 0 Function: 0 Offset: 114h			
Bit	**Attr**	**Default**	**Description**
31:14	RV	0h	Reserved
13	RWS	0	**IO20Msk: Advisor Non-Fatal Error Mask**
12	RWS	0	IO16Msk: Replay Timer Time-out Mask
11:9	RV	0h	Reserved
8	RWS	0	IO15Msk: Replay_Num Rollover Mask
7	RWS	0	IO14Msk: Bad DLLP Mask
6	RWS	0	IO13Msk: Bad TLP Mask
5:1	RV	0h	Reserved
0	RWS	0	IO12Msk: Receiver Error Mask

3.8.12.10 AERRCAPCTRL[9:0] - Advanced Error Capabilities and Control Register

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

Device: 9 - 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 Intel® 5400 chipset MCH does not support ECRC.
6	RO	0	ECRCGENEN: ECRC Generation Enable Intel® 5400 chipset MCH does not generate ECRC.
5	RO	0	ECRCGENCAP: ECRC Generation Capable Intel® 5400 chipset MCH does not generate ECRC.
4:0	ROS	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.

3.8.12.11 HDRLOG0[9:0] - 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: 9 - 0
Function: 0
Offset: 11Ch

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

3.8.12.12 HDRLOG1[9:0] - Header Log 1

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

Device: 9 - 0
Function: 0
Offset: 120h

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

3.8.12.13 HDRLOG2[9:0] - Header Log 2

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

Device: 9 - 0
Function: 0
Offset: 124h

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

3.8.12.14 HDRLOG3[9:0] - Header Log 3

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

Device: 9 - 0
Function: 0
Offset: 128h

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

3.8.12.15 RPERRCMD[9:0] - Root Port Error Command

This register controls behavior upon detection of errors.

Device: 9 - 0
Function: 0
Offset: 12Ch

Bit	Attr	Default	Description
31:8	RV	0h	Reserved

Device: 9 - 0 Function: 0 Offset: 12Ch			
Bit	**Attr**	**Default**	**Description**
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

3.8.12.16 RPERRSTS[9:0] - 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 the MCH, 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 bit 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 3.8.11.12 and the Root Error Command register defined in Section 3.8.12.15.

Device: 9 - 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 an the MSI message if assigned more than one message number to be used of any status in this capability. When there are more than one MSI/MSI-X interrupt vector, 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 this register is set and interrupts are enabled through the "RPERRCMD[9:0]: Root Port Error Command" register. The chipset is required to update the field if the number of MSI messages changes.
26:7	RV	0h	Reserved
6	RW1CS	0	**FAT_ERR_Rcvd:** Fatal Error Messages Received Set when one or more Fatal Uncorrectable error Messages[1] have been received.
5	RW1CS	0	**NFAT_ERR_Rcvd**: Non-Fatal Error Messages Received Set when one or more Non-Fatal Uncorrectable error Messages have been received.
4	RW1CS	0	**FRST_UNCOR_FATAL:** First Uncorrectable Fatal Set when the first Uncorrectable error message (which is FATAL) is received.

Device: 9 - 0 Function: 0 Offset: 130h			
Bit	**Attr**	**Default**	**Description**
3	RW1CS	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	RW1CS	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	RW1CS	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	RW1CS	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:

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

3.8.12.17 RPERRSID[9:0] - 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 3.8.12.16. This register is updated regardless of the settings of the Root Control register defined in Section 3.8.11.12 and the Root Error Command register defined in.Section 3.8.12.15.

Device: 9 - 0 Function: 0 Offset: 134h			
Bit	**Attr**	**Default**	**Description**
31:16	ROS	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	ROS	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.

3.8.12.18 PEX_ERR_DOCMD[9:0] - PCI Express Error Do Command Register

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

Device: 9 - 0 Function: 0 Offset: 144h			
Bit	**Attr**	**Default**	**Description**
31:12	RV	0h	*Reserved*.

Device: 9 - 0 Function: 0 Offset: 144h			
Bit	**Attr**	**Default**	**Description**
11:10	RW	00	**PEX_UNIT_RP_FAT_MAP: Root Port steering for unit fatal errors** 00: ERR[0] 01: ERR[1] 10: ERR[2] 11: MCERR The Unit Root Port Fatal errors are routed to one of the ERR[2:0] pins or MCERR.
9:8	RW	00	**PEX_UNIT_RP_NF_MAP: Root Port steering for Unit non-fatal errors** 00: ERR[0] 01: ERR[1] 10: ERR[2] 11: MCERR The Unit Root Port Fatal errors are routed to one of the ERR[2:0] pins or MCERR.
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.

3.8.12.19 PEX_ERR_PIN_MASK[9:0]: PCI Express Error Pin Mask Register

This register is used to mask PCI Express defined errors from escalating to ERR[2:0] and MCERR.

Device: 9-0
Function: 0
Offset: 146h

Bit	Attr	Default	Description
15:5	RV	0h	***Reserved.***
4	RW	0	DIS_ERR_PINS_OVERRIDE: Disable Error Pins Mask Override This bit disables the Error Pin Mask functionality controlled by MASK_ERR_PINS and FAT/NFAT/COR_ERR_PIN_EN fields of this register.
3	RW	0	MASK_ERR_PINS: Mask Error Pins This bit when set to '1' is used to mask Error Pin assertion for PCI Express defined errors for all classes (Correctable, Non-Fatal, and Fatal) regardless of the value in the FAT/NFAT/COR_ERR_PIN_EN fields of this register.
2	RW1C	0	FAT_ERR_PIN_EN: Fatal Error Messages Received This bit is set by hardware whenever a Fatal Uncorrectable Error Messages[1] have been received for PCI Express defined fatal errors. BIOS will clear this bit to prevent PCI Express defined Correctable errors from asserting the Error Pins.
1	RW1C	0	NFAT_ERR_PIN_EN: Non-Fatal Error Messages Received This bit is set by hardware whenever a Non-Fatal Uncorrectable Error Messages[1] have been received for PCI Express defined non-fatal errors. BIOS will clear this bit to prevent PCI Express defined Correctable errors from asserting the Error Pins.
0	RW1C	0	COR_ERR_PIN_EN: Correctable Error Messages Received This bit is set by hardware whenever a Correctable Error Messages[1] have been received for PCI Express defined correctable errors. BIOS will clear this bit to prevent PCI Express defined Correctable errors from asserting the Error Pins.

Notes:

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

3.8.12.20 EMASK_UNCOR_PEX[0] - Uncorrectable Error Detect Mask For ESI

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.

Device: 0
Function: 0
Offset: 148h

Bit	Attr	Default	Description
31:22	RV	0h	Reserved
21	RV	0	**Reserved. ACS Violation Status (Unsupported Feature)**
20	RWS	0	**IO2DetMsk:** Received an Unsupported Request
19	RV	0	Reserved
18	RWS	0	**IO9DetMsk:** Malformed TLP Status
17	RWS	0	**IO10DetMsk:** Receiver Buffer Overflow Status
16	RWS	0	**IO8DetMsk:** Unexpected Completion Status
15	RWS	0	**IO7DetMsk:** Completer Abort Status
14	RWS	0	**IO6DetMsk:** Completion Time-out Status
13	RWS	0	**IO5DetMsk:** Flow Control Protocol Error Status

Device: 0 Function: 0 Offset: 148h			
Bit	**Attr**	**Default**	**Description**
12	RWS	0	**IO4DetMsk:** Poisoned TLP Status
11:5	RV	0h	Reserved
4	RWS	0	**IO0DetMsk:** Data Link Protocol Error Status
3:0	RV	0h	Reserved

3.8.12.21 EMASK_UNCOR_PEX[9:1] - 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.

Device: 9 - 1 Function: 0 Offset: 148h			
Bit	**Attr**	**Default**	**Description**
31:22	RV	0h	Reserved
21	RV	0	**Reserved. ACS Violation Status (Unsupported Feature)**
20	RWS	0	**IO2DetMsk:** Received an Unsupported Request
19	RV	0	Reserved
18	RWS	0	**IO9DetMsk:** Malformed TLP Status
17	RWS	0	**IO10DetMsk:** Receiver Buffer Overflow Status
16	RWS	0	**IO8DetMsk**: Unexpected Completion Status
15	RWS	0	**IO7DetMsk:** Completer Abort Status
14	RWS	0	**IO6DetMsk**: Completion Time-out Status
13	RWS	0	**IO5DetMsk:** Flow Control Protocol Error Status
12	RWS	0	**IO4DetMsk:** Poisoned TLP Status
11:6	RV	0h	Reserved
5	RWS	0	IO19DetMsk: Surprise Link Down Mask
4	RWS	0	**IO0DetMsk:** Data Link Protocol Error Status
3:0	RV	0h	Reserved

3.8.12.22 EMASK_COR_PEX[9:0] - 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: 9 - 0 Function: 0 Offset: 14Ch			
Bit	**Attr**	**Default**	**Description**
31:14	RV	0h	Reserved
13	RWS	0	**IO20DetMsk: Advisory Non-Fatal Error Detect Mask**
12	RWS	0	**IO16DetMsk:** Replay Timer Time-out Mask
11:9	RV	0h	Reserved

Device: 9 - 0 Function: 0 Offset: 14Ch			
Bit	**Attr**	**Default**	**Description**
8	RWS	0	**IO15DetMsk:** Replay_Num Rollover Mask
7	RWS	0	**IO14DetMsk:** Bad DLLP Mask
6	RWS	0	**IO13DetMsk:** Bad TLP Mask
5:1	RV	0h	Reserved
0	RWS	0	**IO12DetMsk:** Receiver Error Mask

3.8.12.23 EMASK_RP_PEX[9:0] - 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: 9 - 0 Function: 0 Offset: 150h			
Bit	**Attr**	**Default**	**Description**
31:3	RV	0h	Reserved
2	RWS	0	**IO1DetMsk:** Fatal Message Detect Mask
1	RWS	0	**IO11DetMsk:** Uncorrectable Message Detect Mask
0	RWS	0	**IO17DetMsk:** Correctable Message Detect Mask

3.8.12.24 PEX_FAT_FERR[9:0] - PCI Express First Fatal Error Register

This register records the occurrence of the first unmasked PCI Express FATAL errors and written by the MCH if the respective bits are not set prior. These errors are written by the MCH if the respective bits are not set prior. 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 described in Section 3.8.12.6.

Device: 9 - 0 Function: 0 Offset: 154h			
Bit	**Attr**	**Default**	**Description**
31:24	RV	0h	*Reserved*
23	RW1CS	0	**First_FAT_Err_IO32: Received CA Posted Request**
22	RW1CS	0	**First_FAT_Err_IO31: Received UR Posted Request**
21	RW1CS	0	**First_FAT_Err_IO30: VT-d Internal HW**
20	RW1CS	0	**First_FAT_Err_IO29: PEX - MSI Address**
19	RV	0	**Reserved**
18	RW1CS	0	**First_FAT_Err_IO27: PEX - Stop & Scream**
17	RW1CS	0	**First_FAT_Err_IO26: PEX - Received CA response**
16	RW1CS	0	**First_FAT_Err_IO25: PEX - Received UR Response**
15	RW1CS	0	**First_FAT_Err_IO24: PEX - Outbound poisoned TLP**
14	RW1CS	0	**First_FAT_Err_IO23: PEX - VT-d Fault**

Device: 9 - 0 Function: 0 Offset: 154h			
Bit	**Attr**	**Default**	**Description**
13	RW1CS	0	**First_FAT_Err_IO22: PEX - Internal Header/Control Parity**
12	if (port 0) {RW1CS} else {RV} endif	0	**First_FAT_Err_IO18: ESI Reset timeout (ESI port only)**
11	RV	0	**Reserved**
10	RW1CS	0	First_FAT_Err_IO1: PEX - Received Fatal Error Message
9	RW1CS	0	First_FAT_Err_IO2: PEX - Received Unsupported Request (inbound)
8	RW1CS	0	First_FAT_Err_IO9: PEX - Malformed TLP
7	RW1CS	0	First_FAT_Err_IO10: PEX - Receive Buffer Overflow Error
6	RW1CS	0	First_FAT_Err_IO8: PEX - Unexpected Completion Error
5	RW1CS	0	First_FAT_Err_IO7: PEX - Completer Abort (Inbound)
4	RW1CS	0	First_FAT_Err_IO6: PEX - Completion Timeout
3	RW1CS	0	First_FAT_Err_IO5: PEX - Flow Control Protocol Error
2	RW1CS	0	First_FAT_Err_IO4: PEX - Poisoned TLP (Inbound)
1	RW1CS	0	**First_FAT_Err_IO19: Surprise Link Down**
0	RW1CS	0	First_FAT_Err_IO0: PEX - Data Link Layer Protocol Error

3.8.12.25 PEX_NF_COR_FERR[9:0] - 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 the MCH if the respective bits are not set prior. 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 described in Section 3.8.12.6

Device: 9 - 0 Function: 0 Offset: 158h			
Bit	**Attr**	**Default**	**Description**
31:30	RV	0h	*Reserved*
29	RW1CS	0	**First_NFAT_COR_Err_IO33: PEX - Link Autonomous Bandwidth Change (correctable)**
28	RW1CS	0	**First_NFAT_COR_Err_IO32: PEX - Received CA Posted Request (uncorrectable)**
27	RW1CS	0	**First_NFAT_COR_Err_IO31: PEX - Received UR Posted Request (uncorrectable)**
26	RW1CS	0	**First_NFAT_COR_Err_IO30: PEX - VT-d Internal HW (uncorrectable)**
25	RW1CS	0	**First_NFAT_COR_Err_IO29: PEX - MSI Address (uncorrectable)**
24	RW1CS	0	**First_NFAT_COR_Err_IO28: PEX - Link BW Change (correctable)**
23	RW1CS	0	**First_NFAT_COR_Err_IO27: PEX - Stop & Scream (uncorrectable)**
22	RW1CS	0	**First_NFAT_COR_Err_IO26: PEX - Received CA response (uncorrectable)**

Device: 9 - 0 Function: 0 Offset: 158h			
Bit	**Attr**	**Default**	**Description**
21	RW1CS	0	**First_NFAT_COR_Err_IO25: PEX - Received UR Response (uncorrectable)**
20	RW1CS	0	**First_NFAT_COR_Err_IO24: PEX - Outbound poisoned TLP (uncorrectable)**
19	RW1CS	0	**First_NFAT_COR_Err_IO23: PEX - VT-d Fault (uncorrectable)**
18	RW1CS	0	First_NFAT_COR_Err_IO11: PEX - Received Non Fatal (uncorrectable) Error Message
17	RW1CS	0	First_NFAT_COR_Err_IO17: PEX - Received Correctable Error Message
16	RV	0	*Reserved*
15	RW1CS	0	First_NFAT_COR_Err_IO2: PEX - Received Unsupported Request (uncorrectable)
14	RW1CS	0	First_NFAT_COR_Err_IO9: PEX -Malformed TLP (uncorrectable)
13	RW1CS	0	First_NFAT_COR_Err_IO10: PEX - Receive Buffer Overflow Error (uncorrectable)
12	RW1CS	0	First_NFAT_COR_Err_IO8: PEX - Unexpected Completion Error (correctable as defined by PCIE spec v1.1 or later)
11	RW1CS	0	First_NFAT_COR_Err_IO7: PEX - Completer Abort (uncorrectable)
10	RW1CS	0	First_NFAT_COR_Err_IO6: PEX - Completion Timeout (uncorrectable)
9	RW1CS	0	First_NFAT_COR_Err_IO5: PEX - Flow Control Protocol Error (uncorrectable)
8	RW1CS	0	First_NFAT_COR_Err_IO4: PEX - Poisoned TLP (correctable as determined by MCH behavior and PCIE spec v1.1 or later)
7	RW1CS	0	**First_NFAT_COR_Err_IO19: PEX - Surprise Link Down (uncorrectable)**
6	RW1CS	0	First_NFAT_COR_Err_IO0: PEX - Data Link Layer Protocol Error (uncorrectable)
5	RW1CS	0	**First_NFAT_COR_Err_IO20: PEX - Advisory Non-Fatal Error (correctable)**
4	RW1CS	0	First_NFAT_COR_Err_IO16: PEX - Replay Timer Timeout (correctable)
3	RW1CS	0	First_NFAT_COR_Err_IO15: PEX - Replay_Num Rollover (correctable)
2	RW1CS	0	First_NFAT_COR_Err_IO14: PEX - BAD DLLP Error (correctable)
1	RW1CS	0	First_NFAT_COR_Err_IO13: PEX - Bad TLP Error (correctable)
0	RW1CS	0	First_NFAT_COR_Err_IO12: PEX - Receiver Error (correctable)

3.8.12.26 PEX_FAT_NERR[9:0] - PCI Express Next Fatal Error Register

This register records the subsequent occurrences after the first unmasked PCI Express FATAL errors and written by the MCH if the respective bits are set in the PEX_FERR_FAT register.

Device: 9 - 0 Function: 0 Offset: 15Ch			
Bit	**Attr**	**Default**	**Description**
31:24	RV	0h	*Reserved*
23	RW1CS	0	**Next_FAT_Err_IO32: PEX - Received CA Posted Request**

Device: 9 - 0 Function: 0 Offset: 15Ch			
Bit	**Attr**	**Default**	**Description**
22	RW1CS	0	**Next_FAT_Err_IO31: PEX - Received UR Posted Request**
21	RW1CS	0	**Next_FAT_Err_IO30: PEX - VT-d Internal HW**
20	RW1CS	0	**Next_FAT_Err_IO29: PEX - MSI Address**
19	RV	0	*Reserved*
18	RW1CS	0	**Next_FAT_Err_IO27: PEX - Stop & Scream**
17	RW1CS	0	**Next_FAT_Err_IO26: PEX - Received CA response**
16	RW1CS	0	**Next_FAT_Err_IO25: PEX - Received UR Response**
15	RW1CS	0	**Next_FAT_Err_IO24: PEX - Outbound poisoned TLP**
14	RW1CS	0	**Next_FAT_Err_IO23: PEX - VT-d Fault**
13	RW1CS	0	**Next_FAT_Err_IO22: PEX - Internal Header/Control Parity**
12	if (port 0) {RW1CS} else {RV} endif	0	**Next_FAT_Err_IO18: ESI - ESI Reset timeout (ESI port only)**
11	RV	0	*Reserved*
10	RW1CS	0	**Next_FAT_Err_IO1: PEX - Received Fatal Error Message**
9	RW1CS	0	**Next_FAT_Err_IO2: PEX - Received Unsupported Request**
8	RW1CS	0	**Next_FAT_Err_IO9: PEX - Malformed TLP**
7	RW1CS	0	**Next_FAT_Err_IO10: PEX - Receive Buffer Overflow Error**
6	RW1CS	0	**Next_FAT_Err_IO8: PEX - Unexpected Completion Error**
5	RW1CS	0	Next_FAT_Err_IO7: PEX - Completer Abort
4	RW1CS	0	Next_FAT_Err_IO6: PEX - Completion Timeout
3	RW1CS	0	Next_FAT_Err_IO5: PEX - Flow Control Protocol Error
2	RW1CS	0	Next_FAT_Err_IO4: PEX - Poisoned TLP
1	RW1CS	0	Next_FAT_Err_IO19: PEX - Surprise Link Down
0	RW1CS	0	Next_FAT_Err_IO0: PEX - Data Link Layer Protocol Error

3.8.12.27 PEX_NF_COR_NERR[9:0] - PCI Express Non Fatal or Correctable Next Error Register

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

Device: 9 - 0 Function: 0 Offset: 160h			
Bit	**Attr**	**Default**	**Description**
31:30	RV	0h	*Reserved*
29	RW1CS	0	**Next_NFAT_COR_Err_IO33: PEX - Link Autonomous Bandwidth Change (correctable)**
28	RW1CS	0	**Next_NFAT_COR_Err_IO32: PEX - Received CA Posted Request (uncorrectable)**

Device: 9 - 0
Function: 0
Offset: 160h

Bit	Attr	Default	Description
27	RW1CS	0	**Next_NFAT_COR_Err_IO31: PEX - Received UR Posted Request (uncorrectable)**
26	RW1CS	0	**Next_NFAT_COR_Err_IO30: PEX - VT-d Internal HW (uncorrectable)**
25	RW1CS	0	**Next_NFAT_COR_Err_IO29: PEX - MSI Address (uncorrectable)**
24	RW1CS	0	**Next_NFAT_COR_Err_IO28: PEX - Link BW Change (correctable)**
23	RW1CS	0	**Next_NFAT_COR_Err_IO27: PEX - Stop & Scream (uncorrectable)**
22	RW1CS	0	**Next_NFAT_COR_Err_IO26: PEX - Received CA response (uncorrectable)**
21	RW1CS	0	**Next_NFAT_COR_Err_IO25: PEX - Received UR Response (uncorrectable)**
20	RW1CS	0	**Next_NFAT_COR_Err_IO24: PEX - Outbound poisoned TLP (uncorrectable)**
19	RW1CS	0	**Next_NFAT_COR_Err_IO23: PEX - VT-d Fault (uncorrectable)**
18	RW1CS	0	Next_NFAT_COR_Err_IO11: PEX - Received Non Fatal (uncorrectable) Error Message
17	RW1CS	0	Next_NFAT_COR_Err_IO17: PEX - Received Correctable Error Message
16	RV	0	Reserved
15	RW1CS	0	Next_NFAT_COR_Err_IO2: PEX - Received Unsupported Request (uncorrectable)
14	RW1CS	0	Next_NFAT_COR_Err_IO9: PEX - Malformed TLP (uncorrectable)
13	RW1CS	0	Next_NFAT_COR_Err_IO10: PEX - Receive Buffer Overflow Error (uncorrectable)
12	RW1CS	0	Next_NFAT_COR_Err_IO8: PEX - Unexpected Completion Error (correctable as defined by PCIE spec v1.1 or later)
11	RW1CS	0	Next_NFAT_COR_Err_IO7: PEX - Completer Abort (uncorrectable)
10	RW1CS	0	Next_NFAT_COR_Err_IO6: PEX - Completion Timeout (uncorrectable)
9	RW1CS	0	Next_NFAT_COR_Err_IO5: PEX - Flow Control Protocol Error (uncorrectable)
8	RW1CS	0	Next_NFAT_COR_Err_IO4: PEX - Poisoned TLP (correctable as determined by MCH behavior and PCIE spec v1.1 or later)
7	RW1CS	0	**Next_NFAT_COR_Err_IO19: PEX - Surprise Link Down (uncorrectable)**
6	RW1CS	0	Next_NFAT_COR_Err_IO0: PEX - Data Link Layer Protocol Error (uncorrectable)
5	RW1CS	0	**Next_NFAT_COR_Err_IO20: PEX - Advisory Non-Fatal Error (correctable)**
4	RW1CS	0	Next_NFAT_COR_Err_IO16: PEX - Replay Timer Timeout (correctable)
3	RW1CS	0	Next_NFAT_COR_Err_IO15: PEX - Replay_Num Rollover (correctable)
2	RW1CS	0	Next_NFAT_COR_Err_IO14: PEX - BAD DLLP Error (correctable)
1	RW1CS	0	Next_NFAT_COR_Err_IO13: PEX - Bad TLP Error (correctable)
0	RW1CS	0	Next_NFAT_COR_Err_IO12: PEX - Receiver Error (correctable)

3.8.12.28 EMASK_UNIT_PEX[9:0]: Unit Error Detect Mask

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

Device: 9 - 0
Function: 0
Offset: 164h

Bit	Attr	Default	Description
31:14	RV	0h	*Reserved*
13	RWS	0	**IO33DetMsk: Link Autonomous BW change Error Detect Mask**
12	RWS	0	**IO32DetMsk: Received CA Posted Request Error Detect Mask**
11	RWS	0	**IO31DetMsk: Received UR Posted Request Error Detect Mask**
10	RWS	0	**IO30DetMsk: VT-d Internal HW Error Detect Mask**
9	RWS	0	**IO29DetMsk: MSI Address Error Detect Mask**
8	RWS	0	**IO28DetMsk: Link BW Change Error Detect Mask**
7	RWS	0	**IO27DetMsk: Stop & Scream Error Detect Mask**
6	RWS	0	**IO26DetMsk: Received CA Response Detect Mask**
5	RWS	0	**IO25DetMsk: Received UR Response Detect Mask**
4	RWS	0	**IO24DetMsk: Outbound Poisoned Data Detect Mask**
3	RWS	0	**IO23DetMsk: VTd Fault Error Detect Mask**
2	RWS	0	**IO22DetMsk: Internal Header/Control Parity Error Detect Mask**
1	RWS	0	**IO18DetMsk: ESI Reset Timeout Error Detect Mask**
0	RO	0	**VPPErrDetMsk: Correctable Message Detect Mask**

3.8.12.29 PEX_UNIT_STS[9:0] - PCI Express Unit Error Status

This register records the occurrence of the unit errors that are specific to this PCI Express port caused by external activities. The unit errors are sent to the Coherency Engine to classify as to which port cluster it came from ports 1 - 4 or ports 5 - 8 or port 9 and the errors are recorded in Coherency Engine and appropriate interrupts generated through ERR pins.

Device: 9 - 0
Function: 0
Offset: 168h

Bit	Attr	Default	Description
31:14	RV	0h	*Reserved*
13	RW1CS	0	**IO33Err: Link Autonomous Bandwidth Change Error**
12	RW1CS	0	**IO32Err: Received CA Posted Request Error**
11	RW1CS	0	**IO31Err: Received UR Posted Request Error**
10	RW1CS	0	**IO30Err: VTd Internal HW Error**
9	RW1CS	0	**IO29Err: MSI Address Error**
8	RW1CS	0	**IO28Err: Link BW Change Error**
7	RW1CS	0	**IO27Err: Stop & Scream Error**
6	RW1CS	0	**IO26Err: Received CA Response Error**
5	RW1CS	0	**IO25Err: Received UR Response Error**

Device: 9 - 0 Function: 0 Offset: 168h			
Bit	**Attr**	**Default**	**Description**
4	RW1CS	0	**IO24Err: Outbound Poisoned Data Error**
3	RW1CS	0	**IO23Err: VTd Fault Error**
2	RW1CS	0	**IO22Err: Internal Header/Control Parity Error**
1	RW1CS	0	**IO18Err: ESI Reset Timeout Error**
0	RW1CS	0	**VPPErr: VPP Error**

3.8.12.30 PEX_UNIT_SEV[9:0]: PCI Express Unit Severity Register

This register set the PCIE Express unit error severity as Fatal or non-Fatal.

Device: 9 - 0 Function: 0 Offset: 16Ch			
Bit	**Attr**	**Default**	**Description**
31:14	RV	0h	*Reserved*
13	RV	0	*Reserved*
12	RWS	0	**IO32Severity: Received CA Posted Request Error Detect Severity**
11	RWS	0	**IO31Severity: Received UR Posted Request Error Detect Severity**
10	RWS	0	**IO30Severity: VTd Internal HW Error Detect Severity**
9	RWS	0	**IO29Severity: MSI Address Error Detect Severity**
8	RV	0	*Reserved*
7	RWS	0	**IO27Severity: Stop & Scream Error Detect Severity**
6	RWS	0	**IO26Severity: Received CA Response Detect Severity**
5	RWS	0	**IO25Severity: Received UR Response Detect Severity**
4	RWS	0	**IO24Severity: Outbound Poisoned Data Detect Severity**
3	RWS	0	**IO23Severity: VTd Fault Error Detect Severity**
2	RO	1	**IO22Severity: Internal Header/Control Parity Error Detect Severity**
1	RO	1	**IO18Severity: ESI Reset Timeout Error Severity**
0	RO	1	**VPPErrSeverity: VPP Error Severity**

3.8.12.31 PEX_UNIT_MASK[9:0]: Unit Error Report Mask

This register masks (blocks) the reporting of the selected bits associated with the unit errors. The are still detected. The VPP Error is never reported.

Device: 9 - 0 Function: 0 Offset: 170h			
Bit	**Attr**	**Default**	**Description**
31:14	RV	0h	*Reserved*
13	RWS	0	**IO33Msk: Link Autonomous Bandwidth Change Error Mask**
12	RWS	0	**IO32Msk: Received CA Posted Request Error Mask**
11	RWS	0	**IO31Msk: Received UR Posted Request Error Mask**

Device: 9 - 0 Function: 0 Offset: 170h			
Bit	**Attr**	**Default**	**Description**
10	RWS	0	**IO30Msk: VTd Internal HW Error Mask**
9	RWS	0	**IO29Msk: MSI Address Error Mask**
8	RWS	0	**IO28Msk: Link BW Change Error Mask**
7	RWS	0	**IO27Msk: Stop & Scream Error Mask**
6	RWS	0	**IO26Msk: Non-Fatal Received CA Response Error Mask**
5	RWS	0	**IO25Msk: Non-Fatal Received UR Response Error Mask**
4	RWS	0	**IO24Msk: Non-Fatal Outbound Poisoned Data Error Mask**
3	RWS	0	**IO23Msk: Non-Fatal VTd Fault Error Mask**
2	RWS	0	**IO22Msk: Fatal Internal Header/Control Parity Error Mask**
1	RWS	0	**IO18Msk: Fatal ESI Reset Timeout Error Mask**
0	RO	1	**VPPErrDetMsk: VPP Error Mask**

3.8.12.32 PEX_UNIT_RPTCTRL[9:0]: PCI Express Unit Error Report Control and Status Register

This register controls and records the occurrence of PCI Express Unit errors.

Device: 9 - 0 Function: 0 Offset: 174h			
Bit	**Attr**	**Default**	**Description**
31:11	RV	0h	*Reserved*
10	RW	0h	USEFEEN: Unit System Error on Fatal Error Enable This field controls generation of system errors in the PCI Express port for fatal unit errors. 1: Indicates that a System Error should be generated if a fatal unit error is reported by this PCI Express port. 0: No System Error should be generated on a fatal unit error reported by the PCI Express port.
9	RW	0h	USENFEEN: Unit System Error on Non-Fatal Error Enable This field controls generation of system errors in the PCI Express port for non-fatal unit errors. 1: Indicates that a System Error should be generated if a non-fatal unit error is reported by this PCI Express port. 0: No System Error should be generated on a non-fatal unit error reported by the PCI Express port.
8	RW	0h	USECOREEN: Unit System Error on Correctable Error Enable This field controls generation of system errors in the PCI Express port for Correctable unit errors. 1: Indicates that a System Error should be generated if a correctable unit error is reported by this PCI Express port. 0: No System Error should be generated on a correctable unit error reported by the PCI Express port.

Device: 9 - 0 Function: 0 Offset: 174h			
Bit	**Attr**	**Default**	**Description**
7	RW1CS	0h	Unit_COR_ERR_Rcvd: Correctable Unit Error Received Set when one or more Correctable Unit errors have been received.
6	RW1CS	0h	Unit_FAT_ERR_Rcvd: Fatal Unit Error Received Set when one or more Fatal Uncorrectable Unit errors have been received.
5	RW1CS	0h	Unit_NFAT_ERR_Rcvd: Non-Fatal Unit Error Received Set when one or more Non-Fatal Uncorrectable Unit errors have been received.
4:0	ROS	0	UFERRPTR: Unit First Error Pointer The First Error pointer is a read-only field that identifies the bit position of the first unit error reported in the PEX_UNIT_STS register. Left most error bit if multiple bits occurred simultaneously.

3.8.13 Error Registers

This section describes the registers that record the first and next errors, logging, detection masks, signalling masks, and error injection control. The FERR_GLOBAL (first error register) is used to record the first error condition. The NERR_GLOBAL register is used to record subsequent errors.

The contents of FERR_GLOBAL and NERR_GLOBAL are “sticky” across a reset (while PWRGOOD remains asserted). This provides the ability for firmware to perform diagnostics across reboots.

3.8.13.1 FERR_GLOBAL - Global First Error Register

The first fatal and/or first non-fatal errors are flagged in the FERR_GLOBAL register, subsequent errors are indicated in the NERR_GLOBAL register.

Device: 16 Function: 2 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
31	RW1CS	0	Global_FERR_31 Internal MCH Fatal Error
30	RW1CS	0	Global_FERR_30 Intel® QuickData Technology Device Fatal Error
29	RW1CS	0	Global_FERR_29 FSB1 Fatal Error
28	RW1CS	0	Global_FERR_28 FSB 0 Fatal Error
27	RW1CS	0	Global_FERR_27 FB-DIMM Channel 0,1,2, or 3 Fatal Error
26	RW1CS	0	Global_FERR_26 Thermal Fatal Error
25	RW1CS	0	Global_FERR_25 PCI Express Device 9 Fatal Error
24	RW1CS	0	Global_FERR_24 PCI Express Device 8 Fatal Error

Device: 16
Function: 2
Offset: 40h

Bit	Attr	Default	Description
23	RW1CS	0	Global_FERR_23 PCI Express Device 7 Fatal Error
22	RW1CS	0	Global_FERR_22 PCI Express Device 6 Fatal Error
21	RW1CS	0	Global_FERR_21 PCI Express Device 5 Fatal Error
20	RW1CS	0	Global_FERR_20 PCI Express Device 4 Fatal Error
19	RW1CS	0	Global_FERR_19 PCI Express Device 3 Fatal Error
18	RW1CS	0	Global_FERR_18 PCI Express Device 2 Fatal Error
17	RW1CS	0	Global_FERR_17 PCI Express Device 1 Fatal Error
16	RW1CS	0	Global_FERR_16 ESI Fatal Error
15	RW1CS	0	Global_FERR_15 Internal MCH Non-Fatal Error
14	RW1CS	0	Global_FERR_14 Intel® QuickData Technology Device Non Fatal Error
13	RW1CS	0	Global_FERR_13 FSB1 Non-Fatal Error
12	RW1CS	0	Global_FERR_12 FSB 0 Non-Fatal Error
11	RW1CS	0	Global_FERR_11 FB-DIMM Channel 0,1,2, or 3 Non-Fatal Error
10	RW1CS	0	Global_FERR_10 Thermal Non-Fatal Error
9	RW1CS	0	Global_FERR_09 PCI Express Device 9 Non-Fatal Error
8	RW1CS	0	Global_FERR_08 PCI Express Device 8 Non-Fatal Error
7	RW1CS	0	Global_FERR_07 PCI Express Device 7 Non-Fatal Error
6	RW1CS	0	Global_FERR_06 PCI Express Device 6 Non-Fatal Error
5	RW1CS	0	Global_FERR_05 PCI Express Device 5 Non-Fatal Error
4	RW1CS	0	Global_FERR_04 PCI Express Device 4 Non-Fatal Error
3	RW1CS	0	Global_FERR_03 PCI Express Device 3 Non-Fatal Error
2	RW1CS	0	Global_FERR_02 PCI Express Device 2 Non-Fatal Error
1	RW1CS	0	Global_FERR_01 PCI Express Device 1 Non-Fatal Error

Device: 16 Function: 2 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
0	RW1CS	0	Global_FERR_00 ESI Non-Fatal Error

3.8.13.2 NERR_GLOBAL - Global Next Error Register

Once an error has been logged in the FERR_GLOBAL, subsequent errors are logged in the NERR_GLOBAL register.

Device: 16 Function: 2 Offset: 44h			
Bit	**Attr**	**Default**	**Description**
31	RW1CS	0	Global_NERR_31 Internal MCH Fatal Error
30	RW1CS	0	Global_NERR_30 Intel® QuickData Technology Device Fatal Error
29	RW1CS	0	Global_NERR_29 FSB1 Fatal Error
28	RW1CS	0	Global_NERR_28 FSB 0 Fatal Error
27	RW1CS	0	Global_NERR_27 FB-DIMM Channel 0,1,2 or 3 Fatal Error
26	RV	0	Global_NERR_26 RESERVED Fatal Error
25	RW1CS	0	Global_NERR_25 PCI Express Device 9 Fatal Error
24	RW1CS	0	Global_NERR_24 PCI Express Device 8 Fatal Error
23	RW1CS	0	Global_NERR_23 PCI Express Device 7 Fatal Error
22	RW1CS	0	Global_NERR_22 PCI Express Device 6 Fatal Error
21	RW1CS	0	Global_NERR_21 PCI Express Device 5 Fatal Error
20	RW1CS	0	Global_NERR_20 PCI Express Device 4 Fatal Error
19	RW1CS	0	Global_NERR_19 PCI Express Device 3 Fatal Error
18	RW1CS	0	Global_NERR_18 PCI Express Device 2 Fatal Error
17	RV	0	Global_NERR_17 PCI Express Device 1 Fatal Error
16	RW1CS	0	Global_NERR_16 ESI Fatal Error
15	RW1CS	0	Global_NERR_15 Internal MCH Non-Fatal Error

Device: 16 Function: 2 Offset: 44h			
Bit	**Attr**	**Default**	**Description**
14	RW1CS	0	Global_NERR_14 Intel® QuickData Technology Device Non Fatal Error
13	RW1CS	0	Global_NERR_13 FSB1 Non-Fatal Error
12	RW1CS	0	Global_NERR_12 FSB 0 Non-Fatal Error
11	RW1CS	0	Global_NERR_11 FB-DIMM Channel 0,1, 2 or 3 Non-Fatal Error
10	RV	0	Global_NERR_10 RESERVED Non-Fatal Error
9	RW1CS	0	Global_NERR_09 PCI Express Device 9 Non-Fatal Error
8	RW1CS	0	Global_NERR_08 PCI Express Device 8 Non-Fatal Error
7	RW1CS	0	Global_NERR_07 PCI Express Device 7 Non-Fatal Error
6	RW1CS	0	Global_NERR_06 PCI Express Device 6 Non-Fatal Error
5	RW1CS	0	Global_NERR_05 PCI Express Device 5 Non-Fatal Error
4	RW1CS	0	Global_NERR_04 PCI Express Device 4 Non-Fatal Error
3	RW1CS	0	Global_NERR_03 PCI Express Device 3 Non-Fatal Error
2	RW1CS	0	Global_NERR_02 PCI Express Device 2 Non-Fatal Error
1	RW1CS	0	Global_NERR_01 PCI Express Device 1 Non-Fatal Error
0	RW1CS	0	Global_NERR_00 ESI Non-Fatal Error

3.8.13.3 FERR_FAT_FSB[1:0]: FSB First Fatal Error Register

Device: 16 Function: 0 Offset: 480h, 180h			
Bit	**Attr**	**Default**	**Description**
7:6	RV	00	Reserved
5	RW1CS	0	**F9Err:** FSB protocol Error
4	RV	0h	Reserved
3	RW1CS	0	**F2Err:** Unsupported Processor Bus Transaction
2:1	RV	0h	Reserved
0	RW1CS	0	**F1Err:** Request/Address Parity Error

3.8.13.4 FERR_NF_FSB[1:0]: FSB First Non-Fatal Error Register

Device: 16
Function: 0
Offset: 481h, 181h

Bit	Attr	Default	Description
7:3	RV	00000	Reserved
2	RW1CS	0	**F7Err:** Detected MCERR from a processor
1	RW1CS	0	**F8Err:** Detected BINIT from a processor
0	RW1CS	0	**F6Err:** Parity Error in Data from FSB Interface

3.8.13.5 NERR_FAT_FSB[1:0]: FSB Next Fatal Error Register

This register logs all FSB subsequent errors after the FERR_FAT_FSB has logged the 1st fatal error.

Device: 16
Function: 0
Offset: 482h, 182h

Bit	Attr	Default	Description
7:6	RV	00	Reserved
5	RW1CS	0	**F9Err:** FSB protocol Error
4	RV	0h	Reserved
3	RW1CS	0	**F2Err:** Unsupported Processor Bus Transaction
2:1	RV	0h	Reserved
0	RW1CS	0	**F1Err:** Request/Address Parity Error

3.8.13.6 NERR_NF_FSB[1:0]: 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: 16
Function: 0
Offset: 483h, 183h

Bit	Attr	Default	Description
7:3	RV	00000	Reserved
2	RW1CS	0	**F7Err:** Detected MCERR from a processor
1	RW1CS	0	**F8Err:** Detected BINIT from a processor
0	RW1CS	0	**F6Err:** Parity Error in Data from FSB Interface

3.8.13.7 NRECFSB[1:0]: Non Recoverable 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: 16
Function: 0
Offset: 484h, 184h

Bit	Attr	Default	Description
31:29	RV	000	Reserved
28:24	ROS	00000	**REQA:** REQa[4:0] fields of the FSB
23:21	ROS	000	**REQB:** REQb[2:0] fields of the FSB
20:16	ROS	00000	**EXF:** EXF[4:0] fields of the FSB
15:8	ROS	00h	**ATTR:** ATTR[7:0] fields of the FSB
7:0	ROS	00h	**DID:** DID[7:0] fields of the FSB

3.8.13.8 RECFSB[1:0]: 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: 16
Function: 0
Offset: 488h, 188h

Bit	Attr	Default	Description
31:29	RV	000	Reserved
28:24	ROS	00000	**REQA:** REQa[4:0] fields of the FSB
23:21	ROS	000	**REQB:** REQb[2:0] fields of the FSB
20:16	ROS	00000	**EXF:** EXF[4:0] fields of the FSB
15:8	ROS	00h	**ATTR**: ATTR[7:0] fields of the FSB
7:0	ROS	00h	**DID:** DID[7:0] fields of the FSB

3.8.13.9 NRECADDRL[1:0]: Non Recoverable 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
Offset: 48Ch, 18Ch

Bit	Attr	Default	Description
31:4	ROS	0h	**A31DT4:** FSB Address [31:4]
3	ROS	0	**A3:** FSB Address [3]
2:0	RV	000	Reserved

3.8.13.10 NRECADDRH[1:0]: Non Recoverable 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: 16
Function: 0
Offset: 490h, 190h

Bit	Attr	Default	Description
7:0	ROS	00h	**A39DT32:** FSB Address [37:32]

3.8.13.11 EMASK_FSB[1:0]: FSB Error Mask Register

A '0' in any field enables that error.

Device: 16
Function: 0
Offset: 492h, 192h

Bit	Attr	Default	Description
15:9	RV	0h	Reserved
8	RWS	1	**F9Msk:** FSB Protocol Error
7	RWS	1	**F8Msk:** B-INIT
6	RWS	1	**F7Msk:** Detected MCERR
5	RWS	1	**F6Msk:** Data Parity Error
4	RV	0h	Reserved
3	RV	0h	Reserved
2	RV	0h	Reserved
1	RWS	1	**F2Msk:** Unsupported Processor Bus Transaction
0	RWS	1	**F1Msk:** Request/Address Parity Error

3.8.13.12 ERR2_FSB[1:0]: 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: 16
Function: 0
Offset: 498h, 198h

Bit	Attr	Default	Description
15:9	RV	0h	Reserved
8	RW	1	**F9Msk:** FSB Protocol Error
7	RW	1	**F8Msk:** B-INIT
6	RW	1	**F7Msk:** Detected MCERR
5	RW	1	**F6Msk:** Data Parity Error
4	RV	0	Reserved
3	RV	0h	Reserved

Device: 16 Function: 0 Offset: 498h, 198h			
Bit	**Attr**	**Default**	**Description**
2	RV	0h	Reserved
1	RW	1	**F2Msk:** Unsupported Processor Bus Transaction
0	RW	1	**F1Msk:** Request/Address Parity Error

3.8.13.13 ERR1_FSB[1:0]: 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: 16 Function: 0 Offset: 496h, 196h			
Bit	**Attr**	**Default**	**Description**
15:9	RV	0h	Reserved
8	RW	1	**F9Msk:** FSB Protocol Error
7	RW	1	**F8Msk:** B-INIT
6	RW	1	**F7Msk:** Detected MCERR
5	RW	1	**F6Msk:** Data Parity Error
4	RV	0h	Reserved
3	RV	0h	Reserved
2	RV	0h	Reserved
1	RW	1	**F2Msk:** Unsupported Processor Bus Transaction
0	RW	1	**F1Msk:** Request/Address Parity Error

3.8.13.14 ERR0_FSB[1:0]: 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: 16 Function: 0 Offset: 494h, 194h			
Bit	**Attr**	**Default**	**Description**
15:9	RV	0h	Reserved
8	RW	1	**F9Msk:** FSB Protocol Error
7	RW	1	**F8Msk:** B-INIT
6	RW	1	**F7Msk:** Detected MCERR
5	RW	1	**F6Msk:** Data Parity Error
4	RV	0h	Reserved
3	RV	0h	Reserved
2	RV	0h	Reserved

Device: 16 Function: 0 Offset: 494h, 194h			
Bit	**Attr**	**Default**	**Description**
1	RW	1	**F2Msk:** Unsupported Processor Bus Transaction
0	RW	1	**F1Msk:** Request/Address Parity Error

3.8.13.15 MCERR_FSB[1:0]: 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.

Device: 16 Function: 0 Offset: 49Ah, 19Ah			
Bit	**Attr**	**Default**	**Description**
15:9	RV	0h	Reserved
8	RW	1	**F9Msk:** FSB Protocol Error
7	RW	1	**F8Msk:** B-INIT
6	RW	1	**F7Msk:** Detected MCERR
5	RW	1	**F6Msk:** Data Parity Error
4	RV	0h	Reserved
3	RV	0h	Reserved
2	RV	0h	Reserved
1	RW	1	**F2Msk:** Unsupported Processor Bus Transaction
0	RW	1	**F1Msk:** Request/Address Parity Error

3.8.13.16 NRECSF0 - Non-Recoverable Error Control Information of Snoop Filter 0

Lower 32 bits of the Conceptual 64-bit non-recoverable error register.

Device: 16 Function: 2 Offset: B0h			
Bit	**Attr**	**Default**	**Description**
31:15	ROS	0000h	Tag(A[34:18]): Lower 17 bits of 22 bit Tag address.
14:3	ROS	000h	Set(A[17:6]): SF is organized in 4 K sets, indicated by 12 bits.
2	ROS	0	State: Coherency State 1: The cache line is in E/M state, either exclusive (clean) or modified (dirty). 0: The cache line is in non-E/M shared state.
1:0	ROS	00	Presence Vector: Bus Presence Vector 00: The entry is invalid, neither FSB has ownership, state should be 0. 01 = FSB0 has ownership (state could be 0 or 1) 10 = FSB1 has ownership (state could be 0 or 1) 11 = FSB0 & FSB1 have shared ownership (state must be 0)

3.8.13.17 NRECSF1: Non-Recoverable Error Control Information of Snoop Filter 1

Upper 32 bits of the conceptual 64-bit non-recoverable error register.

Device: 16 Function: 2 Offset: B4h			
Bit	**Attr**	**Default**	**Description**
31:13	RV	00000h	Reserved
12:11	ROS	00	Affinity[1:0]: Indicates one of the 4 affinities in SF. 00: Affinity 0 01: Affinity 1 10: Affinity 2 11: Affinity 3
10:6	ROS	00000	Way[4:0]: Indicates which way is implicated in the SF. 00000: Way 0 (Lowest Supported Way) ... 10111: Way 23 (Highest Supported Way) 11000: Way 24 (Illegal Way) ... 11111: Way 31 (Illegal Way)
5	ROS	0	Hit(1), Miss(0): Hit indicates that the entry is in the SF, Miss indicates that the entry is not in the SF
4:0	ROS	0h	Tag(A[39:35]): Upper 5 bits of 22 bit Tag address.

3.8.13.18 RECSF0 - Recoverable Error Control Information of Snoop Filter 0

Lower 32 bits of the conceptual 64-bit recoverable error register.

Device: 16 Function: 2 Offset: B8h			
Bit	**Attr**	**Default**	**Description**
31:15	ROS	0000h	Tag(A[34:19]): Lower 17 bits of 22 bit Tag address.
14:3	ROS	000h	Set(A[18:7]): SF is organized in 4 K sets, indicated by 12 bits.
2	ROS	0	State: Coherency State 1: The cache line is in E/M state, either exclusive (clean) or modified (dirty). 0: The cache line is in non-E/M shared state.
1:0	ROS	00	Presence Vector: Bus Presence Vector 00: The entry is invalid, neither FSB has ownership, state should be 0. 01 = FSB0 has ownership (state could be 0 or 1) 10 = FSB1 has ownership (state could be 0 or 1) 11 = FSB0 & FSB1 have shared ownership (state must be 0)

3.8.13.19 RECSF1: Recoverable Error Control Information of Snoop Filter 1

Upper 32 bits of the conceptual 64-bit recoverable error register.

Device: 16 Function: 2 Offset: BCh			
Bit	**Attr**	**Default**	**Description**
31:13	RV	00000h	Reserved
12:11	ROSTT	00	Affinity[1:0]: Indicates one of the 4 affinities in SF. 00: Affinity 0 01: Affinity 1 10: Affinity 2 11: Affinity 3
10:6	ROS	00000	Way[4:0]: Indicates which way is implicated in the SF. 24 legal values, 8 illegal values 00000: Way 0(Lowest Supported Way) ... 10111: Way 23 (Highest Supported Way) 11000: Way 24 (Illegal Way) ... 11111: Way 31 (Illegal Way)
5	ROS	0	Hit(1), Miss(0): Hit indicates that the entry is in the SF, Miss indicates that the entry is not in the SF.
4:0	ROS	0h	Tag(A[37:32]): Upper 5 bits of 22 bit Tag address.

3.8.13.20 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	RV	0	**B25Err**: Illegal HiSMM/TSEG access
6	RW1CS	0	B23Err: Vt Unaffiliated Port Error
5	RW1CS	0	B21Err: Illegal Way
4	RW1CS	0	**B7Err:** Multiple ECC error in any of the ways during SF lookup
3	RW1CS	0	B4Err: Virtual Pin Error
2	RW1CS	0	**B3Err:** Coherency Violation Error for EWB
1	RW1CS	0	**B2Err:** Multi-Tag Hit SF
0	RW1CS	0	**B1Err:** CE Parity Error

3.8.13.21 FERR_FAT_INT2: Internal 2nd (Orthogonal) First Fatal Error Register

FERR_FAT _INT2 latches the first orthogonal MCH internal fatal error. All subsequent orthogonal fatal errors get logged in the NERR_FAT_INT2.

Device: 16 Function: 2 Offset: C1h			
Bit	**Attr**	**Default**	**Description**
7:3	RV	00h	Reserved
2	RW1CS	0	B14Err: SF Scrub DBE
1	RV	0	Reserved
0	RW1CS	0	B12Err: Parity Protected Register Error

3.8.13.22 FERR_NF_INT - Internal First Non-Fatal Error Register

Device: 16 Function: 2 Offset: C2h			
Bit	**Attr**	**Default**	**Description**
7	RW1CS	0	B22Err: Victim ROM parity error)
6	RW1CS	0	B20Err: Configuration Write Abort (
5	RW1CS	0	B11Err: Victim RAM parity Error
4	RW1CS	0	B10Err: DM Parity Error
3	RW1CS	0	B9Err: Illegal Access Error Illegal 64b configuration access (to a 32-bit only register)
2	RW1CS	0	Reserved
1	RW1CS	0	B6Err: Single ECC error on SF lookup
0	RW1CS	0	B5Err: Address Map Error

3.8.13.23 FERR_NF_INT2: Internal 2nd (Orthogonal) First Non-Fatal Error Register

FERR_NF _INT2 latches the first MCH internal orthogonal fatal error. All subsequent orthogonal non-fatal errors get logged in the NERR_NF_INT2.

Device: 16 Function: 2 Offset: C3h			
Bit	**Attr**	**Default**	**Description**
7	RV	00	Reserved
6	RW1CS	0	B27Err: Request received when in S1(CE)
5	RW1CS	0	B24Err: DFXERR (DFX) - DFX Response function for error generation
4	RW1CS	0	B19Err: Scrub SBE (SF)
3	RW1CS	0	B18Err: Perfmon task completion - (Logical OR of PERFSTAT bits)
2	RW1CS	0	B17Err: JTAG/TAP error status (From JTAG register stack status bits)
1	RW1CS	0	B16Err: SMBus error status (From SMB register stack status bits)
0	RV	0h	Reserved

3.8.13.24 NERR_FAT_INT - Internal Next Fatal Error Register

Device: 16
Function: 2
Offset: C4h

Bit	Attr	Default	Description
7	RW1CS	0	B25ERR: Illegal TSEG access
6	RW1CS	0	B23Err: Vt Unaffiliated Port Error
5	RW1CS	0	B21Err: Illegal Way (SF)
4	RW1CS	0	B7Err: Multiple ECC error in any of the ways during SF lookup
3	RW1CS	0	B4Err: Virtual Pin Error (VPP_PEX)
2	RW1CS	0	B3Err: Coherency Violation Error for EWB
1	RW1CS	0	B2Err: Multi-Tag Hit SF
0	RW1CS	0	B1Err: CE Parity Error

3.8.13.25 NERR_FAT_INT2: Internal 2nd (Orthogonal) Next Fatal Error Register

FERR_FAT _INT2 latches the first orthogonal MCH internal fatal error. All subsequent orthogonal fatal errors get logged in the NERR_FAT_INT2.

Device: 16
Function: 2
Offset: C5h

Bit	Attr	Default	Description
7:3	RV	0h	Reserved
2	RW1CS	0	B14Err: SF Scrub DBE
1	RV	0h	Reserved
0	RW1CS	0	B12Err: Parity Protected Register Error

3.8.13.26 NERR_NF_INT - Internal Next Non-Fatal Error Register

Device: 16
Function: 2
Offset: C6h

Bit	Attr	Default	Description
7	RW1CS	0	B22Err: Victim ROM parity error (SF)
6	RW1CS	0	B20Err: Configuration Write Abort (CE).
5	RW1CS	0	B11Err: Victim RAM parity error (SF)
4	RW1CS	0	B10Err: DM Parity Error (DM)
3	RW1CS	0	B9Err: Illegal Access Error (COH)
2	RW1CS	0	**B26Err: Illegal Access to Non-Coherent Address Space (CE)**
1	RW1CS	0	**B6Err:** Single ECC error on SF lookup (SF)
0	RW1CS	0	**B5Err:** Address Map Error (COH)

3.8.13.27 NERR_NF_INT2: Internal 2nd (Orthogonal) Next Non-Fatal Error Register

FERR_NF_INT2 latches the first MCH internal orthogonal non-fatal error. All subsequent orthogonal non-fatal errors get logged in the NERR_NF_INT2.

Device: 16
Function: 2
Offset: C7h

Bit	Attr	Default	Description
7:5	RV	00	Reserved
4	RW1CS	0	B19Err: Scrub SBE (SF)
3	RW1CS	0	B18Err: Perfmon task completion - (Logical OR of PERFSTAT bits)
2	RW1CS	0	B17Err: JTAG/TAP error status (From JTAG register stack status bits)
1	RW1CS	0	B16Err: SMBus error status
0	RV	0h	Reserved

3.8.13.28 NRECINT - Non Recoverable Internal MCH 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 3.8.13.20. For debugging VPP errors in this register, e.g. if VPP_PEX_PORT2-3 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 3.8.12.29 to determine the failing port. The same can be repeated for the FB-DIMM Channels when VPP_FB-DIMM is set.

Device: 16
Function: 2
Offset: C8h

Bit	Attr	Default	Description
31:24	RV	0h	Reserved
23:21	ROS	000	VTIOU: VT Unaffiliated Port IOU Error Source. Bit 23 indicates IOU0 (Ports 1-4) Bit 22 indicates IOU1 (Ports 5-8) Bit 21 indicates IOU2 (ESI/Port 9)
20:13	ROS	0h	DM entry
12:11	RV	00	Reserved
10:8	ROS	000	Internal Block that detected the Failure 001: VPP_PEX_PORT1-4 010: VPP_PEX_PORT5-8 011: VPP_PEX_PORT9 100: COH 101:DM 110: IOG 111: *Reserved*
7	RV	0	Reserved
6:0	ROS	0h	COH Entry of Failed Location

3.8.13.29 RECINT - Recoverable Internal MCH Error Log Register

This register is not currently used as there are no correctable errors with in the internal data path of the MCH.

Device: 16
Function: 2
Offset: CCh

Bit	Attr	Default	Description
31:24	RV	0h	Reserved
23:21	ROS	000	VTIOU: VT Unaffiliated Port IOU Error Source. Bit 23 indicates IOU0 (Ports 1-4) Bit 22 indicates IOU1 (Ports 5-8) Bit 21 indicates IOU2 (ESI/Port 9)
20:13	ROS	0h	DM entry
12:11	RV	00	Reserved
10:8	ROS	000	Internal Block that detected the Failure 001: VPP_PEX_PORT1-4 010: VPP_PEX_PORT5-8 011: VPP_PEX_PORT9 100: COH 101:DM 110: IOG 111: *Reserved*
7	RV	0	Reserved
6:0	ROS	00h	COH Entry of Failed Location

3.8.13.30 EMASK_INT - Internal Error Mask Register

A '0' in any bit position enables the corresponding error.

Device: 16
Function: 2
Offset: D0h

Bit	Attr	Default	Description
31:26	RV	00h	Reserved
25	RWS	1	Software must not change the value of this bit
24	RWS	1	B25Msk: Illegal HiSMM/TSEG access (CE)
22	RWS	1	B23Msk: Vt Unaffiliated Port Error (IOG)
21	RWS	1	B22Msk: Victim ROM parity error (SF)
20	RWS	1	B21Msk: Illegal Way (SF)
19	RWS	1	B20Msk: Configuration Write Abort (CE).
18	RWS	1	B19Msk: Scrub SBE (SF)
14	RV	0	Reserved
13	RWS	1	B14Msk: Scrub DBE (SF)
12	RV	0	Reserved
11	RWS	1	B12Msk: Parity Protected Register Error
10	RWS	1	B11Msk: Victim RAM parity error (SF)
9	RWS	1	B10Msk: DM Parity Error (DM)

Device: 16 Function: 2 Offset: D0h			
Bit	**Attr**	**Default**	**Description**
8	RWS	1	B9Msk: Illegal Access Error (COH)
7	RWS	1	**B8Msk:** SF Coherency Error for BIL
6	RWS	1	**B7Msk:** Multiple ECC error in any of the ways during SF lookup
5	RWS	1	**B6Msk:** Single ECC error on SF lookup
4	RWS	1	**B5Msk:** Address Map Error
3	RWS	1	**B4Msk:** Virtual Pin Port Error
2	RWS	1	**B3Msk:** Coherency Violation Error for EWB
1	RWS	1	**B2Msk:** Multi-Tag Hit SF
0	RWS	1	**B1Msk:** DM Parity Error

3.8.13.31 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 in the ERR2_INT, ERR1_INT, ERR0_INT, and MCERR_INT for each of the corresponding bits.

Device: 16 Function: 2 Offset: DCh			
Bit	**Attr**	**Default**	**Description**
31:26	RV	00h	Reserved
25	RWS	1	Software must not change the value of this bit
24	RW	1	B25Err2Msk: Illegal HiSMM/TSEG access (CE)
22	RW	1	B23Err2Msk: Vt Unaffiliated Port Error (IOG)
21	RW	1	B22Err2Msk: Victim ROM parity error (SF)
20	RW	1	B21Err2Msk: Illegal Way (SF)
19	RW	1	B20Err2Msk: Configuration Write Abort (CE).
18	RW	1	B19Err2Msk: Scrub SBE (SF)
14	RV	0	Reserved
13	RW	1	B14Err2Msk: Scrub DBE (SF)
12	RV	0	Reserved
11	RW	1	B12Err2Msk: Parity Protected Register Error
10	RW	1	B11Err2Msk: Victim RAM parity error (SF)
9	RW	1	B10Err2Msk: DM Parity Error (DM)
8	RW	1	B9Err2Msk: Illegal Access Error (COH)
7	RW	1	**B8Err2Msk:** SF Coherency Error for BIL
6	RW	1	**B7Err2Msk:** Multiple ECC error in any of the ways during SF lookup
5	RW	1	**B6Err2Msk:** Single ECC error on SF lookup
4	RW	1	**B5Err2Msk:** Address Map Error
3	RW	1	**B4Err2Msk:** SMBus Virtual Pin Error
2	RW	1	**B3Err2Msk:** Coherency Violation Error for EWB
1	RW	1	**B2Err2Msk:** Multi-Tag Hit SF

Device: 16 Function: 2 Offset: DCh			
Bit	**Attr**	**Default**	**Description**
0	RW	1	B1Err2Msk: CE Parity Error

3.8.13.32 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 in the ERR2_INT, ERR1_INT, ERR0_INT, and MCERR_INT for each of the corresponding bits.

Device: 16 Function: 2 Offset: D8h			
Bit	**Attr**	**Default**	**Description**
31:26	RV	00h	Reserved
25	RWS	1	Software must not change the value of this bit
24	RW	1	B25Err1Msk: Illegal HiSMM/TSEG access (CE)
22	RW	1	B23Err1Msk: Vt Unaffiliated Port Error (IOG)
21	RW	1	B22Err1Msk: Victim ROM parity error (SF)
20	RW	1	B21Err1Msk: Illegal Way (SF)
19	RW	1	B20Err1Msk: Configuration Write Abort (CE).
18	RW	1	B19Err1Msk: Scrub SBE (SF)
14	RV	0	Reserved
13	RW	1	B14Err1Msk: Scrub DBE (SF)
12	RV	0	Reserved
11	RW	1	B12Err1Msk: Parity Protected Register Error
10	RW	1	B11Err1Msk: Victim RAM parity error (SF)
9	RW	1	B10Err1Msk: DM Parity Error (DM)
8	RW	1	B9Err1Msk: Illegal Access Error (COH)
7	RW	1	**B8Err1Msk:** SF Coherency Error for BIL
6	RW	1	**B7Err1Msk:** Multiple ECC error in any of the ways during SF lookup
5	RW	1	**B6Err1Msk:** Single ECC error on SF lookup
4	RW	1	**B5Err1Msk:** Address Map Error
3	RW	1	**B4Err1Msk:** SMBus Virtual Pin Error
2	RW	1	**B3Err1Msk:** Coherency Violation Error
1	RW	1	**B2Err1Msk:** Multi-Tag Hit SF
0	RW	1	B1Err1Msk: CE Parity Error

3.8.13.33 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 in the ERR2_INT, ERR1_INT, ERR0_INT, and MCERR_INT for each of the corresponding bits.

Device: 16
Function: 2
Offset: D4h

Bit	Attr	Default	Description
31:26	RV	00h	Reserved
25	RWS	1	Software must not change the value of this bit
24	RW	1	B25Err0Msk: Illegal HiSMM/TSEG access (CE)
22	RW	1	B23Err0Msk: Vt Unaffiliated Port Error (IOG)
21	RW	1	B22Err0Msk: Victim ROM parity error (SF)
20	RW	1	B21Err0Msk: Illegal Way (SF)
19	RW	1	B20Err0Msk: Configuration Write Abort (CE).
18	RW	1	B19Err0Msk: Scrub SBE (SF)
14	RV	0	Reserved
13	RW	1	B14Err0Msk: Scrub DBE (SF)
12	RV	0	Reserved
11	RW	1	B12Err0Msk: Parity Protected Register Error
10	RW	1	B11Err0Msk: Victim RAM parity error (SF)
9	RW	1	B10Err0Msk: DM Parity Error (DM)
8	RW	1	B9Err0Msk: Illegal Access Error (COH)
7	RW	1	**B8Err0Msk:** SF Coherency Error for BIL
6	RW	1	**B7Err0Msk:** Multiple ECC error in any of the ways during SF lookup
5	RW	1	**B6Err0Msk:** Single ECC error on SF lookup
4	RW	1	**B5Err0Msk:** Address Map Error
3	RW	1	**B4Err0Msk:** SMBus Virtual Pin Error
2	RW	1	**B3Err0Msk:** Coherency Violation Error for EWB
1	RW	1	**B2Err0Msk:** Multi-Tag Hit SF
0	RW	1	**B1Err0Msk:** CE Parity Error

3.8.13.34 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.

Device: 16
Function: 2
Offset: E0h

Bit	Attr	Default	Description
31:26	RV	00h	Reserved
25	RWS	1	Software must not change the value of this bit

Device: 16 Function: 2 Offset: E0h			
Bit	**Attr**	**Default**	**Description**
24	RW	1	B25McErrMsk: Illegal HiSMM/TSEG access (CE)
22	RW	1	B23McErrMsk: Vt Unaffiliated Port Error (IOG)
21	RW	1	B22McErrMsk: Victim ROM parity error (SF)
20	RW	1	B21McErrMsk: Illegal Way (SF)
19	RW	1	B20McErrMsk: Configuration Write Abort (CE).
18	RW	1	B19McErrMsk: Scrub SBE (SF)
14	RV	0	Reserved
13	RW	1	B14McErrMsk: Scrub DBE (SF)
12	RV	0	Reserved
11	RW	1	B12McErrMsk: Parity Protected Register Error
10	RW	1	B11McErrMsk: Victim RAM parity error (SF)
9	RW	1	B10McErrMsk: DM Parity Error (DM)
8	RW	1	B9McErrMsk: Illegal Access Error (COH)
7	RW	1	**B8McErrMsk:** SF Coherency Error for BIL
6	RW	1	**B7McErrMsk:** Multiple ECC error in any of the ways during SF lookup
5	RW	1	**B6McErrMsk:** Single ECC error on SF lookup
4	RW	1	**B5McErrMsk:** Address Map Error
3	RW	1	**B4McErrMsk:** SMBus Virtual Pin Error
2	RW	1	**B3McErrMsk:** Coherency Violation Error for EWB
1	RW	1	**B2McErrMsk:** Multi-Tag Hit SF
0	RW	1	**B1McErrMsk:** CEParity Error

3.8.13.35 FERR_FAT_THR: Throttling First Non-Fatal Error Register

FERR_FAT _THR latches the first fatal throttling error. All subsequent fatal throttling errors get logged in the NERR_FAT_THR.

Device: 16 Function: 2 Offset: F0h			
Bit	**Attr**	**Default**	**Description**
7:2	RV	0	Reserved
1	RW1CS	0	TH2Err: >Tmid Thermal event with intelligent throttling disabled (THRTSTS.GTMID)
0	RW1CS	0	**TH1Err: Catastrophic On-Die Thermal Event (THERMTRIP#)** (CTSTS.THRMTRIP)

3.8.13.36 FERR_NF_THR: Throttling First Non-Fatal Error Register

FERR_NF _THR latches the first non-fatal throttling error. All subsequent non-fatal throttling errors get logged in the NERR_NF_THR.

Device: 16 Function: 2 Offset: F1h			
Bit	**Attr**	**Default**	**Description**
7:5	RV	0	Reserved1
4	RW1CS	0	TH5Err: Deadman Timeout on Cooling Update (THRTSTS.DMTO)
3	RW1CS	0	TH4Err: TSMAX Updated (CTSTS.PKTRK)
2	RW1CS	0	**TH3Err: On-Die Throttling Event** (CTSTS.THRMALRT)
1:0	RV	0	Reserved

3.8.13.37 NERR_FAT_THR: Throttling Next Non-Fatal Error Register

Device: 16 Function: 2 Offset: F2h			
Bit	**Attr**	**Default**	**Description**
7:2	RV	0	Reserved
1	RW1CS	0	TH2Err: >Tmid Thermal event with intelligent throttling disabled (THRTSTS.GTMID)
0	RW1CS	0	TH1Err: Catastrophic On-Die Thermal Event (THERMTRIP#) (CTSTS.THRMTRIP)

3.8.13.38 NERR_NF_THR: Throttling Next Non-Fatal Error Register

Device: 16 Function: 2 Offset: F3h			
Bit	**Attr**	**Default**	**Description**
7:5	RV	0	Reserved1
4	RW1CS	0	TH5Err: Deadman Timeout on Cooling Update (THRTSTS.DMTO)
3	RW1CS	0	TH4Err: TSMAX Updated (CTSTS.PKTRK)
2	RW1CS	0	**TH3Err: On-Die Throttling Event** (CTSTS.THRMALRT)
1:0	RV	0	Reserved

3.8.13.39 EMASK_THR: Throttling Error Mask Register

A value of '0' in any field enables the error while a value of '1' suppresses it.

Device: 16 Function: 2 Offset: F6h			
Bit	**Attr**	**Default**	**Description**
15:5	RV	00h	Reserved
4	RWS	1	TH5Err: Deadman Timeout on Cooling Update

Device: 16 Function: 2 Offset: F6h			
Bit	**Attr**	**Default**	**Description**
3	RWS	1	TH4Err: TSMAX Updated
2	RWS	1	TH3Err: On-Die Throttling Event
1	RWS	1	TH2Err: >Tmid Thermal event with intelligent throttling disabled
0	RWS	1	TH1Err: Catastrophic On-Die Thermal Event (THERMTRIP#)

3.8.13.40 ERR0_THR: Throttling 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_THR, ERR1_THR, ERR0_THR, and MCERR_THR 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: F8h			
Bit	**Attr**	**Default**	**Description**
15:5	RW	00h	Reserved
4	RW	1	TH5Err: Deadman Timeout on Cooling Update
3	RW	1	TH4Err: TSMAX Updated
2	RW	1	TH3Err: On-Die Throttling Event
1	RW	1	TH2Err: >Tmid Thermal event with intelligent throttling disabled
0	RW	1	TH1Err: Catastrophic On-Die Thermal Event (THERMTRIP#)

3.8.13.41 ERR1_THR: Throttling 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_THR, ERR1_THR, ERR0_THR, and MCERR_THR 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: FAh			
Bit	**Attr**	**Default**	**Description**
15:5	RW	00h	Reserved
4	RW	1	TH5Err: Deadman Timeout on Cooling Update
3	RW	1	TH4Err: TSMAX Updated
2	RW	1	TH3Err: On-Die Throttling Event
1	RW	1	TH2Err: >Tmid Thermal event with intelligent throttling disabled
0	RW	1	TH1Err: Catastrophic On-Die Thermal Event (THERMTRIP#)

3.8.13.42 ERR2_THR: Throttling 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_THR, ERR1_THR, ERR0_THR, and MCERR_THR 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: FCh

Bit	Attr	Default	Description
15:5	RW	00h	Reserved
4	RW	1	TH5Err: Deadman Timeout on Cooling Update
3	RW	1	TH4Err: TSMAX Updated
2	RW	1	TH3Err: On-Die Throttling Event
1	RW	1	TH2Err: >Tmid Thermal event with intelligent throttling disabled
0	RW	1	TH1Err: Catastrophic On-Die Thermal Event (THERMTRIP#)

3.8.13.43 MCERR_THR: Throttling 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_THR, ERR1_THR, ERR0_THR, and MCERR_THR 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: FEh

Bit	Attr	Default	Description
15:5	RW	00h	Reserved
4	RW	1	TH5Err: Deadman Timeout on Cooling Update
3	RW	1	TH4Err: TSMAX Updated
2	RW	1	TH3Err: On-Die Throttling Event
1	RW	1	TH2Err: >Tmid Thermal event with intelligent throttling disabled
0	RW	1	TH1Err: Catastrophic On-Die Thermal Event (THERMTRIP#)

3.8.14 On-Die Thermal Sensor Throttling Registers

3.8.14.1 TSFSC: On-Die Thermal Sensor Fan-Speed-Control Register

Device: 16
Function: 4
Offset: F3h

Bit	Attr	Default	Description
7:0	RO	00h	**TSFSC: Thermal Sensor Fan Speed Control** This field contains the difference between the die temperature and the maximum permissible die temperature. This register has a resolution of 0.5°C. Hence, a register value in the range of 0 to 127 (decimal) refers to a temperature difference of 0 to 63.5°C, and a value in the range of 128 to 255 (decimal) refers to a temperature difference of -64°C to 0.5°C. Positive values indicate that the die temperature is lower than the maximum permissible die temperature. A negative difference beyond -64°C "floors" at -64°C. A positive difference beyond 63.5°C "ceilings" at 63.5°C. **Note:** Even though the default is shown as 00h, this is only guaranteed to be the value during the first clock period immediately following reset. At the first clock edge following reset, the value will be updated based on other register settings and the die temperature. For this reason, the register read after reset, will not return a particular fixed value

3.8.14.2 TSCTRL: On-Die Thermal Sensor Control Register

Device: 16
Function: 4
Offset: E8h

Bit	Attr	Default	Description
15:3	RV	00h	*Reserved*
12:10	RW	0	SLOPEADJ: Thermal Sensor Slope Adjust.
9	RW		**SWTHROTTLE: Software Throttle.** 0: Software throttling is disabled. TSDIS gates throttling. 1: Throttling is forced to all interfaces.
8	RW		**TSDIS: Thermal Sensor Throttling Disable.** 0: The thermal sensor determines closed-loop thermal throttling events when SWTHROTTLE = 0. 1: Thermal sensor throttling is disabled. SWTHROTTLE controls throttling.
7	RW1CS		**STSEVHI: Status Event High** 0: Thermal sensor event/interrupt is not generated by the sensor logic 1: Thermal sensor event/
6	RW1CS		**STSEVLO: Status Event Low** 0: Thermal sensor event/interrupt is not generated by the sensor logic. 1: Thermal sensor event/interrupts is set when the thermal sensor low threshold trip point is crossed.
5:1	RV		Reserved
0	RW		**TSEN: Thermal Sensor Enable** 0: Disable Thermal Sensor 1: Enable Thermal Sensor. When this bit is logic 1 a free-running counter is activated to produce a thermal measurement cycle. A measurement duration is 131,076 core cycles for the BandGap (BG) sensor and 32,768 core cycles for the CurrentMode (CML) sensor. The period between measurements is approximately 500us

3.8.14.3 TTSTHRRFSB: FSB Throttling Threshold Ratio Register

Device: 16
Function: 4
Offset: EAh

Bit	Attr	Default	Description
7:3	RV	00h	*Reserved*
2:0	RW	000	**TTFSBR: FSB Throttle Ratio** This register sets the throttling ratio for the processor buses.This setting is an approximate percentage of peak theoretical request bandwidth for this interface. Value Throttle Level Peak Bandwidth 000: 00.0% 100.0% Normal Unthrottled setting 001: 50.0% 50.0% 010: 75.0% 25.0% 011: 87.5% 12.5% 100: 93.8% 6.2% 101: 96.9% 3.1% 110: 98.5% 1.5% 111: 99.3% 0.7% Maximum Throttling enabled

3.8.14.4 TSTHRRPEX: PEX Throttling Threshold Ratio Register

Device: 16
Function: 4
Offset: EBh

Bit	Attr	Default	Description
7:3	RV	00h	*Reserved*
2:0	RW	000	**TTPEXR: PEX Throttle Ratio** This register sets the ratio for throttling the PEX's and ESI. This setting is an approximate percentage of peak theoretical bandwidth for this interface. Value / Throttle Level / Peak Bandwidth 000: 00.0% 100% Normal Unthrottled setting 001: 50.0% 50% 010: 75.0% 25% 011: 87.5% 12.5% 100: 93.8% 6.2% 101: 96.9% 3.1% 110: 98.5% 1.5% 111: 99.3% 0.7% Maximum Throttling enabled

3.8.14.5 TSTHRLO: On-Die Thermal Sensor Low Threshold Register

Device: 16
Function: 4
Offset: ECh

Bit	Attr	Default	Description
15:9	RV	0h	Reserved
8:0	RW	B4h	TSTHRLOLM: Thermal Sensor Threshold Low Limit The field is initialized by software to set the "low" threshold mark for the thermal sensor logic. (Tsr_lo). Resolution of this register is 0.5^{o}C. Two's-complement binary, range of -128^{o}C to 127.5^{o}C Default value is 90^{o}C. i.e B4h (180d)

3.8.14.6 TSTHRHI: On-Die Thermal Sensor High Threshold Register

Device: 16
Function: 4
Offset: EEh

Bit	Attr	Default	Description
15:9	RV	0h	Reserved
8:0	RW	0C8h	TSTHRHILM: Thermal Sensor Threshold High Limit The field is initialized by software to set the "high" threshold for the Thermal sensor logic. (Tsr_hi) Resolution of this register is 0.5^{o}C. Two's-complement binary, range of -128^{o}C to 127.5^{o}C Default value is 100^{o}C. i.e C8h (200d)

3.8.14.7 CTSTS: On-Die Throttling Status Register

Device: 16
Function: 4
Offset: F4h

Bit	Attr	Default	Description
7:3	RV	00h	Reserved1
2	RW1CS	0	PKTRK: TSMAX Was Updated. TSMAX was updated while CTCTRL.NOMAX was cleared.
1	RW1CS	0	THRMALRT: Throttling Event
0	RW1CS	0	**THRMTRIP: Catastrophic Thermal Event**

3.8.14.8 CTCOOL: Die Cooling Level Register

Device: 16
Function: 4
Offset: F6h

Bit	Attr	Default	Description
7:1	RV	00h	*Reserved*
0	RW	0h	MAXCOOL: Maximum Cooling Die is immersed in maximum cooling, so disable on-die throttling and allow TSMAX tracking of TSVAL peaks.

3.8.14.9 CTCTRL: On-Die Throttling Control Register

Device: 16
Function: 4
Offset: F7h

Bit	Attr	Default	Description
7:4	RV	00h	*Reserved*
3	RW	0h	NOBMC: No BMC Mode '1: TSTHRHI and TSTHRLO drive THERMALERT#. TSTHRNOBMC drives throttling. '0: TSTHRHI and TSTHRLO drive THERMALERT#. TSTHRHI and TSTHRLO also drive throttling, TSTHRNOBMC is un-used.

Device: 16 Function: 4 Offset: F7h			
Bit	**Attr**	**Default**	**Description**
2	RW	0h	CTALERT: Connect On-Die Thermal Sensor to THERMALERT# **When NOBMC is cleared:** When this bit is set, on-die thermal throttling occurring during a THERMALERT window will appear on the THERMALERT# pin during the next THERMALERT window. **When NOBMC is set:** When this bit is set, excursions of TSTHRHI and TSTHRLO occurring during a THERMALERT window will appear on the THERMALERT# pin during the next THERMALERT window.
1	RW	0h	HINTEN: On-Die Throttle Hint Enable When this bit is set, on-die throttling hints are enabled.
0	RW	0	NOMAX: TSMAX Tracking Mode. Temporarily toggle NOMAX to '1 then back to '0

3.9 Memory Control Registers

3.9.1 MC - Memory Control Settings

Miscellaneous controls not implemented in other registers.

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 operations
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	**FB-DIMMALIGN: FB-DIMM Frame Alignment Mode** '1' = Capped alignment. "Pad" tRRL by one frame after transition from "Ready" to "Active". 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: Intel® 5400 chipset will use a 38-bit system address with up to 255.75GB of memory. The Memory Controller will internally use a 38-bit decoder. 1: MCH will use 36-bit system address with up to 64GB of memory. The Memory Controller will internally use a 36-bit decoder. This capability field can be used by BIOS/software to determine whether the system is using a 38-bit or 36-bit address space and then use it to initialize the "**Enable40bADDR" (bit 22)** field in this register for the appropriate application

Device: 16 Function: 1 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
22	RW	0	Enable40bADDR: Enable 40-bit ADDRess 0: Intel® 5400 chipset 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: Intel® 5400 chipset will use a 38-bit system address with up to 128GB of DDR DIMMs. The MC will be set in the 38-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 Intel® 5400 chipset MCH to BIOS. BIOS sets this bit to 1 after initialization of the DRAM memory array is complete. This bit has no effect on Intel® 5400 chipset 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 FB-DIMM link comes up, trains, goes to L0, sends NOPs, does alerts, syncs, fast resets, AMB configurations, etc., but does not perform: a) Memory reads b) Memory writes c) Refreshes Not preserved by SAVCFG bit in the SYRE register.
19:17	RV	0h	Reserved2
16	RW	0	**MIRROR: Mirror mode enable** '1' = mirroring enabled '0' = mirroring disabled. MCH does not support this feature
15:9	RV	0h	Reserved
8	RW	0	SCRBALGO: Scrub 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. This bit must not be set when MIRROR is set.
5	RW	0	**ERRDETEN: Error Detection Enable** '1' = Northbound CRC/ECC checking enabled. '0' = Northbound CRC/ECC checking disabled FB-DIMM "Alert" detection is disabled, status packets are ignored, northbound error logging and data poisoning are disabled when Northbound CRC/ECC checking is disabled.
4	RWC	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 Intel® 5400 chipset MCH will stop the scrub cycle immediately.
3:0	RV	0h	Reserved

Table 3-38. Global Activation Throttling as a Function of Global Activation Throttling Limit 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	65536	16
2	131072	32
16	1048576	256
32	2097152	512
64	4194304	1024
96	6291456	1536
100	6553600	1600
128	8388608	2048
150	9830400	2400
168	11010048 (100% BW)	2688 (100% BW)

3.9.2 THRTCTRL: Memory Thermal Throttling Control Register

Device: 16
Function: 3
Offset: 67h

Bit	Attr	Default	Description
7:2	RV	0h	Reserved
1	RW	0	**THRMODE: Thermal Throttle Mode** 0: THRTSTS.THRMTHRT register is initialized by the MCH such that they vary in range between THRTMID and THRTHI above Tmid (staircase) 1: THRTSTS.THRMTHRT register field is "slammed" to THRTHI above Tmid.
0	RW	0	**THRMHUNT: Intelligent Thermal Throttle Enable** 0: THRTSTS.THRMTHRT register is not enabled 1: THRTSTS.THRMTHRT register is enabled for the temperature to have any influence on the throttle parameters. If THRMHUNT=0 only the GBLTHRT bit from the Global Throttle Window can change the THRMTHRT register field.

3.9.3 THRTCTRLA: Memory Throttling Control Register A

Device: 16
Function: 3
Offset: 40h

Bit	Attr	Default	Description
31:18	RV	0h	Reserved
17	RV	0	Reserved
16:10	RV	0h	Reserved
9	RW	0h	HINTEN: Memory Throttle Hint Enable When this bit is set, memory throttling hints are enabled.
8	RW	0h	MTALERT: Connect Memory Throttle to THERMALERT# When this bit is set, memory throttling occurring during a global throttle window will appear on the THERMALERT# pin during the next global throttle window.

Device: 16 Function: 3 Offset: 40h			
Bit	**Attr**	**Default**	**Description**
7:4	RW	0h	DMPER: Deadman Timer Period This register field is used to define the period of the dead-man timer. Each increment is eight global activation timer periods.
3	RW	0	MTEN: Memory Throttle Enable This register field is used to define the relative open-loop thermal weight of a CAS relative to a RAS. I.e. presuming a RAS always has a weight of "1", the CAS has a weight of "1/N". Legal values span the inclusive range of "3" to "7". Operation outside of these bounds is not specified.
2:0	RV	000	Reserved

3.9.4 THRTSTS: Memory Throttling Status Register

This register records the global activation throttle status and internal thermal throttling value for memory.

Device: 16 Function: 3 Offset: 68h			
Bit	**Attr**	**Default**	**Description**
15:12	RV	0h	Reserved
11	RW1CS	0	THRMALRT: Memory Throttling Event
10	RWC1S	0	DMTO: Deadman Timeout on Cooling Update An interval of 8*THRTCTRLA.DMPER was exceeded since the last write to MTCOOL.
9	RWC1S	0	GTMID: >Tmid Thermal event with intelligent throttling disabled An FB-DIMM channel received a >Tmid status indication from one of its AMB's while THRTCTRL.THRMHUNT = 0.
8	RO	0h	GBLTHRT: Open Loop Throttling Level This register field is used to indicate which threshold is being used in open-loop memory throttling: '0: THRTLOW '1: THRTMID If the number of activates in the global window exceeds the number indicated by THRTMID[MTCOOL], then this bit is set to select THRTMID as the throttling threshold. THRTMID will remain active until consecutive global throttling windows have elapsed without any DIMM exceeding THRTMID[MTCOOL].
7:0	RO	0h	**THRMTHRT: Thermal Throttle Value** This field holds the current activation throttling value based on the throttling algorithm 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.

3.9.5 THRTLOWA: Memory Throttling Low Register A

This register specifies a base throttling level that is applied when all of the DIMM temperatures are in the low range (below FB-DIMM.Tlow) and THRTCTRL.THRMHUNT is set, or, the THRTSTS.GBLTHRT bit is not set by the Global Throttling Window logic.

If Software sets this value greater than the maximum allowable value, the chipset will internally use the maximum allowable value.

The resolution of each field is 4 activations per value increment.

Device: 16
Function: 3
Offset: E0h

Bit	Attr	Default	Description
31:24	RW	0h	THRTLOWLM3: Thermal Throttle Low Limit for Cooling Level 3
23:16	RW	0h	THRTLOWLM2: Thermal Throttle Low Limit for Cooling Level 2
15:8	RW	0h	THRTLOWLM1: Thermal Throttle Low Limit for Cooling Level 1
7:0	RW	0h	THRTLOWLM0: Thermal Throttle Low Limit for Cooling Level 0

3.9.6 THRTLOWB: Memory Throttling Low Register B

Device: 16
Function: 3
Offset: E4h

Bit	Attr	Default	Description
31:24	RW	0h	THRTLOWLM7: Thermal Throttle Low Limit for Cooling Level 7
23:16	RW	0h	THRTLOWLM6: Thermal Throttle Low Limit for Cooling Level 6
15:8	RW	0h	THRTLOWLM5: Thermal Throttle Low Limit for Cooling Level 5
7:0	RW	0h	THRTLOWLM4: Thermal Throttle Low Limit for Cooling Level 4

3.9.7 THRTLOWC: Memory Throttling Low Register C

Device: 16
Function: 3
Offset: E8h

Bit	Attr	Default	Description
31:24	RW	0h	THRTLOWLM11: Thermal Throttle Low Limit for Cooling Level 11
23:16	RW	0h	THRTLOWLM10: Thermal Throttle Low Limit for Cooling Level 10
15:8	RW	0h	THRTLOWLM9: Thermal Throttle Low Limit for Cooling Level 9
7:0	RW	0h	THRTLOWLM8: Thermal Throttle Low Limit for Cooling Level 8

3.9.8 THRTLOWD: Memory Throttling Low Register D

Device: 16
Function: 3
Offset: ECh

Bit	Attr	Default	Description
31:24	RW	0h	THRTLOWLM15: Thermal Throttle Low Limit for Cooling Level 15

Device: 16 Function: 3 Offset: ECh			
Bit	**Attr**	**Default**	**Description**
23:16	RW	0h	THRTLOWLM14: Thermal Throttle Low Limit for Cooling Level 14
15:8	RW	0h	THRTLOWLM13: Thermal Throttle Low Limit for Cooling Level 13
7:0	RW	0h	THRTLOWLM12: Thermal Throttle Low Limit for Cooling Level 12

3.9.9 THRTMIDA: Memory Throttling Mid Register A

This register specifies a medium throttling level that is applied when the temperature is in the middle range (above or equal to FB-DIMM.Tlow but below FB-DIMM.Tmid) and THRTCTRL.THRMHUNT is set, or, the THRTSTS.GBLTHRT* bit is set by the Global Throttling Window logic.

If Software sets this value greater than the maximum allowable value, the chipset will internally use the maximum allowable value.

The resolution of each field is 4 activations per value increment.

Device: 16 Function: 3 Offset: F0h			
Bit	**Attr**	**Default**	**Description**
31:24	RW	0h	THRTMIDLM3: Thermal Throttle Middle Limit for Cooling Level 3
23:16	RW	0h	THRTMIDLM2: Thermal Throttle Middle Limit for Cooling Level 2
15:8	RW	0h	THRTMIDLM1: Thermal Throttle Middle Limit for Cooling Level 1
7:0	RW	0h	THRTMIDLM0: Thermal Throttle Middle Limit for Cooling Level 0

3.9.10 THRTMIDB: Memory Throttling Mid Register B

Device: 16 Function: 3 Offset: F4h			
Bit	**Attr**	**Default**	**Description**
31:24	RW	0h	THRTMIDLM7: Thermal Throttle Middle Limit for Cooling Level 7
23:16	RW	0h	THRTMIDLM6: Thermal Throttle Middle Limit for Cooling Level 6
15:8	RW	0h	THRTMIDLM5: Thermal Throttle Middle Limit for Cooling Level 5
7:0	RW	0h	THRTMIDLM4: Thermal Throttle Middle Limit for Cooling Level 4

3.9.11 THRTMIDC: Memory Throttling Mid Register C

Device: 16 Function: 3 Offset: F8h			
Bit	**Attr**	**Default**	**Description**
31:24	RW	0h	THRTMIDLM11: Thermal Throttle Middle Limit for Cooling Level 11
23:16	RW	0h	THRTMIDLM10: Thermal Throttle Middle Limit for Cooling Level 10
15:8	RW	0h	THRTMIDLM9: Thermal Throttle Middle Limit for Cooling Level 9

Device: 16 Function: 3 Offset: F8h			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0h	THRTMIDLM8: Thermal Throttle Middle Limit for Cooling Level 8

3.9.12 THRTMIDD: Memory Throttling Mid Register D

Device: 16 Function: 3 Offset: FCh			
Bit	**Attr**	**Default**	**Description**
31:24	RW	0h	THRTMIDLM15: Thermal Throttle Middle Limit for Cooling Level 15
23:16	RW	0h	THRTMIDLM14: Thermal Throttle Middle Limit for Cooling Level 14
15:8	RW	0h	THRTMIDLM13: Thermal Throttle Middle Limit for Cooling Level 13
7:0	RW	0h	THRTMIDLM12: Thermal Throttle Middle Limit for Cooling Level 12

3.9.13 THRTHI: Memory Thermal Throttling High Register

This register specifies the highest level of throttling (i.e. minimum number of activations), independent of cooling level. When THRTCTRL.THRMODE=1, this level is applied whenever the temperature is above or equal to FB-DIMM.Tmid. When THRTCTRL.THRMODE=0, this level is the ceiling of the closed loop throttling hunting algorithm. The temperature being above or equal to FB-DIMM.Tmid forces the application of THRTHI, superseding any other throttling consideration.

This throttling will be enabled if THRTCTRL.THRMHUNT is set.

If Software sets this value greater than the maximum allowable value, the chipset will cap the THRTHILM field to the maximum allowable value.

The resolution of each field is 4 activations per value increment.

Device: 16 Function: 3 Offset: 66h			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0h	THRTHILM: Thermal Throttle High Limit

3.9.14 GBLACT: CLTT Global Activation Throttle Register

This register specifies the global activation throttle limit in closed-loop-therm-throttling mode (CLTT) (THRTCTRL.THRMHUNT=1), independent of cooling level. When THRTCTRL.THRMHUNT=1, this level is applied whenever the temperature is less than or equal to FB-DIMM.Tlow. When THRTCTRL.THRMHUNT=0, this level does not establish the global activation limit; rather, THRTMID establishes the global activation limit.

If Software sets this value greater than the maximum allowable value, the chipset will cap the THRTHILM field to the maximum allowable value. A value of zero allows unlimited activations.

The resolution is 65,536 lines per value increment.

Device: 16
Function: 3
Offset: 64h

Bit	Attr	Default	Description
7:0	RW	0h	GBLACTLM: Global Activation Throttle Limit

3.9.15 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	0h	**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. A default value of 54d (36h) is chosen to account for the pending read requests in the MCH pipeline (Micro-Architecture and timing dependent) that can potentially be transferred to the deferred queue
21:16		32h	**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 50d (32h) 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	0	Reserved
14	RW	0	**SCHDIMM:** Single Channel DIMM Operation 0: The MC assumes that the Intel® 5400 chipset MCH is operating normally, i.e. MC is not operating with only one FB-DIMM channel as in single channel mode. 1: In this mode, the Intel® 5400 chipset 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.

Device: 16 Function: 1 Offset: 58h			
Bit	**Attr**	**Default**	**Description**
13:8	RW	1Ah	**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. A default value of 26d (1Ah) is chosen to account for the pending write requests in the MCH pipeline (Micro-Architecture and timing dependent) that can potentially be transferred to the deferred queue
7:6	RV	0	*Reserved*.
5:0	RW	16h	**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 22d (16h) is chosen to set the low threshold for the write request queue

3.9.16 Memory Gearing Registers

3.9.16.1 DDRFRQ - DDR Frequency Ratio

This register specifies the CORE:DDR frequency ratio. This register must be written once after a hard reset.

Device: 16 Function: 1 Offset: 56h			
Bit	**Attr**	**Default**	**Description**
7:4	RV	0	Reserved
3	RV	0	**Reserved**
2:0	RWS	100	**DDRFRQ: CORE:DDR Frequency Ratio** Unlike the previous generation of memory controller hub which required a hard reset to "latch in", an update to this frequency ratio will take effect immediately (within core cycles). The FB-DIMMRST.SOFTCORERESET# and FB-DIMMRST.CORERESET# bits must be asserted when this register is written. '000' = **1:1**. BUSCLK=266MHz, DDR=533MHz. '001' = **1:1**. BUSCLK=333MHz, DDR=667MHz. '010' = **4:5**. BUSCLK=266MHz, DDR=667MHz. '011' = **5:4**. BUSCLK=333MHz, DDR=533MHz. '100' = **1:1**. BUSCLK=400MHz, DDR=400MHz (DDR2-800) '101' = **Reserved** '110' = **Reserved** '111' = **5:4**. BUSCLK=400MHz, DDR=320MHz (DDR2-640). This field will only set the relationship between the CORE-domain and FB-DIMM-domain (MEM) clocks. This field will not set the frequency of the FB-DIMM SCiD link... that is entirely determined by the frequency of the FB-DIMMCLK reference clocks. To achieve successful FB-DIMM channel initialization, the frequency of the FB-DIMMCLK reference clock must match the frequency of the FB-DIMM-domain clock. E.g. if the BUSCLK=333MHz and this field specifies a ratio of 1:1, then FB-DIMM channel initialization will succeed with an FB-DIMMCLK frequency of 333 MHz.

3.9.16.2 DRAMPDCTL: DRAM Power Down Control

The feature will allow the MCH to drop CKE(DRAM powerdown entry) to save power during idle conditions.

Device: 16 Function: 1 Offset: 64h			
Bit	**Attr**	**Default**	**Description**
31:16	RV	0	Reserved
15:4	RW	000h	**IdleCountLimit:** Idle cycle Count Limit/threshold. Powerdown entry will be requested if no traffic (excluding scrub & refresh) has been sent before the limit/threshold expires. The register implements the upper 12-bits of a 16-bit counter limit, whose lower 4 bits are always zero. Because of this the counter limit is restricted to a mod-16 number(0,16,32,48,etc). If these 12 MSbs are all zero then the limit is zero, and the powerdown feature is disabled.
3:1	RW	100	**PDExitCountLimit:** Power Down Exit Count Limit. Once a "CKE HIGH" has been issued the state machine waits this many cycles before allowing any activate commands. This gives us the ability to program our own "powerdown-to-ODT" latency. (Note: tXP is enforced regardless.)
:0	RW	0h	**PerRankCKEDEF:** Per Rank CKE Defeature This is a per rank cke defeature bit which allows us to come out of powerdown on a per rank basis to issue refresh when asserted. Note: Any powerdown exits related to scrub or demand traffic will be issued on a per channel basis.

3.9.16.3 FB-DIMMTOHOSTGRCFG0: FB-DIMM to Host Gear Ratio Configuration

This register consists of 8 nibbles of mux select data for the proper selection of gearing behavior on the FB-DIMM. This is the first of two registers to control the behavior for the FB-DIMM to host (north bound) data flow.

Device: 16 Function: 1 Offset: 160h			
Bit	**Attr**	**Default**	**Description**
31:0	RWS	00000001h	FB-DIMMHSTGRMUX: FB-DIMM to Host Clock Gearing mux selector. Eight nibbles of mux select for memory/DDR2 to FSB/core geared clock boundary crossing phase enables. Refer to Table 3-39 for the programming details.

Table 3-39. FB-DIMM to Host Gear Ratio Mux

Memory Frequency	Gear Ratio[1]	Option	Value
400:400 333:333 267:267	1:1	only	00000001h
400:320 333:267	5:4	aggressive	00023023h
267:333	4:5	aggressive	00002423h

Notes:

1. For 4:5 gear ratio, Software should use either conservative or aggressive mode for all the respective memory gearing registers. (no mix and match)

3.9.16.4 FB-DIMMTOHOSTGRCFG1: FB-DIMM to Host Gear Ratio Configuration 1

This register consists of eight nibbles of mux select data for the proper selection of gearing behavior on the FB-DIMM for the 1:1 and 4:5 modes.This is the second register for FB-DIMM to Host gearing control.

Device: 16 Function: 1 Offset: 164h			
Bit	**Attr**	**Default**	**Description**
31:0	RWS	00000000h	FB-DIMMHSTGRMUX: FB-DIMM to Host Clock Gearing mux selector. Eight nibbles of mux select for memory/DDR2 to FSB/core geared clock boundary crossing phase enables. Refer to Table for the programming details.

Table 3-40. FB-DIMM to Host Gear Ratio Mux

FSB:Memory Frequency	Gear Ratio[1]	Option	Value
400:400 333:333 267:267	1:1	only	00000000h
400:320 333:267	5:4	aggressive	00000000h
267:333	4:5	aggressive	00000030h

Notes:

1. For the 4:5 gear ratio, Software should use either conservative or aggressive mode for all the respective memory gearing registers. (no mix and match)

3.9.16.5 HOSTTOFB-DIMMGRCFG: Host to FB-DIMM Gear Ratio Configuration

This register consists of eight nibbles of mux select data for the proper selection of gearing behavior on the Host to FB-DIMM path (south bound).

Device: 16 Function: 1 Offset: 168h			
Bit	**Attr**	**Default**	**Description**
31:0	RWS	00000001h	HSTFB-DIMMGRMUX: Host to FB-DIMM Clock Gearing mux selector. Eight nibbles of mux select for FSB/core to memory/DDR2 geared clock boundary crossing phase enables. Refer to Table for the programming details.

Table 3-41. Host to FB-DIMM Gear Ratio Mux Select (Sheet 1 of 2)

FSB:Memory Frequency	Gear Ratio[1]	Option	Value
400:400 333:333 267:267	1:1	only	00000001h

Table 3-41. Host to FB-DIMM Gear Ratio Mux Select (Sheet 2 of 2)

FSB:Memory Frequency	Gear Ratio[1]	Option	Value
400:320 333:267	5:4	aggressive	00002433h
267:333	4:5	aggressive	00020323h

Notes:

1. For the 4:5 gear ratio, Software should use either conservative or aggressive mode for all the respective memory gearing registers. (no mix and match)

3.9.16.6 GRFB-DIMMVLDCFG: FB-DIMM Valid Configuration

This register provides valid signals to assert data in the FB-DIMM side for various gearing ratios. It primarily affects the southbound data path for 4:5 gearing and determines when a NOP packet is to be inserted into the FB-DIMM.

Device: 16
Function: 1
Offset: 16Ch

Bit	Attr	Default	Description
7:0	RWS	0h	FB-DIMMVLDMUX: FB-DIMM Data Valid Mux selector. Determines which valid host cycle to insert NOP. Refer to Table for the programming details. This primarily affects the 4:5 gearing ratio.

Table 3-42. FB-DIMM Host Data Cycle Valid Mux Select

FSB:Memory Frequency	Gear Ratio[1]	Option	Value
400:400 333:333 267:267	1:1	only	00h
400:320 333:267	5:4	aggressive	00h
267:333	4:5	aggressive	08h

Notes:

1. For the 4:5 gear ratio, Software should use either conservative or aggressive mode for all the respective memory gearing registers. (no mix and match)

3.9.16.7 GRHOSTFULLCFG: Host Full Flow Control Configuration

This register configures flow control when the host is full. It primarily effects the Southbound data path and determines when the flow control signal to the core is asserted.

Device: 16
Function: 1
Offset: 16Dh

Bit	Attr	Default	Description
7:0	RWS	0h	FCMUX: Flow Control Mux Selector Configures Flow control on the host according to Table 3-43. This primarily affect the 5:4 gearing ratio.

Table 3-43. FB-DIMM to Host Flow Control Mux Select

FSB:Memory Frequency	Gear Ratio[1]	Option	Value
400:400 333:333 267:267	1:1	only	00h
400:320 333:267	5:4	aggressive	08h
267:333	4:5	aggressive	01h

Notes:

1. For the 4:5 gear ratio, Software should use either conservative or aggressive mode for all the respective memory gearing registers. (no mix and match)

3.9.16.8 GRBUBBLECFG: FB-DIMM Host Bubble Configuration

This register provides valid signals to assert data in the FB-DIMM side for various gearing ratios. This primarily affects the Northbound data path for the 5:4 configuration and determines when a bubble is inserted when gearing up.

Device: 16
Function: 1
Offset: 16Eh

Bit	Attr	Default	Description
7:0	RWS	0h	FB-DIMMBBLMUX: FB-DIMM Data Bubble Mux selector. Configures bubbles in the host according to Table . This primarily affect the 5:4 gearing ratio.

Table 3-44. FB-DIMM Bubble Mux Select

FSB:Memory Frequency	Gear Ratio	Option	Value
400:400 333:333 267:267	1:1	only	00h
400:320 333:267	5:4	aggressive	02h
267:333	4:5	aggressive	10h

3.9.16.9 GRFB-DIMMTOHOSTDBLCFG: FB-DIMM To Host Double Configuration

This register provides valid signals to assert data in the FB-DIMM side for various gearing ratios. This primarily affects the Northbound data path of the 4:5 config and determines when both the lanes in core contain valid FB-DIMM data.

Device: 16
Function: 1
Offset: 16Fh

Bit	Attr	Default	Description
7:0	RWS	0h	FB-DIMMHSTDBLMUX: FB-DIMM to Host Double Mux Selector Configures when both data lines are valid according to Table . This primarily affect the 4:5 gearing ratio.

Table 3-45. FB-DIMM to Host Double Config Mux Select

FSB:Memory Frequency	Gear Ratio[1]	Option	Value
400:400 333:333 267:267	1:1	only	00h
400:320 333:267	5:4	aggressive	00h
267:333	4:5	aggressive	02h

Notes:

1. For the 4:5 gear ratio, Software should use either conservative or aggressive mode for all the respective memory gearing registers. (no mix and match)

3.9.16.10 Summary of Memory Gearing Register Operating Modes

- FB-DIMMTOHOSTGRCFG1, GRFB-DIMMVLDCFG, and GRFB-DIMMTOHOSTDBLCFG are used only in 4:5 mode.
- GRBUBBLECFG is only used in 5:4 mode.
- GRHOSTFULLCFG is used in both 4:5 and 5:4 modes.
- FB-DIMMTOHOSTGRCFG0 and HOSTTOFB-DIMMGRCFG are used in 4:5, 5:4, AND 1:1 modes.

Table 3-46. "A" Multiplier for DRTA and DRTB Timings

Core: FB-DIMM frame gear ratio	A
1:1, 5:4	1
4:5	0.8

3.9.17 DRTA - DRAM Timing Register A

This register defines timing parameters for all DDR2 SDRAMs in the memory subsystem. The parameters for these devices are obtained by serial presence detect. This register must be set to provide timings that satisfy the specifications of all DRAMs detected. E.g., if DRAMs present have different TRCs, the maximum should be used to program this register. Consult the JEDEC DDR2 DRAM specifications for the technology of the devices in use.

Device: 16 Function: 1 Offset: 48h			
Bit	**Attr**	**Default**	**Description**
31	RV	0	Reserved

Device: 16
Function: 1
Offset: 48h

Bit	Attr	Default	Description
30:26	RW	0h	**TFAW: Electrical Throttling Window** This parameter is the smallest window over which four activates can be issued to a given rank: no more than four activates can be issued within any given (sliding) t_{FAW} window. This parameter prevents DRAM power-supply droop violations. This parameter is defined in core cycles. This parameter is set to greater than or equal to the largest t_{FAW} of any DIMM on the memory sub-system. This parameter is defined as follows: (A * ((t_{FAW} / $t_{FB-DIMM}$) rounded up to the nearest integer) rounded up to the nearest integer), where $t_{FB-DIMM}$ is the cycle time of an FB-DIMM frame, and t_{CORE} is the gear-ratio multiplier defined in Table 3-46. For further details, refer to "An all 0 frame fits this case of good CRC and ECC. This is just as unlikely as any other random frame contents when interpreting EI." on page 327.
25:20	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: (A * ((CL – 1) + BL/2 +((t_{RCD} + t_{WR} + t_{RP}) / $t_{FB-DIMM}$) rounded up to the nearest integer)) rounded up to the nearest integer), where t_{RCD} is the DDR ras-to-cas delay (maximum 5), 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, $t_{FB-DIMM}$ is the cycle time of an FB-DIMM frame, and A is the gear-ratio multiplier defined in Table 3-46.
19:14	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: (A * ((t_{RC} / $t_{FB-DIMM}$) rounded up to the nearest integer) rounded up to the nearest integer) where $t_{FB-DIMM}$ is the cycle time of an FB-DIMM frame and A is the gear-ratio multiplier defined in Table 3-46.
13:6	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} / $t_{FB-DIMM}$) rounded up to the nearest integer) rounded up to the nearest integer) where $t_{FB-DIMM}$ is the cycle time of an FB-DIMM frame and A is the gear-ratio multiplier defined in Table 3-46.
5:2	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: (A * ((t_{RRD} / $t_{FB-DIMM}$) rounded up to the nearest integer) rounded up to the nearest integer) where $t_{FB-DIMM}$ is the cycle time of an FB-DIMM frame and A is the gear-ratio multiplier defined in Table 3-46.
1:0	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. This parameter is defined in multiples of 3.9 microseconds. This parameter is set to less than or equal to the smallest TREF of any DIMM on the memory sub-system. A value of zero disables refresh and clears the refresh counter. The maximum value is "10", for 7.8 microseconds.

3.9.18 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 TR2Ws, the maximum should be used to program this register.

Device: 16
Function: 1
Offset: 4Ch

Bit	Attr	Default	Description
31:19	RV	000h	Reserved
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 ((A * 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 FB-DIMM frames, and A is the gear-ratio multiplier defined in Table 3-46.
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 ((A * 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 FB-DIMM frames, and A is the gear-ratio multiplier defined in Table 3-46.
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 (A * (CL - 1 + BL/2 + max((((t_{WTR} / $t_{FB-DIMM}$) 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, $t_{FB-DIMM}$ is the cycle time of an FB-DIMM frame, t_{FRR} is the turnaround time from write to read on the DIMM in FB-DIMM frames, and A is the gear-ratio multiplier defined in Table 3-46.
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 ((A * 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 FB-DIMM frames, BL is the burst length, and A is the gear-ratio multiplier defined in Table 3-46.
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 ((A * BL/2) rounded up to the nearest integer). BL is the burst length and A is the gear-ratio multiplier defined in Table 3-46.

3.9.19 ERRPER - Error Period Prescaler

Non-zero CERRCNT counts are decremented when the DIMMPAIRCNT reaches their programmed values. There are 8 of these DIMM pair counters, one per each DIMM pair; 4 per branch. The output of the Error Period register is what increments the DIMM pair counters. The error period counter is cleared on reset or when it reaches this threshold. The error period counter increments every 16 cycles. The output of the programmable Error Period counter increments the DIMM pair counters. Table 3-47 below indicates the timing characteristics of this register:

Table 3-47. Timing Characteristics of ERRPER

Core Frequency	Per Increment	Maximum Period (increment period x ((2^32)-1))Maximum Period
400 MHz	40 ns	171.8 seconds
333 MHz	48 ns	206.2 seconds
266 MHz	60 ns	257.7 seconds

Device: 16
Function: 1
Offset: 50h

Bit	Attr	Default	Description
31:0	RW	0h	THRESH: Global ERRPER increment threshold. A value of 0 prevents incrementing ERRPER as well as DIMMPAIRCNT, and thus decrementing CERRCNT. When the value in this register is reached, the DIMM pair counters are decremented by 1.

3.9.20 DIMMPAIRCNT[1:0][3:0]: DIMM Pair Count

Non-zero CERRCNT counts are decremented when their respective DIMM pair count register reaches its programmed values. There are 8 of these DIMM pair counters, one per each DIMM pair; 4 per branch. The DIMM Pair Counter is cleared on reset or when it reaches its threshold. The output of the Error Period register is what increments the DIMM pair counters. Table 3-48 indicates the timing characteristics of this register.

Table 3-48. Timing Characteristics of DIMMPAIRCNT

Core Frequency	Per Increment	Maximum Period (increment period x ((2^16)-1))
400 Mhz	40 ns - 171.8 seconds	2.62 microseconds - 130.3 days
333 MHz	48 ns - 206.2 seconds	3.15 microseconds - 156.4 days
266 MHz	60 ns - 257.7 seconds	3.93 microseconds - 195.5 days

Device: 22,21 Function: 0 Offset: B6h,B4h,B2h,B0h			
Bit	**Attr**	**Default**	**Description**
15:0	RW	0h	THRESH: DIMM Pair CERRCNT decrement threshold. A value of 0 prevents incrementing DIMMPAIRCNT, and thus decrementing CERRCNT. When the value in this register is reached, the leaky bucket counters are decremented by 1.

3.9.21 Memory Map Registers

3.9.21.1 TOLM - Top Of Low Memory

This register defines the low MMIO gap below 4 GB. See Section 3.9.21.2.

Whereas the MIR.LIMITs 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 4GB. Addresses **equal to** or greater than the TOLM, and less than 4G, 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 256 MB boundary. This field must be set to at least 1h, for a minimum of 256 MB of DRAM. There is also a minimum of 256 MB between TOLM and 4 GB (for low MMIO, MMCFG, chipset, interrupt/SMM and firmware) since TOLM is on a 256 MB boundary. 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

3.9.21.2 MIR[1:0] - Memory Interleave Range

These registers define each memory branch's interleave participation in processor-physical (A) space.

Each register defines a range. If the processor-physical address falls in the range defined by an MIR, the "way" fields in that MIR defines 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. In mirror mode, the WAY0 and WAY1 fields should be set to 11b. Matching addresses participate in the corresponding ways.

Compensation for a non-4GB MMIO gap size is performed by adjusting the limit of each range upward if it is above TOLM as shown in Table 3-49.

MIR updates can only occur in the RESET, READY, FAULT, DISABLED states.

Table 3-49. Interleaving of an address is governed by MIR[i]

Limit with respect to TOLM	Match MIR[i]
if MIR[i].LIMIT[11:0] <= TOLM	then MIR[i].LIMIT[11:0] > A[38:28] >= 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[38:28] >= MIR[i-1][1].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[38:28] >= MIR[i-1][1].LIMIT[11:0] + (10H - TOLM)

Notes:

1. for MIR[0], MIR[i-1] is defined to be 0

Device: 16
Function: 1
Offset: 84h, 80h

Bit	Attr	Default	Description
15:4	RW	000h	**LIMIT** This field defines the highest address in the range A[37:28] prior to modification by the TOLM register. Note: The maximum value is 1FFFh (255.75GB), and the minimum value is 1h (256 MB). The most-significant bit of this field is ignored.
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).

3.9.21.3 AMIR[1:0] - 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: 90h, 8Ch

Bit	Attr	Default	Description
15:0	RW	0000h	**ADJLIMIT:** Adjusted MIR Limit

3.9.22 FB-DIMM Error Registers

3.9.22.1 FERR_FAT_FB-DIMM - FB-DIMM First Fatal Errors

The first fatal error for an FB-DIMM branch is flagged in these registers. Only one flag is ever set. Lower-numbered branches have higher priority than higher-numbered branches. Lower-numbered channels have higher priority than higher-numbered

channels. Higher-order error bits within a register have higher priority than lower-order bits. The FB-DIMMChan_Indx field is not an error. This register will display invalid index channel data until an error has occurred.

Device: 16
Function: 1
Offset: 98h

Bit	Attr	Default	Description
31:30	RV	00	Reserved
29:28	RW1CS	00	**FB-DIMMChan_Indx:** Logs channel in which the highest-order error occurred
27:23	RV	0h	Reserved
22	RW1CS	0	M23Err: Non-Redundant Fast Reset Timeout
21:3	RV	0h	Reserved
1	RW1CS	0	**M2Err:** Northbound CRC error on non-redundant retry
0	RW1CS	0	**M1Err:** Alert on non-redundant retry or fast reset timeout

3.9.22.2 NERR_FAT_FB-DIMM - FB-DIMM Next Fatal Errors

If an error is already flagged in FERR_FAT_FB-DIMM, subsequent and lower-priority fatal errors are logged in NERR_FAT_FB-DIMM.

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
1	RW1CS	0	**M2Err:** Northbound CRC error on non-redundant retry
0	RW1CS	0	**M1Err:** Memory Write error on non-redundant retry or FB-DIMM configuration Write error on retry

3.9.22.3 FERR_NF_FB-DIMM - FB-DIMM First Non-Fatal Errors

The first non-fatal error for an FB-DIMM branch is flagged in these registers. Only one flag is ever set. Lower-numbered branches have higher priority than higher-numbered branches. Lower-numbered channels have higher priority than higher-numbered channels. Higher-order error bits within a register have higher priority than lower-order bits. The FB-DIMMChan_Indx field is not an error. This register will display invalid index channel data until an error has occurred.

Device: 16
Function: 1
Offset: A0h

Bit	Attr	Default	Description
31:30	RV	00	Reserved

Device: 16 Function: 1 Offset: A0h			
Bit	**Attr**	**Default**	**Description**
29:28	RW1CS	00	FB-DIMMChan_Indx: Logs channel in which the error occurred 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.
27:26	RV	0h	Reserved
25	RW1CS	0	M29Err: DIMM-Isolation Completed
24	RW1CS	0	**M28Err:** DIMM-Spare Copy Completed
23	RW1CS	0	M27Err: Correctable Counter Threshold Exceeded
22	RW1CS	0	M26Err: Redundant Fast Reset Timeout
21	RW1CS	0	M25Err: Memory Write error on redundant retry
20	RW1CS	0	M24Err: Refresh error
19	RV	0	Reserved
18	RW1CS	0	**M22Err:** SPD protocol Error
17	RW1CS	0	**M21Err:** FB-DIMM Northbound CRC error on FB-DIMM Sync Status
16	RW1CS	0	**M20Err:** Correctable Patrol Data ECC
15	RW1CS	0	**M19Err:** Correctable Spare-Copy Data ECC
14	RW1CS	0	**M18Err:** Correctable Mirrored Demand Data ECC. Intel® 5400 chipset does not support memory mirroring
13	RW1CS	0	**M17Err:** Correctable Non-Mirrored Demand Data ECC
12	RW1CS	0	M16Err: Channel Failed-Over Occurred
11	RW1CS	0	M15Err: Memory or FB-DIMM configuration CRC read error
10	RW1CS	0	M14Err: FB-DIMM Configuration Write error on first attempt
9	RW1CS	0	M13Err: Memory Write error on first attempt
8	RW1CS	0	**M12Err:** Non-Aliased Uncorrectable Patrol Data ECC
7	RW1CS	0	**M11Err:** Non-Aliased Uncorrectable Spare-Copy Data ECC
6	RW1CS	0	**M10Err:** Non-Aliased Uncorrectable Mirrored Demand Data ECC
5	RW1CS	0	**M9Err:** Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC
4	RW1CS	0	**M8Err:** Aliased Uncorrectable Patrol Data ECC
3	RW1CS	0	**M7Err:** Aliased Uncorrectable Spare-Copy Data ECC
2	RW1CS	0	**M6Err:** Aliased Uncorrectable Mirrored Demand Data ECC. Intel® 5400 chipset does not support memory mirroring
1	RW1CS	0	**M5Err:** Aliased Uncorrectable Non-Mirrored Demand Data ECC
0	RW1CS	0	**M4Err:** Uncorrectable Data ECC on Replay

3.9.22.4 NERR_NF_FB-DIMM - FB-DIMM Next Fatal Errors

If an error is already flagged in FERR_NF_FB-DIMM, subsequent and lower-priority non-fatal errors are logged in NERR_NF_FB-DIMM.

Device: 16
Function: 1
Offset: A4h

Bit	Attr	Default	Description
31:26	RV	00h	Reserved
25	RW1CS	0	M29Err: DIMM-Isolation Completed
24	RW1CS	0	**M28Err:** DIMM-Spare Copy Completed
23	RW1CS RW1CS	0	M27Err: Correctable Counter Threshold Exceeded
22	RW1CS	0	M26Err: Redundant Fast Reset Timeout
21	RW1CS	0	M25Err: Memory Write error on redundant retry
20	RW1CS	0	M24Err: Refresh error
19	RV	0	Reserved
18	RW1CS	0	**M22Err:** SPD protocol Error
17	RW1CS	0	**M21Err:** FB-DIMM Northbound parity error on FB-DIMM Sync Status
16	RW1CS	0	**M20Err:** Correctable Patrol Data ECC
15	RW1CS	0	**M19Err:** Correctable Spare-Copy Data ECC
14	RW1CS	0	**M18Err:** Correctable Mirrored Demand Data ECC. Intel® 5400 chipset does not support memory mirroring.
13	RW1CS	0	**M17Err:** Correctable Non-Mirrored Demand Data ECC
12	RW1CS	0	M16Err: Channel Failed-Over Occurred
11	RW1CS	0	M15Err: Memory or FB-DIMM configuration CRC read error
10	RW1CS	0	M14Err: FB-DIMM Configuration Write error on first attempt
9	RW1CS	0	M13Err: Memory Write error on first attempt
8	RW1CS	0	**M12Err:** Non-Aliased Uncorrectable Patrol Data ECC
7	RW1CS	0	**M11Err:** Non-Aliased Uncorrectable Spare-Copy Data ECC
6	RW1CS	0	**M10Err:** Non-Aliased Uncorrectable Mirrored Demand Data ECC
5	RW1CS	0	**M9Err:** Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC
4	RW1CS	0	**M8Err:** Aliased Uncorrectable Patrol Data ECC
3	RW1CS	0	**M7Err:** Aliased Uncorrectable Spare-Copy Data ECC
2	RW1CS	0	**M6Err:** Aliased Uncorrectable Mirrored Demand Data ECC. Intel® 5400 chipset does not support memory mirroring.
1	RW1CS	0	**M5Err:** Aliased Uncorrectable Non-Mirrored Demand Data ECC
0	RW1CS	0	**M4Err:** Uncorrectable Data ECC on Replay

3.9.22.5 EMASK_FB-DIMM - FB-DIMM 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	RWS	1	M29Err: DIMM-Isolation Completed
27	RWS	1	M28Err: DIMM-Spare Copy Completed
26	RWS	1	M27Err: Correctable Counter Threshold Exceeded
25	RWS	1	M26Err: Redundant Fast Reset Timeout
24	RWS	1	M25Err: Memory Write error on redundant retry
23	RWS	1	M24Err: Refresh error
22	RWS	1	M23Err: Non-Redundant Fast Reset Timeout
21	RWS	1	M22Err: SPD protocol Error
20	RWS	1	M21Err: FB-DIMM Northbound parity error on FB-DIMM Sync Status
19	RWS	1	M20Err: Correctable Patrol Data ECC
18	RWS	1	M19Err: Correctable Resilver- or Spare-Copy Data ECC
17	RWS	1	**M18Err: Correctable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
16	RWS	1	M17Err: Correctable Non-Mirrored Demand Data ECC
15	RWS	1	M16Err: Channel Failed-Over Occurred
14	RWS	1	M15Err: Memory or FB-DIMM configuration CRC read error
13	RWS	1	M14Err: FB-DIMM Configuration Write error on first attempt
12	RWS	1	M13Err: Memory Write error on first attempt
11	RWS	1	M12Err: Non-Aliased Uncorrectable Patrol Data ECC
10	RWS	1	M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC
9	RWS	1	M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC
8	RWS	1	M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC
7	RWS	1	M8Err: Aliased Uncorrectable Patrol Data ECC
6	RWS	1	M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC
5	RWS	1	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
4	RWS	1	M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC
3	RWS	1	M4Err: Uncorrectable Data ECC on Replay
2	RV	0	Reserved
1	RWS	1	M2Err: Memory or FB-DIMM configuration CRC read error
0	RWS	1	M1Err: Memory Write error on non-redundant retry

3.9.22.6 ERR0_FB-DIMM: FB-DIMM 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	Reserved1
28	RW	1	M29Err: DIMM-Isolation Completed
27	RW	1	M28Err: DIMM-Spare Copy Completed
26	RW	1	M27Err: Correctable Counter Threshold Exceeded
25	RW	1	M26Err: Redundant Fast Reset Timeout
24	RW	1	M25Err: Memory Write error on redundant retry
23	RW	1	M24Err: Refresh error
22	RW	1	M23Err: Non-Redundant Fast Reset Timeout
21	RW	1	M22Err: SPD protocol Error
20	RW	1	M21Err: FB-DIMM Northbound parity error on FB-DIMM Sync Status
19	RW	1	M20Err: Correctable Patrol Data ECC
18	RW	1	M19Err: Correctable Resilver- or Spare-Copy Data ECC
17	RW	1	**M18Err: Correctable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
16	RW	1	M17Err: Correctable Non-Mirrored Demand Data ECC
15	RW	1	M16Err: Channel Failed-Over Occurred
14	RW	1	M15Err: Memory or FB-DIMM configuration CRC read error
13	RW	1	M14Err: FB-DIMM Configuration Write error on first attempt
12	RW	1	M13Err: Memory Write error on first attempt
11	RW	1	M12Err: Non-Aliased Uncorrectable Patrol Data ECC
10	RW	1	M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC
9	RW	1	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
8	RW	1	M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC
7	RW	1	M8Err: Aliased Uncorrectable Patrol Data ECC
6	RW	1	M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC
5	RW	1	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
4	RW	1	M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC
3	RW	1	M4Err: Uncorrectable Data ECC on Replay
2	RV	0	Reserved
1	RW	1	M2Err: Memory or FB-DIMM configuration CRC read error
0	RW	1	M1Err: Memory Write error on non-redundant retry

3.9.22.7 ERR1_FB-DIMM: FB-DIMM 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	Reserved1
28	RW	1	M29Err: DIMM-Isolation Completed
27	RW	1	M28Err: DIMM-Spare Copy Completed
26	RW	1	M27Err: Correctable Counter Threshold Exceeded
25	RW	1	M26Err: Redundant Fast Reset Timeout
24	RW	1	M25Err: Memory Write error on redundant retry
23	RW	1	M24Err: Refresh error
22	RW	1	M23Err: Non-Redundant Fast Reset Timeout
21	RW	1	M22Err: SPD protocol Error
20	RW	1	M21Err: FB-DIMM Northbound parity error on FB-DIMM Sync Status
19	RW	1	M20Err: Correctable Patrol Data ECC
18	RW	1	M19Err: Correctable Resilver- or Spare-Copy Data ECC
17	RW	1	**M18Err: Correctable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
16	RW	1	M17Err: Correctable Non-Mirrored Demand Data ECC
15	RW	1	M16Err: Channel Failed-Over Occurred
14	RW	1	M15Err: Memory or FB-DIMM configuration CRC read error
13	RW	1	M14Err: FB-DIMM Configuration Write error on first attempt
12	RW	1	M13Err: Memory Write error on first attempt
11	RW	1	M12Err: Non-Aliased Uncorrectable Patrol Data ECC
10	RW	1	M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC
9	RW	1	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
8	RW	1	M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC
7	RW	1	M8Err: Aliased Uncorrectable Patrol Data ECC
6	RW	1	M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC
5	RW	1	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
4	RW	1	M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC
3	RW	1	M4Err: Uncorrectable Data ECC on Replay
2	RV	0	Reserved
1	RW	1	M2Err: Memory or FB-DIMM configuration CRC read error
0	RW	1	M1Err: Memory Write error on non-redundant retry

3.9.22.8 ERR2_FB-DIMM: FB-DIMM 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	Reserved1
28	RW	1	M29Err: DIMM-Isolation Completed
27	RW	1	M28Err: DIMM-Spare Copy Completed
26	RW	1	M27Err: Correctable Counter Threshold Exceeded
25	RW	1	M26Err: Redundant Fast Reset Timeout
24	RW	1	M25Err: Memory Write error on redundant retry
23	RW	1	M24Err: Refresh error
22	RW	1	M23Err: Non-Redundant Fast Reset Timeout
21	RW	1	M22Err: SPD protocol Error
20	RW	1	M21Err: FB-DIMM Northbound parity error on FB-DIMM Sync Status
19	RW	1	M20Err: Correctable Patrol Data ECC
18	RW	1	M19Err: Correctable Resilver- or Spare-Copy Data ECC
17	RW	1	**M18Err: Correctable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
16	RW	1	M17Err: Correctable Non-Mirrored Demand Data ECC
15	RW	1	M16Err: Channel Failed-Over Occurred
14	RW	1	M15Err: Memory or FB-DIMM configuration CRC read error
13	RW	1	M14Err: FB-DIMM Configuration Write error on first attempt
12	RW	1	M13Err: Memory Write error on first attempt
11	RW	1	M12Err: Non-Aliased Uncorrectable Patrol Data ECC
10	RW	1	M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC
9	RW	1	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
8	RW	1	M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC
7	RW	1	M8Err: Aliased Uncorrectable Patrol Data ECC
6	RW	1	M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC
5	RW	1	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
4	RW	1	M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC
3	RW	1	M4Err: Uncorrectable Data ECC on Replay
2	RV	0	Reserved
1	RW	1	M2Err: Memory or FB-DIMM configuration CRC read error
0	RW	1	M1Err: Memory Write error on non-redundant retry

3.9.22.9 MCERR_FB-DIMM - FB-DIMM 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	Reserved1
28	RW	1	M29Err: DIMM-Isolation Completed
27	RW	1	M28Err: DIMM-Spare Copy Completed
26	RW	1	M27Err: Correctable Counter Threshold Exceeded
25	RW	1	M26Err: Redundant Fast Reset Timeout
24	RW	1	M25Err: Memory Write error on redundant retry
23	RW	1	M24Err: Refresh error
22	RW	1	M23Err: Non-Redundant Fast Reset Timeout
21	RW	1	M22Err: SPD protocol Error
20	RW	1	M21Err: FB-DIMM Northbound parity error on FB-DIMM Sync Status
19	RW	1	M20Err: Correctable Patrol Data ECC
18	RW	1	M19Err: Correctable Resilver- or Spare-Copy Data ECC
17	RW	1	**M18Err: Correctable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
16	RW	1	M17Err: Correctable Non-Mirrored Demand Data ECC
15	RW	1	M16Err: Channel Failed-Over Occurred
14	RW	1	M15Err: Memory or FB-DIMM configuration CRC read error
13	RW	1	M14Err: FB-DIMM Configuration Write error on first attempt
12	RW	1	M13Err: Memory Write error on first attempt
11	RW	1	M12Err: Non-Aliased Uncorrectable Patrol Data ECC
10	RW	1	M11Err: Non-Aliased Uncorrectable Resilver- or Spare-Copy Data ECC
9	RW	1	**M10Err: Non-Aliased Uncorrectable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
8	RW	1	M9Err: Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC
7	RW	1	M8Err: Aliased Uncorrectable Patrol Data ECC
6	RW	1	M7Err: Aliased Uncorrectable Resilver- or Spare-Copy Data ECC
5	RW	1	**M6Err: Aliased Uncorrectable Mirrored Demand Data ECC.** Intel® 5400 chipset does not support memory mirroring
4	RW	1	M5Err: Aliased Uncorrectable Non-Mirrored Demand Data ECC
3	RW	1	M4Err: Uncorrectable Data ECC on Replay
2	RV	0	Reserved
1	RW	1	M2Err: Memory or FB-DIMM configuration CRC read error
0	RW	1	M1Err: Memory Write error on non-redundant retry

3.9.22.10 Error Log Operation

The eight Error Logs consist of the NRECMEM, NRECFGLOG, NRECFB-DIMM, REDMEM, RECMEM, RECFGLOG, RECFB-DIMM, and RECFB-DIMMRP register sets. An Error Log is not valid until triggered by a bit in the FERR_FAT_FB-DIMM, NERR_FAT_FB-DIMM, FERR_NF_FB-DIMM, or NERR_NF_FB-DIMM "trigger" registers. See Figure 5-36, "Intel® 5400 Chipset Error List" on page 393 for the association between the errors and the Error Logs.

For the sake of brevity in the ensuing discussion "FERR" refers to FERR_FAT_FB-DIMM and FERR_NF_FB-DIMM, "NERR" refers to NERR_FAT_FB-DIMM and NERR_NF_FB-DIMM, "FAT" refers to FERR_FAT_FB-DIMM and NERR_FAT_FB-DIMM, and "NF" refers to FERR_NF_FB-DIMM and NERR_NF_FB-DIMM. The two severity classes are "FAT" and "NF".

A lower-order channel takes priority over a higher-order channel contending for the same error bit in the same cycle. Higher-order bits take priority over lower-order bits contending for the same Error Log in the same cycle in the same register. When both FERR's are cleared, then FAT FERR bits take priority over NF FERR bits contending for the same Error Log in the same cycle. When both FERR's are non-zero, then FAT NERR bits take priority over NF NERR bits contending for the same Error Log in the same cycle. Since NERR bits for a given severity class can't be set until a FERR bit has been set in that severity class, then FERR bits take temporal priority over NERR bits contending for the same Error Log. The "winner" triggers its Error Log. The bit that locked the log is recorded in the log's "MERR" field. A triggered Error Log is "locked" until it's trigger bit is cleared.

3.9.22.11 RECFB-DIMMRPA[1:0]: Recoverable FB-DIMM Replay Error Log Register A

The RECFB-DIMM Replay registers defined below (A through F) has data and ECC interlaced as defined in the *FB-DIMM Architecture and Protocol Specification, Rev. 0.8*. This register latches the replayed frame on the first northbound CRC error on a replay. (Note that the expected CRC bits are not captured.) This log has no "MERR" field because it can only be triggered by M15Err. However, it is not triggered by M15Err due to a patrol scrub read.

Note: The register will not capture the replayed northbound config read data with a CRC error.

Device: 22, 21
Function: 1
Offset: 40h

Bit	Attr	Default	Description
31:0	ROS	0h	BITS: Bits [31:0] of the packet

3.9.22.12 RECFB-DIMMRPB[1:0]: Recoverable FB-DIMM Replay Error Log Register B

This register latches information on the first detected non-fatal northbound CRC error on a replay.

Note: The register will not capture the replayed northbound config read data with a CRC error.

Device: 22, 21
Function: 1
Offset: 44h

Bit	Attr	Default	Description
31:0	ROS	0h	BITS: Bits [63:32] of the packet

3.9.22.13 RECFB-DIMMRPC[1:0]: Recoverable FB-DIMM Replay Error Log Register C

This register latches information on the first detected non-fatal northbound CRC error on a replay.

Note: The register will not capture the replayed northbound config read data with a CRC error.

Device: 22, 21
Function: 1
Offset: 48h

Bit	Attr	Default	Description
31:0	ROS	0h	BITS: Bits [95:64] of the packet

3.9.22.14 RECFB-DIMMRPD[1:0]: Recoverable FB-DIMM Replay Error Log Register D

This register latches information on the first detected non-fatal northbound CRC error on a replay.

Note: The register will not capture the replayed northbound config read data with a CRC error.

Device: 22, 21
Function: 1
Offset: 4Ch

Bit	Attr	Default	Description
31:0	ROS	0h	BITS: Bits [127:96] of the packet

3.9.22.15 RECFB-DIMMRPE[1:0]: Recoverable FB-DIMM Replay Error Log Register E

This register latches information on the first detected non-fatal northbound CRC error on a replay.

Note: Note: The register will not capture the replayed northbound config read data with a CRC error.

Device: 22, 21
Function: 1
Offset: 50h

Bit	Attr	Default	Description
31:0	ROS	0h	BITS: Bits [159:128] of the packet

3.9.22.16 RECFB-DIMMRPF[1:0]: Recoverable FB-DIMM Error Log Register F

This register latches information on the first detected non-fatal northbound CRC error on a replay

Note: The register will not capture the replayed northbound config read data with a CRC error.

Device: 22, 21
Function: 1
Offset: 54h

Bit	Attr	Default	Description
7:0	ROS	0h	BITS: Bits [167:160] of the packet

3.9.22.17 VALIDLOG[1:0]: Valid Log Markers

Each of the six Error Logs maintain their individual channel indices here. These indices complement the indices captured in FERR_FAT_FB-DIMM and FERR_NF_FB-DIMM.

Device: 22, 21
Function: 1
Offset: 70h

Bit	Attr	Default	Description
23	RW1CS	0	RECFB-DIMMRPVALID: RECFB-DIMMRP logs are valid '0' = RECFB-DIMMRP[A:F] register logs are invalid. '1' = RECFB-DIMMRP[A:F] register logs are valid; captured for an M15Err. Note: RECFB-DIMMRPVALID will not be set after RECFB-DIMMVALID due to a config read that encounters a NB CRC error. Consequently the replayed data will not be captured in RECFB-DIMMRP for this specific case.
22	RW1CS	0	RECFB-DIMMVALID: RECFB-DIMM and RECFB-DIMMIDLE logs are valid '0' = Invalid '1' = Valid
21	RW1CS	0	FB-DIMMChan_Indx_RECFB-DIMM: Logs channel which triggered RECFB-DIMM Valid only when associated error is set in FERR_NF_FB-DIMM or NERR_NF_FB-DIMM.

Device: 22, 21
Function: 1
Offset: 70h

Bit	Attr	Default	Description
20:19	ROS	00	RECFB-DIMMMERR: Indicates type of memory error captured in RECFB-DIMM '00' = M15Err (Read CRC error, excluding patrol scrub reads) '01' = M21Err (Sync Status error) '10' = M25Err or M24Err or M14Err or M13Err '11' = Reserved Note: M25Err=Memory write error on redundant retry, M24Err=Refresh error, M14Err=Config write error on 1st attempt, M13Err=Memory write error on 1st attempt. Note: Although patrol scrub reads cause an M15Err to occur, nothing is logged in RECFB-DIMM (or RECFB-DIMMRP)
18	RW1CS	0	RECFGLOGVALID: RECFGLOG log is valid '0' = Invalid '1' = Valid
17	RW1CS	00	FB-DIMMChan_Indx_RECFGLOG: Logs channel which triggered RECFGLOG Valid only when associated error is set in FERR_NF_FB-DIMM or NERR_NF_FB-DIMM.
16	RW1CS	0	RECFB-DIMMRPBadCRC: RECFB-DIMM Replay with Bad CRC This field in only valid if **RECFB-DIMMRPVALID** is asserted. '0' = Good CRC on replayed frame. '1' = Bad CRC on replayed frame.
15	RW1CS	0	REDMEMVALID: REDMEM log is valid '0' = Invalid '1' = Valid
14	RW1CS	0	RECMEMVALID: RECMEM log is valid '0' = Invalid '1' = Valid This bit isn't set until after the replay if the initial request produced an ECC-uncorrectable error.
13	RW1CS	0	FB-DIMMChan_Indx_RECMEM: Logs branch which triggered RECMEM Valid only when associated error is set in FERR_NF_FB-DIMM or NERR_NF_FB-DIMM. The least-significant-bit of this field has no significance.
12:11	RV	00	Reserved4
10	RW1CS	0	NRECFB-DIMMVALID: NRECFB-DIMM and NRECFB-DIMMIDLE logs are valid '0' = Invalid '1' = Valid
9	RW1CS	0	FB-DIMMChan_Indx_NRECFB-DIMM: Logs channel which triggered NRECFB-DIMM Valid only when associated error is set in FERR_NF_FB-DIMM or NERR_NF_FB-DIMM.
8:7	RV	00	Reserved3
6	RW1CS	0	NRECFGLOGVALID: NRECFGLOG log is valid '0' = Invalid '1' = Valid
5	RW1CS	00	FB-DIMMChan_Indx_NRECFGLOG: Logs channel which triggered NRECFGLOG Valid only when associated error is set in FERR_NF_FB-DIMM or NERR_NF_FB-DIMM.
4:3	RV	00	Reserved2

Device: 22, 21 Function: 1 Offset: 70h			
Bit	**Attr**	**Default**	**Description**
2	RW1CS	0	NRECMEMVALID: NRECMEM log is valid '0' = Invalid '1' = Valid
1	RW1CS	00	FB-DIMMChan_Indx_NRECMEM: Logs branch which triggered NRECMEM Valid only when associated error is set in FERR_FAT_FB-DIMM, NERR_FAT_FB-DIMM, FERR_NF_FB-DIMM, or NERR_NF_FB-DIMM. The least-significant-bit of this field has no significance.
0	RV	00	Reserved1

3.9.22.18 NRECFB-DIMMIDLE[1:0]: Non-Recoverable FB-DIMM IDLE Log Register

This register latches the northbound idle pattern for bits [13:0] of transfer 0 on the first detected fatal northbound CRC error.

Device: 22, 21 Function: 1 Offset: 92h			
Bit	**Attr**	**Default**	**Description**
15:14	RV	00	Reserved
13:0	ROS	0h	BITS: Bits [13:0] of transfer 0 of the packet During fail-over, bit [13] = bit [12].

3.9.22.19 RECFB-DIMMIDLE[1:0]: Recoverable FB-DIMM IDLE Log Register B

This register latches the northbound idle pattern for bits [13:0] of transfer 0 on the first detected non-fatal northbound CRC error.

Device: 22, 21 Function: 1 Offset: 96h			
Bit	**Attr**	**Default**	**Description**
15:14	RV	00	Reserved
13:0	ROS	0h	BITS: Bits [13:0] of transfer 0 of the packet During fail-over, bit [13] = bit [12].

3.9.22.20 NRECMEMA[1:0] - Non-Recoverable Memory Error Log Register A

This register latches information on the first detected fatal memory error.

Device: 22, 21 Function: 1 Offset: BEh			
Bit	**Attr**	**Default**	**Description**
15	RV	0	Reserved
14:12	ROS	0h	**BANK:** Bank of the failed request
11:8	ROS	0h	RANK: Rank of the failed request

Device: 22, 21 Function: 1 Offset: BEh			
Bit	**Attr**	**Default**	**Description**
7:0	ROS	00h	**REC_FB-DIMM_DM_BUF_ID:** DM Buffer ID of the failed request

3.9.22.21 NRECMEMB[1:0] - Non-Recoverable Memory Error Log Register B

This register latches information on the first detected fatal memory error.

Device: 22,21 Function: 1 Offset: C0h			
Bit	**Attr**	**Default**	**Description**
31	ROS	0	RDWR '0' = Read '1' = Write
30:29	ROS	0h	MERR: Identifies the error that triggered this log: '00' = M1 '01' = M2 '10' = Reserved '11' = M4
28:16	ROS	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	ROS	0h	**RAS:** RAS address of the failed request

3.9.22.22 NRECFGLOG[1:0] - Non-Recoverable DIMM Configuration Access Error Log Register

This register latches information on the first detected non-fatal DIMM configuration register access.

Device: 22,21 Function: 1 Offset: 74h			
Bit	**Attr**	**Default**	**Description**
31:29	RV	0h	Reserved
28	ROS	0h	MERR: Identifies the error that triggered this log: '0' = M1 '1' = M2
27:24	ROS	0h	**BE:** Byte Enables of the failed request
23:16	ROS	00h	**REG:** Register Address of the failed request
15:12	RV	0h	Reserved
11	ROS	0	**RDWR** '0' = Read '1' = Write
10:8	ROS	0h	**FUNCTION:** Function Number of the failed request
7:0	ROS	00h	**CFG_FB-DIMM_DM_BUF_ID:** DM Buffer ID of the failed request

3.9.22.23 NRECFB-DIMMA[1:0]: Non-Recoverable FB-DIMM Error Log Register A

The NRECFB-DIMM registers defined below (A through E) have the following mapping: This register latches information on the first detected fatal northbound CRC error.

Device: 22,21 Function: 1 Offset: C4h			
Bit	**Attr**	**Default**	**Description**
31:0	ROS	0h	**BITS:** Bits [31:0] of the packet

3.9.22.24 NRECFB-DIMMB[1:0] - Non-Recoverable FB-DIMM Error Log Register B

This register latches information on the first detected fatal northbound CRC error.

Device: 22,21 Function: 1 Offset: C8h			
Bit	**Attr**	**Default**	**Description**
31:0	ROS	0h	**BITS:** Bits [63:32] of the packet

3.9.22.25 NRECFB-DIMMC[1:0] - Non-Recoverable FB-DIMM Error Log Register C

This register latches information on the first detected fatal northbound CRC error.

Device: 22,21 Function: 1 Offset: CCh			
Bit	**Attr**	**Default**	**Description**
31:0	ROS	0h	**BITS:** Bits [95:64] of the packet

3.9.22.26 NRECFB-DIMMD[1:0] - Non-Recoverable FB-DIMM Error Log Register D

This register latches information on the first detected fatal northbound CRC error.

Device: 22,21 Function: 1 Offset: D0h			
Bit	**Attr**	**Default**	**Description**
31:0	ROS	0h	**BITS:** Bits [127:96] of the packet

3.9.22.27 NRECFB-DIMME[1:0] - Non-Recoverable FB-DIMM Error Log Register E

This register latches information on the first detected fatal northbound CRC error.

Device: 22,21
Function: 1
Offset: D4h

Bit	Attr	Default	Description
31:28	RV	0h	Reserved
27:0	ROS	0h	**BITS:** Bits [155:128] of the packet

3.9.22.28 NRECFB-DIMMF[1:0]: Non-Recoverable FB-DIMM Error Log Register F

This register latches information on the first detected fatal northbound CRC error.

Device: 22,21
Function: 1
Offset: D8h

Bit	Attr	Default	Description
31:26	ROS	0h	HOSTCRCD: MCH's CRC calculation D This is bits E2.6 through E2.11 of the FB-DIMM Architecture and Protocol Specification, Rev. 0.8. In 13b NB mode, these bits are zero.
25:20	ROS	0h	HOSTCRCC: MCH's CRC calculation C This is bits E2.0 through E2.5 of the FB-DIMM Architecture and Protocol Specification, Rev. 0.8.
19:14	ROS	0h	HOSTCRCB: MCH's CRC calculation B This is bits E1.6 through E1.11 of the FB-DIMM Architecture and Protocol Specification, Rev. 0.8. In 13b NB mode, these bits are zero.
13:8	ROS	0h	HOSTCRCA: MCH's CRC calculation A This is bits E1.0 through E1.5 of the FB-DIMM Architecture and Protocol Specification, Rev. 0.8.
7:0	ROS	0h	BITS: Bits [167:160] of the packet

3.9.22.29 REDMEMB[1:0]: Recoverable Memory Data Error Log Register B

This register latches information on the first detected correctable ECC error.

Device: 22,21
Function: 1
Offset: 7Ch

Bit	Attr	Default	Description
31:18	RV	0	*Reserved*
17:0	ROS	0h	**ECC _Locator:** identifies the adjacent symbol pair in error for correctable errors according to Table 3-50, "ECC Locator Mapping Information", Figure 5-18, "Code Layout for Single-Channel Branches" on page 314 and Figure 5-19, "Code Layout for Dual-Channel Branches" on page 315.

3.9.22.30 RECMEMA[1:0] - Recoverable Memory Error Log Register A

Table 3-50. 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

This register latches information on the first detected non-fatal memory error.

Table 3-51. RECMEM Err to MERR Association

Err Trigger	MERR Encoding
M5	5
M6	6
M7	7
M8	8
M9	9
M10	10
M11	11
M12	12
M13	13
M17	17
M18	18
M19	19
M20	20
M26	26

Device: 22,21 Function: 1 Offset: E0h			
Bit	**Attr**	**Default**	**Description**
23:20	RV	0	Reserved
19:15	ROS	0h	**MERR:** identifies the error that triggered the RECMEM and/or the REDMEM logs according to Table 3-51.
14:12	ROS	0h	BANK: Bank of the failed request
11:8	ROS	0h	RANK: Rank of the failed request
7:0	ROS	00h	**REC_FB-DIMM_DM_BUF_ID:** DM Buffer ID of the failed request

3.9.22.31 RECMEMB[1:0] - Recoverable Memory Error Log Register B

This register latches information on the first detected non-fatal memory error.

Device: 22,21 Function: 1 Offset: E4h			
Bit	**Attr**	**Default**	**Description**
31	ROS	0	RDWR '0' = Read '1' = Write
30:29	RV	00	Reserved
28:16	ROS	000h	CAS: CAS address of the failed request The CAS address will map from 12:0 while bit 10 is hardwired to 0.
15:0	ROS	0h	**RAS:** RAS address of the failed request

3.9.22.32 DIMMISO: DIMM Isolation Register

This register latches information on the replayed transaction that initially triggered the RECMEM log. This log is only triggered by error M29.

Table 3-52. DIMM Isolation for Uncorrectable Errors

PERSISTENT	DIMM	Description
0	01	single-channel uncorrectable
	x1	lockstep non-mirrored transient uncorrectable
	1x	
1	xx	lockstep non-mirrored persistent uncorrectable

Device: 16 Function: 1 Offset: BCh			
Bit	**Attr**	**Default**	**Description**
15	ROS	0	BRANCH: Branch of the failed request
14	ROS	000h	PERSISTENT: DIMM that generated the uncorrectable error could not be isolated '0' = Transient uncorrectable: the "DIMM" field of this register is valid. '1' = Persistent uncorrectable: the "DIMM" field of this register is invalid.
13:12	ROS	000h	DIMM: DIMM which generated the uncorrectable error Identifies the DIMM(s) responsible for the uncorrectable error: 'x1' = the DIMM on channel 0 or channel 2 was faulty '1x' = the DIMM on channel 1 or channel 3 was faulty If both bits are zero, then this log is not valid.
11:8	ROS	0h	RANK: Rank of the failed request
7:0	ROS	00h	ISO_FB-DIMM_DM_BUF_ID: DM Buffer ID of the failed request

3.9.22.33 RECFGLOG[1:0] - Recoverable DIMM Configuration Access Error Log Register

This register latches information on the first detected fatal DIMM configuration register access.

Device: 22,21 Function: 1 Offset: 78h			
Bit	**Attr**	**Default**	**Description**
31:29	RV	0h	Reserved
28	ROS	0h	MERR: Identifies the error that triggered this log: '0' = M14 '1' = M15
27:24	ROS	0h	**BE:** Byte Enables of the failed request
23:16	ROS	00h	**REG:** Register Address of the failed request
15:12	RV	0h	Reserved
11	ROS	0	**RDWR** '0' = Read '1' = Write
10:8	ROS	0h	**FUNCTION:** Function Number of the failed request
7:0	ROS	00h	**CFG_FB-DIMM_CE_BUF_ID:** DM Buffer ID of the failed request

3.9.22.34 RECFB-DIMMA[1:0] - Recoverable FB-DIMM Error Log Register A

This register latches information on the first northbound CRC error.

Device: 22,21 Function: 1 Offset: E8h			
Bit	**Attr**	**Default**	**Description**
31:0	ROS	0h	**BITS:** Bits [31:0] of the packet

3.9.22.35 RECFB-DIMMB[1:0] - Recoverable FB-DIMM Error Log Register B

This register latches information on the first detected non-fatal northbound CRC error.

Device: 22,21
Function: 1
Offset: ECh

Bit	Attr	Default	Description
31:0	ROS	0h	**BITS:** Bits [63:32] of the packet

3.9.22.36 RECFB-DIMMC[1:0] - Recoverable FB-DIMM Error Log Register C

This register latches information on the first detected non-fatal northbound CRC error.

Device: 22,21
Function: 1
Offset: F0h

Bit	Attr	Default	Description
31:0	ROS	0h	**BITS:** Bits [95:64] of the packet

3.9.22.37 RECFB-DIMMD[1:0] - Recoverable FB-DIMM Error Log Register D

This register latches information on the first detected non-fatal northbound CRC error.

Device: 22,21
Function: 1
Offset: F4h

Bit	Attr	Default	Description
31:0	ROS	0h	**BITS:** Bits [127:96] of the packet

3.9.22.38 RECFB-DIMME[1:0] - Recoverable FB-DIMM Error Log Register E

This register latches information on the first detected non-fatal northbound CRC error.

Device: 22,21
Function: 1
Offset: F8h

Bit	Attr	Default	Description
31:0	ROS	0h	**BITS:** Bits [159:128] of the packet

3.9.22.39 RECFB-DIMMF[1:0]: Recoverable FB-DIMM Error Log Register F

This register latches information on the first detected non-fatal northbound CRC error.

Device: 22, 21 Function: 1 Offset: FCh			
Bit	**Attr**	**Default**	**Description**
31:26	ROS	0h	HOSTCRCD: MCH's CRC calculation D This is bits E2.6 through E2.11 of the FB-DIMM Architecture and Protocol Specification, Rev. 0.8. In 13b NB mode, these bits are zero.
25:20	ROS	0h	HOSTCRCC: MCH's CRC calculation C This is bits E2.0 through E2.5 of the FB-DIMM Architecture and Protocol Specification, Rev. 0.8.
19:14	ROS	0h	HOSTCRCB: MCH's CRC calculation B This is bits E1.6 through E1.11 of the FB-DIMM Architecture and Protocol Specification, Rev. 0.8. In 13b NB mode, these bits are zero.
13:8	ROS	0h	HOSTCRCA: MCH's CRC calculation A This is bits E1.0 through E1.5 of the FB-DIMM Architecture and Protocol Specification, Rev. 0.8.
7:0	ROS	0h	BITS: Bits [167:160] of the packet

3.9.23 FB-DIMM Branch Registers

There are two sets of the following registers, one set for each FB-DIMM branch. They each appear in function 0 of different devices as shown in Functions Specifically handled by MCH: Table 3-3.

3.9.23.1 FB-DIMMLVL[1:0][1:0] - FB-DIMM Packet Levelization

This register controls the FB-DIMM 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 from issue of the FB-DIMM channel's southbound TS2 packet header to the arrival of its northbound response header.

3.9.23.2 FB-DIMMHPC[1:0]: FB-DIMM State Control

This register controls the FB-DIMM channel for Initialization. It consists of a next State field.

The index in FB-DIMMHPC[index] associates the FB-DIMMHPC with branch[index]. FB-DIMMHPC[0] is associated with FB-DIMM branch 0, FB-DIMMHPC[1] is associated with FB-DIMM branch 1.

When software writes to FB-DIMMHPC[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 MCH hardware transitions FB-DIMMST.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[1]: 22, 21 Function: 0 Offset: 4Fh			
Bit	**Attr**	**Default**	**Description**
7:0	RW	00h	NEXTSTATE: FB-DIMM 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**[2] 30h: **Active**[2] 40h: **Redundant**[2] 50h: Disabled 60h: **Redundancy Loss**[2] - may not be written D0h: Fault

Notes:

1. The nomenclature is Device 22 (branch 1), 21 (branch 0)
2. Both sync and refresh packets are sent during this mode.

3.9.23.3 FB-DIMMST[1:0] - FB-DIMM Status

These registers are inspected by software to determine the current FB-DIMM branch state. This register contains Mirroring recovery state, and Initialization state.

The indexing scheme is the same as in FB-DIMMHPC registers. The current FB-DIMM 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: 22,21 Function: 0 Offset: 4Bh			
Bit	**Attr**	**Default**	**Description**
7:0	RO	00h	**STATE:** FB-DIMM Branch State This field describes the current state of the FB-DIMM branch. It can be read by software to determine which FB-DIMM branch is being sequenced through recovery, and how far the FB-DIMM branch has progressed. 00h: Reset 10h: Init 20h: Ready 30h: Active 40h: Redundant 50h: Disabled 60h: Redundancy Loss - may not be written 70h: Recovery Reset - (should only be selected when MC.MIRROR is set) 80h: Recovery Init - (should only be selected when MC.MIRROR is set) 90h: Recovery Ready - (should only be selected when MC.MIRROR is set) A0h: Reserved B0h: Recovery Fault C0h: Recovery Failed D0h: Fault

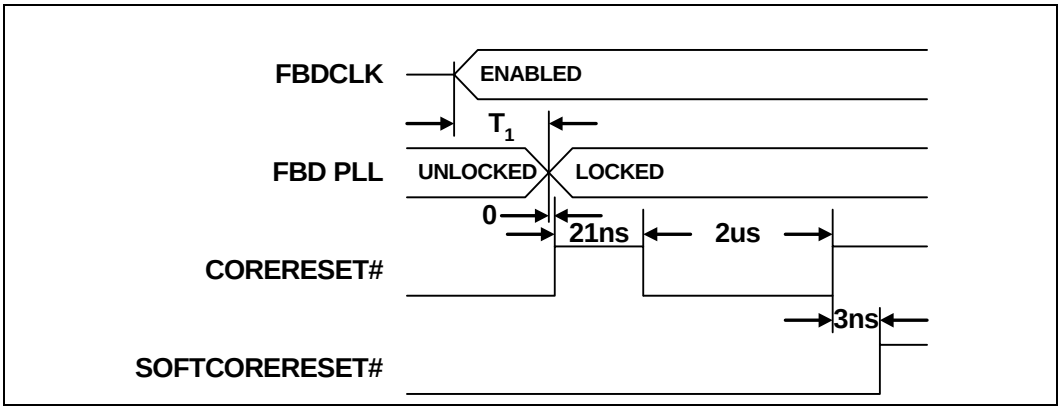
3.9.23.4 FB-DIMMRST[1:0] - FB-DIMM Reset

The FB-DIMM I/O blocks are reset separately from the rest of the Intel® 5400 chipset MCH. These blocks, composed of FAST-clocked (GHz unit-interval clocked) logic, are supplied by a PLL whose FB-DIMMCLK is not available when PWRGOOD is asserted. After FB-DIMMCLK is enabled and the FB-DIMM PLL has acquired lock, CORERESET# is deasserted for a minimum of 21 ns, then asserted for a minimum of 2 µs. After the 2 µs assertion, CORERESET# is deasserted followed by a minimum delay of 3 ns at which time SOFTCORERESET# is deasserted. If the platform removes FB-DIMMCLK on a hot-remove of the branch, CORERESET# and SOFTCORERESET# must be asserted prior to loss of FB-DIMMCLK.

A "disabled" (not enabled) FB-DIMMCLK is floated at the source, and pulled to ground through the termination near the receiver in the Intel® 5400 chipset MCH.

All timing specifications in Figure 3-5 are minimums. After the sequence in Figure 3-5 has been executed, the FB-DIMM branch is ready for initialization.

Figure 3-5. FB-DIMM Reset Timing



Device: 22,21
Function: 0
Offset: 53h

Bit	Attr	Default	Description
7:3	RV	00h	Reserved
2	RWS	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.
1	RWS	0	SOFTCORERESET#: Soft Core Reset See Timing diagram Figure 3-5. 0: Soft Core Reset Asserted 1: Soft Core Reset De-Asserted
0	RWS	0	CORERESET#: Core Reset See Timing diagram Figure 3-5. 0: Core Reset Asserted 1: Core Reset De-Asserted

3.9.23.5 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 its 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.
5	RW	0	SPAREN: Spare Control Enable '1' enables sparing, '0' disables sparing. The SPRANK field defines other characteristics of the sparing operation. 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, 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.

3.9.23.6 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. Generates error M27. 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
3:0	RO	000	**FR: Failed Rank** Rank that was spared. Updated with the CERRCNT rank that has reached threshold when DSCIP is set.

3.9.23.7 MTR[1:0][3:0] - Memory Technology Registers

These registers define the organization of the DIMM's. There is one MTR for each pair of slots comprising either one or two ranks. The parameters for these devices can be obtained by serial presence detect.

MTR[0][3:0] defines slot-pairs [3:0] on branch[0]. MTR[1][3:0] defines slot-pairs [3:0] on branch[1]. MTR[3:0] in Table 3-23 is MTR[3:0] for Device 21 which is MTR[0][3:0] for this Section 3.9.23.7. MTR[3:0] in Table 3-23 is MTR[3:0] for Device 22 which is MTR[1][3:0] for this Section 3.9.23.7.

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")

Device: 22,21
Function: 0
Offset: 86h, 84h, 82h, 80h

Bit	Attr	Default	Description
15:11	RV	00h	Reserved1
10	RW	0	**PRESENT: DIMMs are present** This bit is set if both DIMM's are present and their technologies are compatible.
9	RW	0	**ETHROTTLE: Technology - Electrical Throttle** Defines the electrical throttling level for these DIMM's: '0' = Electrical Throttling is disabled '1' = Electrical Throttling is enabled using the throttling level defined by the DRTA.TFAW configuration field.
8	RW	0	**WIDTH: Technology - Width** Defines the data width of the SDRAMs used on these DIMM's '0' = x4 (4 bits wide) '1' = x8 (8 bits wide)
7	RV	00h	Reserved
6	RW	0	**NUMBANK: Technology - Number of Banks** Defines the number of (real, not shadow) banks on these DIMM's '0' = four-banked '1' = eight-banked
5	RW	0	**NUMRANK: Technology - Number of Ranks** Defines the number of ranks on these DIMM's. '0' = single ranked '1' = double ranked
4	RV	00h	Reserved
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

3.9.23.8 DMIR[1:0][4:0] - 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 with 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: 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	RV	0	Reserved
14:12	RW	000	**RANK3** Defines which rank participates in WAY3.
11	RV	0h	Reserved
10:8	RW	000	**RANK2** Defines which rank participates in WAY2.
7	RV	0h	Reserved
6:4	RW	000	**RANK1** Defines which rank participates in WAY1.
3	RV	0h	Reserved
2:0	RW	000	**RANK0** Defines which rank participates in WAY0.

3.9.23.9 FB-DIMMICMD[1:0][1:0] - FB-DIMM Initialization Command

These registers define channel behavior during the "Init" and "Reset" FB-DIMMST configuration register states. The "AMBID" field for the even-numbered channel also defines branch behavior during fast reset. This register cannot be manipulated until its corresponding FB-DIMMRST register's reset sequence has been executed. For channels 0 and 1, the reset sequence on FB-DIMMRST[0] must be completed. For channels 2 and 3, the reset sequence on FB-DIMMRST[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 specified AMB (not valid in "Reset") "001"=TS1: Training Sequence 1 to last AMB (not valid in "Reset") "010"=TS2: Training Sequence 2 to specified 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 FB-DIMMST.STATE="Reset" or "Recovery Reset" and EN='1'. The note '(valid only in "Reset")' indicates that this is valid only when FB-DIMMST.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.

3.9.23.10 FB-DIMMISTS[1:0][1:0] - FB-DIMM Initialization Status

The contents of this register are valid only during "Initialization" states. The fourteen bits [13:0] correspond to the northbound bit-lanes.

Device: 22,21
Function: 0
Offset: 5Ah, 58h

Bit	Attr	Default	Description
15:14	RV	00	Reserved

Device: 22,21 Function: 0 Offset: 5Ah, 58h			
Bit	**Attr**	**Default**	**Description**
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 FB-DIMMICMD.EN and FB-DIMMICMD.INITPAT fields. Bit-Lane status is evaluated on each change to the FB-DIMMICMD.EN and FB-DIMMICMD.INITPAT.Only bits [2:0] are valid during electrical idle, and only after the FB-DIMMRST reset sequence has been executed. A recognizable training sequence must contain the FB-DIMMICMD.AMBID. TS1 detection is qualified by test patterns specified in section 4.3 of *rev. 0.75 of FB-DIMM DFx specification*, which defines the "SB/NB_Mapping" (1 bit), the "Test Parameters" (24 bits), and the "Electrical Stress Pattern".

3.9.23.11 FB-DIMMCHCFG[1:0][1:0] - FB-DIMM Channel Configuration

Device: 22,21 Function: 0 Offset: 4Ch, 48h			
Bit	**Attr**	**Default**	**Description**
15:9	RV	00	Reserved
8	RW	0	FS: Failover Sticky "1": MCH will not attempt to retrain failed lanes during subsequent fast reset. A previously failed lane will not gate forward progress during TS0, and nondefault NBCFG and SBCFG values will not be updated during fast reset. "0": MCH will always attempt to train lanes during TS0 until all lanes are trained or until bit lock timeout. NBCFG and SBCFG will be updated during fast reset TS1
7:4	RW	Fh	NBCFG: Northbound Channel Configuration This value is sent in the TS3 Config state during FB-DIMM initialization. If failover is enabled, this value will be updated during fast reset according to TS1 Testing state results.
3:0	RW	Fh	SBCFG: Southbound Channel Configuration This value is sent in the TS3 Config state during FB-DIMM initialization. If failover is enabled, this value will be updated during fast reset according to TS1 Testing state results

3.9.23.12 AMBPRESENT[1:0][1:0] - FB-DIMM AMB Slot Present Register

These registers control configuration transaction routing to AMB slots on a per FB-DIMM channel basis. This includes both accesses through memory mapped region (based on AMBASE register, see Section 3.8.3.1) and AMBSELECT. Software needs to program this register after SPD discovery process. Intel® 5400 chipset MCH will check this register before it sends actual FB-DIMM AMB configuration transaction.

Device: 22,21 Function: 0 Offset: 66h, 64h			
Bit	**Attr**	**Default**	**Description**
15:0	RW	0h	**AMBSP: Slot [bit_position] present in the FB-DIMM channel** 1: Indicates AMB slot addressed by DS[3:0] in decimal = [bit_position] is present; configuration transaction will be routed to FB-DIMM 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 FB-DIMM channel.

3.9.24 FB-DIMM RAS Registers

There are two sets of the following registers, one set for each FB-DIMM branch. They each appear in function 0 of different devices as shown in Table 3-3.

3.9.24.1 CERRCNTA[1:0][1:0] - Correctable Error Count A

These registers implement the “leaky-bucket” counters for correctable errors for each rank. Each field “limits” at a value of “15” (“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” 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: Aliased uncorrectable errors are counted as correctable errors in the implementation of this register.

Device: 22,21 Function: 0 Offset: F0h, E0h			
Bit	**Attr**	**Default**	**Description**
31:24	RW1CS	0h	RANK3: Error Count for Rank 3
23:16	RW1CS	0h	RANK2: Error Count for Rank 2
15:8	RW1CS	0h	RANK1: Error Count for Rank 1
7:0	RW1CS	0h	RANK0: Error Count for Rank 0

3.9.24.2 CERRCNTB[1:0][1:0]: Correctable Error Count B

Device: 22, 21
Function: 0
Offset: F4h, E4h

Bit	Attr	Default	Description
31:24	RW1CS	0h	RANK7: Error Count for Rank 7
23:16	RW1CS	0h	RANK6: Error Count for Rank 6
15:8	RW1CS	0h	RANK5: Error Count for Rank 5
7:0	RW1CS	0h	RANK4: Error Count for Rank 4

3.9.24.3 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 0000h" 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	00	Reserved
29:25	RW1CS	00h	**RANK5:** Bad device in Rank 5
24:20	RW1CS	00h	**RANK4:** Bad device in Rank 4
19:15	RW1CS	00h	**RANK3:** Bad device in Rank 3
14:10	RW1CS	00h	**RANK2:** Bad device in Rank 2
9:5	RW1CS	00h	**RANK1:** Bad device in Rank 1
4:0	RW1CS	00h	**RANK0:** Bad device in Rank 0

3.9.24.4 BADRAMB[1:0] - Bad DRAM Marker B

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 DRAM's are presumed "good". Only ranks containing x8 DRAM are "marked".

Device: 22,21
Function: 0
Offset: C4h

Bit	Attr	Default	Description
31:10	RV	00h	Reserved
9:5	RW1CS	00h	**RANK7:** Bad device in Rank 7
4:0	RW1CS	00h	**RANK6:** Bad device in Rank 6

3.9.24.5 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	RW1CS	0000	**RANK7:** Adjacent x8 symbol error count in Rank 7
27:24	RW1CS	0000	**RANK6:** Adjacent x8 symbol error count in Rank 6
23:20	RW1CS	0000	**RANK5:** Adjacent x8 symbol error count in Rank 5
19:16	RW1CS	0000	**RANK4:** Adjacent x8 symbol error count in Rank 4
15:12	RW1CS	0000	**RANK3:** Adjacent x8 symbol error count in Rank 3
11:8	RW1CS	0000	**RANK2:** Adjacent x8 symbol error count in Rank 2
7:4	RW1CS	0000	**RANK1:** Adjacent x8 symbol error count in Rank 1
3:0	RW1CS	0000	**RANK0:** Adjacent x8 symbol error count in Rank 0

3.10 I/OxAPIC Register Definitions

3.10.1 I/OxAPIC I/O Configuration Space

Note that in general, all register bits in the standard PCI header space (offset 0-3F) or in any OS-visible capability registers, that control the address decode like MSE, IOSE, VGAEN or otherwise control transaction forwarding must be treated as dynamic bits in the sense that these register bits could be changed by the OS when there is traffic flowing through the MCH. Note that the address register themselves can be treated as static in the sense that they will not be changed without the decode control bits being clear. Registers outside of the standard header space will be noted as dynamic when appropriate.

3.10.1.1 PCICMD[18]: PCI Command Register

This register defines the PCI 3.0 compatible command register values applicable to PCI Express space.

Device: 18
Function: 0
Offset: 04h

Bit	Attr	Default	Description
15:11	RV	00h	*Reserved.* (by PCI SIG)
10	RO	0	**Interrupt Disable** Controls the ability of the device to generate legacy INTx interrupt (when legacy INTx mode is enabled). This bit has no affect on the operation of the IOxAPIC as it does not generate legacy interrupts and is therefore hardwired to 0.
9	RO	0	**Fast Back-to-Back Enable** Not applicable to IOxAPIC and is hardwired to 0
8	RO	0	SERR Enable This bit has no impact on error reporting from the I/OxAPIC. Hardwired to 0.
7	RO	0	IDSEL Stepping/Wait Cycle Control Not applicable to internal MCH devices. Hardwired to 0.
6	RO	0	Parity Error Response This bit has no impact on error reporting from the I/OxAPIC. Hardwired to 0.
5	RO	0	**VGA palette snoop Enable** Not applicable to internal MCH devices. Hardwired to 0.
4	RO	0	**Memory Write and Invalidate Enable** Not applicable to internal MCH devices. Hardwired to 0.
3	RO	0	Special Cycle Enable Not applicable to IOxAPIC. Hardwired to 0.
2	RW	0	**Bus Master Enable** Controls the ability of the port in generating/forwarding memory (including MSI writes) or I/O transactions (and not messages) or configuration transactions from the secondary side to the primary side. For the I/OxAPIC, this bit enables it to generate memory write/"APIC msg". 1: Enables the I/OxAPIC to generate memory requests. 0: The Bus Master is disabled. When this bit is 0, I/OxAPIC cannot generate any memory transactions.
1	RW	0	**Memory Space Enable** 1: Enables I/OxAPIC's MBAR register (ABAR range decode is not enabled by this bit) to be decoded as valid target addresses. 0: Disables the I/OxAPIC's MBAR register to be decoded as valid target addresses for transactions.
0	RO	0	**IO Space Enable** Not applicable to IOxAPIC. Hardwired to 0

3.10.1.2 PCISTS[18]: PCI Status Register

The PCI Status register is a 16-bit status register that reports the occurrence of various events associated with the primary side of the primary side of the IOxAPIC on the internal MCH bus.

Device: 18
Function: 0
Offset: 06h

Bit	Attr	Default	Description
15	RWC	0	Detected Parity Error This bit is set by a device when it detects an uncorrectable data or address/control parity error. The setting of this bit is regardless of the Parity Error Response bit (PERRE) in the PCICMD register. (This bit is set when RTE parity errors are detected.)
14	RO	0	**Signaled System Error** 0: This device does not report fatal/non-fatal errors with the SERRE bit enabled. For I/OxAPIC, SERRE is hardwired to 0, and therefore so is this bit.
13	RO	0	Received Master Abort Not applicable to IOxAPIC. Hardwired to 0.
12	RO	0	Received Target Abort Not applicable to IOxAPIC. Hardwired to 0.
11	RW1C	0	Signaled Target Abort This bit is set when a device signals a completer abort completion status on the primary side (internal bus of MCH). The I/OxAPIC sets this bit when it receives config/memory transactions larger than a DWORD or cross a DWORD boundary. For MCH, the I/OxAPIC will never see transactions larger than a DWORD, so this bit will never be set; effectively RO 0.
10:9	RO	0h	DEVSEL# Timing Not applicable to IOxAPIC. Hardwired to 0.
8	RO	0	Master Data Parity Error This bit is set by a device if the Parity Error Response bit in the PCI Command register is set and it receives a completion with poisoned data from the primary side or if it forwards a packet with data (including MSI writes) to the primary side with poison. For the I/OxAPIC, the parity error response bit is hardwired to 0, and therefore so is this bit.
7	RO	0	Fast Back-to-Back Not applicable to IOxAPIC. Hardwired to 0.
6	RO	0	*Reserved*
5	RO	0	66 MHz capable Not applicable to IOxAPIC. Hardwired to 0.
4	RO	1	**Capabilities List** This bit indicates the presence of a capabilities list structure
3	RO	0	**INTx Status** Not applicable to IOxAPIC. Hardwired to 0
2:0	RV	0h	*Reserved*

3.10.1.3 CCR[18]: Class Code Register

This register contains the Class Code for the device.

Device: 18 Function: 0 Offset: 09h			
Bit	**Attr**	**Default**	**Description**
23:16	RO	08h	Base Class This field defaults to 08h, indicating it is a “Generic System Peripheral”.
15:8	RO	00h	Sub-Class This field defaults to 00h indicating “Interrupt Controller”.
7:0	RO	20h	Register-Level Programming Interface This field defaults to 20h indicating “I/OxAPIC”.

3.10.1.4 CLS[18]: Cacheline Size Register

Device: 18 Function: 0 Offset: 0Ch			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0	Cacheline Size This register is set as RW for compatibility reasons only. Cacheline size for MCH is always 64B. MCH hardware ignore this setting.

3.10.1.5 MBAR: IOxAPIC Memory Base Address Register

Device: 18 Function: 0 Offset: 10h			
Bit	**Attr**	**Default**	**Description**
31:12	RW	FEC89h	BAR This marks the 4 KB aligned 32-bit base address for memory-mapped registers of IOxAPIC
11:4	RV	0h	Reserved
3	RO	0	Prefetchable The IOxAPIC registers are not prefetchable.
2:1	RO	00	Type The IOAPIC registers can only be placed below 4G system address space.
0	RO	0	Memory Space This Base Address Register indicates memory space.

3.10.1.6 CAPPTR[18]: Capability Pointer

The CAPPTR is used to point to a linked list of additional capabilities implemented by the device. It provides the offset to the first set of capabilities registers located in the PCI compatible space from 40h.

Device: 18
Function: 0
Offset: 34h

Bit	Attr	Default	Description
7:0	RO	6Ch	Capability Pointer Points to the first capability structure for the device.

3.10.1.7 INTL[18]: Interrupt Line Register

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

Device: 18
Function: 0
Offset: 3Ch

Bit	Attr	Default	Description
7:0	RO	00h	**Interrupt Line** This bit is RW for devices that can generate a legacy INTx message and is needed only for compatibility purposes. Not applicable to IOxAPIC. Hardwired to 0.

3.10.1.8 INTP[18]: Interrupt Pin Register

Indicates what INTx message a device generates. Per PCI rules, all devices that generate only one interrupt must use INTA. Hence all devices in MCH use INTA.

Device: 18
Function: 0
Offset: 3Dh

Bit	Attr	Default	Description
7:0	RO	00h	Interrupt Pin Not applicable to IOxAPIC. Hardwired to 0.

3.10.1.9 ABAR: I/OxAPIC Alternate BAR

Past designs use this register to define a memory address region from FECX_YZ00 to FECX_YZFF which is a 256B range. Current designs use a 4 KB range, so the Z field of the register for address decode is ignored. The effective memory address region is therefore from FECX_Y000 to FECX_YFFF.

Device: 18
Function: 0
Offset: 40h

Bit	Attr	Default	Description
15	RW	0	**ABAR Enable:** When set, the range FECX_Y000 to FECX_YFFF (FECX_YZFF where Z is treated as F) is enabled as an alternate access method to the IOxAPIC registers and these addresses are claimed by the MCH's internal I/OxAPIC regardless of the setting the MSE bit in the I/OxAPIC config space. Bits 'XY" & "Z' are defined below.
14:12	RV	0h	Reserved
11:8	RO	8h	Base Address X [Base Address bits 19:16 aka XBAD]: These bits determine the high order bits of the I/O APIC address map. When a memory address is recognized by the MCH which matches FECX_Y000-to-FECX_YFFF, the MCH will respond to the cycle and access the internal I/O APIC.
7:4	RO	9h	Base Address [Base Address bits 15:12 aka YBAD]: These bits determine the low order bits of the I/O APIC address map. When a memory address is recognized by the MCH which matches FECX_Y000-to-FECX_YFFF, the MCH will respond to the cycle and access the internal I/O APIC.
3:0	RO	0h	Base Address [Base Address bits 11:8 aka ZBAD]:These bits determine the low order bits of the I/O APIC address map, except that this field is ignored by the MCH address decode logic. When a memory address is recognized by the MCH which matches FECX_Y000-to-FECX_YFFF, the MCH will respond to the cycle and access the internal I/O APIC.

3.10.1.10 PMCAP[18]: 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: 18
Function: 0
Offset: 6Ch

Bit	Attr	Default	Description
31:27	RO	00000	PME Support
26	RO	0	D2 Support MCH does not support power management state D2.
25	RO	0	D1 Support MCH does not support power management state D1.
24:22	RO	0h	AUX Current
21	RO	0	Device Specific Initialization
20	RV	0	*Reserved.*

Device: 18 Function: 0 Offset: 6Ch			
Bit	**Attr**	**Default**	**Description**
19	RO	0	PME Clock This field is hardwired to 0h as it does not apply to IOxAPIC.
18:16	RO	011	Version This field is set to 3h to indicate that I/OxAPIC device is 1.2 compliant.
15:8	RO	00h	Next Capability Pointer This is the last capability in the chain and hence set to 0.
7:0	RO	01h	Capability ID Provides the PM capability ID assigned by PCI-SIG.

3.10.1.11 PMCSR[18]: Power Management Control and Status Register

This register provides status and control information for PM events in the PCI Express port of the MCH.

Device: 18 Function: 0 Offset: 70h			
Bit	**Attr**	**Default**	**Description**
31:24	RO	00h	Data Not relevant for MCH
23	RO	0h	Bus Power/Clock Control Enable This field is hardwired to 0h as it does not apply.
22	RO	0h	B2/B3 Support This field is hardwired to 0h as it does not apply
21:16	RV	0h	*Reserved.*
15	RO	0h	PME Status This field is hardwired to 0h as it does not apply to IOxAPIC.
14:13	RO	0h	Data Scale Not relevant for MCH
12:9	RO	0h	Dat**a Select** Not relevant for MCH
8	RO	0h	PME Enable This field is hardwired to 0h as it does not apply to IOxAPIC.

Device: 18 Function: 0 Offset: 70h			
Bit	**Attr**	**Default**	**Description**
7:4	RV	0h	*Reserved.*
3	RWO	1	NSR: No Soft Reset Indicates MCH does not reset its registers when transitioning from D3hot to D0.
2	RV	0h	*Reserved.*
1:0	RW	0h	Power State This 2-bit field is used to determine the current power state of the function and to set a new power state as well. 00: D0 01: D1 (reserved) 10: D2 (reserved) 11: D3_hot If Software tries to write 01 or 10 to this field, the power state does not change from the existing power state (which is either D0 or D3hot) and nor do these bits change value. All devices will respond to only Type 0 configuration transactions when in D3hot state (root port will not forward type 1 accesses to the downstream link) and will not respond to memory/IO transactions (i.e. D3hot state is equivalent to MSE/IOSE bits being clear) as target and will not generate any memory/IO/configuration transactions as initiator on the primary bus (messages are still allowed to pass through). For the IOxAPIC, when placed into D3hot, INTx message routing is not affected, IOCFG transactions are not affected, broadcast EOIs are still accepted, MMCFG transactions are master-aborted, and APIC messages are not delivered.

3.10.2 I/OxAPIC Registers in ESI Address Space

The IPxAPIC Control register (IOAPICCTRL) (Device 0, Function 0, Offset 46h) is documented in Section 3.8.8.29, “IOAPICCTRL: IOAPIC Control Register” on page 113.

This register provides control for the interrupt handling behavior of the IOxAPIC. This includes the Disable PCI INTx Routing to bit (0).

3.10.3 I/OxAPIC Memory Mapped Registers

I/OxAPIC has a direct memory mapped space. An index/data register pair is located within the directed memory mapped region and is used to access the redirection table entries. Table 3-53, “I/OxAPIC Indexed Registers (Redirection Table Entries)” provides the direct memory mapped registers of the I/OxAPIC. The offsets shown in the table are from the base address in either ABAR or MBAR or both. Accesses to addresses beyond 40h return all 0s.

The assignment made for the internal IOxAPIC is Bus 0, Device 18, Function 1. The same address offsets as listed in the register descriptions are used regardless of what the transaction source.

Note that while either the ABAR or MBAR can address offsets up to 0xFFF, very few registers are implemented in this space. Only aligned DWORD reads and write are allowed towards the I/OxAPIC memory space. Any other accesses will result in an illegal access error.

3.10.3.1 INDEX: Index Register

The Index Register will select which indirect register appears in the window register to be manipulated by software. Software will program this register to select the desired APIC internal register.

BAR: ABAR; MBAR Offset: 00h			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0	**Index (IDX):** Indirect register to access.

3.10.3.2 WINDOW: Window Register

This is a 32-bit register specifying the data to be read or written to the register pointed to by the index register. This register can be accessed in byte quantities.

BAR: ABAR; MBAR Offset: 10h			
Bit	**Attr**	**Default**	**Description**
31:0	RW	0	**Window (WND):** Data to be written to the indirect register on writes, and location of read data from the indirect register on reads.

3.10.3.3 EOI: End Of Interrupt Register

BAR: ABAR; MBAR Offset: 40h			
Bit	**Attr**	**Default**	**Description**
7:0	RW	0	**EOI:** The EOI register is present to provide a mechanism to efficiently convert level interrupts to edge triggered MSI interrupts. When a write is issued to this register, the I/O(x)APIC will check the lower 8 bits written to this register, and compare it with the vector field for each entry in the I/O Redirection Table. When a match is found, the Remote_IRR bit for that I/O Redirection Entry will be cleared. Note that if multiple I/O Redirection entries, for any reason, assign the same vector, each of those entries will have the Remote_IRR bit reset to '0'. This will cause the corresponding I/OxAPIC entries to resample their level interrupt inputs and if they are still asserted, cause more MSI interrupt(s) (if unmasked) which will again set the Remote_IRR bit. This bit is specified as WO in the APIC spec, but has been implemented as RW in logic.

3.10.4 I/OxAPIC Indexed Registers

Table 3-53. I/OxAPIC Indexed Registers (Redirection Table Entries)

Indexed Register	Index
APICID	00h
VERSION	01h
ARBID	02h
BCFG	03h
	...
RTL0	10h
RTH0	11h
RTL1	12h
RTH1	13h
	...
RTL23	3Eh
RTH23	3Fh
	40h
	...
	FFh

3.10.4.1 APICID: APIC ID Register

This register uniquely identifies an APIC in the system. This register is not typically used by Operating Systems, it is a legacy field that is not used for parallel delivery.

Index: 00h			
Bit	**Attr**	**Default**	**Description**
31:28	RO	0	Reserved
27:24	RW	0	**APICID:** Allows for up to 16 unique APIC IDs in the system.
23:0	RO	0	Reserved

3.10.4.2 VERSION: Version Register

This read-only register identifies the specific implementation attributes and version.

Index: 01h			
Bit	**Attr**	**Default**	**Description**
31:24	RO	0	Reserved
23:16	RO	17h	**Maximum Redirection Entries (MAX):** This is the entry number of the highest entry in the redirection table. It is equal to the number of interrupt inputs minus one. This field is hardwired to 17h to indicate 24 interrupts.
15	RO	0	IRQ Assertion Register Supported (PRQ): This bit is set to 0 to indicate that this version of the I/OxAPIC does not implement the IRQ Assertion register and does not allow PCI devices to write to it to cause interrupts.
14:8	RO	0	Reserved
7:0	RO	20h	**Version (VS):** This identifies the implementation version. This field is hardwired to 20h indicate this is an I/OxAPIC.

3.10.4.3 ARBID: Arbitration ID Register

This register is considered a don't care for parallel message delivery. It tracks the APICID register for compatibility reasons.

Index: 02h			
Bit	**Attr**	**Default**	**Description**
31:28	RO	0	Reserved
27:24	RO	0	**Arbitration ID:** Just tracks the APICID register.
23:0	RO	0	Reserved

3.10.4.4 BCFG: Boot Configuration Register

This register is also known as the local configuration register.

Index: 03h			
Bit	**Attr**	**Default**	**Description**
31:1	RO	0	Reserved
0	RW	1	**Boot Configuration:** This bit has a default of 1 to indicate FSB delivery mode. A written value of 0 has no effect on the behavior of the logic. This bit is left as RW for software compatibility reasons.

3.10.4.5 RTL[0:23]: Redirection Table Entries Low DWORD

The information in this register along with Redirection Table High DWORD register is used to construct the MSI interrupt. There is one of these pairs of registers for every interrupt. The first interrupt has the redirection registers at offset 10h. The second interrupt at 12h, third at 14h, etc. until the final interrupt (interrupt 23) at 3Eh.

Index: 10h, 12h, 14h, 16h, 18h, 1Ah, 1Ch, 1Eh, 20h, 22h, 24h, 26h, 28h, 2Ah, 2Ch, 2Eh, 30h, 32h, 34h, 36h, 38h, 3Ah, 3Ch, 3Eh			
Bit	**Attr**	**Default**	**Description**
31:18	RO	0	Reserved
17	RW	0	Disable Flushing: This bit has no meaning in the MCH. This bit is R/W for software compatibility reasons only
16	RW	1	**Mask (MSK):** When cleared, an edge assertion or level (depending on bit 15 in this register) on the corresponding interrupt input results in delivery of an MSI interrupt using the contents of the corresponding redirection table high/low entry. When set, an edge or level on the corresponding interrupt input does not cause MSI Interrupts and no MSI interrupts are held pending as well (i.e. if an edge interrupt asserted when the mask bit is set, no MSI interrupt is sent and the hardware does not remember the event to cause an MSI later when the mask is cleared). When set, assertion/deassertion of the corresponding interrupt input causes Assert/Deassert_INTx messages to be sent to the legacy I/O Hub, provided the 'Disable PCI INTx Routing to I/O Hub' bit is clear. If the latter is set, Assert/Deassert_INTx messages are not sent to the legacy I/O Hub. When mask bit goes from 1 to 0 for an entry and the entry is programmed for level input, the input is sampled and if asserted, an MSI is sent. Also, if an Assert_INTx message was previously sent to the legacy I/O Hub/internal-coalescing logic on behalf of the entry, when the mask bit is clear, then a Deassert_INTx event is scheduled on behalf of the entry (whether this event results in a Deassert_INTx message to the legacy I/O Hub depends on whether there were other outstanding Deassert_INTx messages from other sources). (Within this context, the term "I/O Hub" refers to the I/O Controller Hub which resides within the Intel 631xESB/632xESB I/O Controller Hub.)
15	RW	0	Trigger Mode (TM): This field indicates the type of signal on the interrupt input that triggers an interrupt. 0 indicates edge sensitive, 1 indicates level sensitive.

Index:	10h, 12h, 14h, 16h, 18h, 1Ah, 1Ch, 1Eh, 20h, 22h, 24h, 26h, 28h, 2Ah, 2Ch, 2Eh, 30h, 32h, 34h, 36h, 38h, 3Ah, 3Ch, 3Eh		
Bit	**Attr**	**Default**	**Description**
14	RO	0	Remote IRR (RIRR): This bit is used for level triggered interrupts; its meaning is undefined for edge triggered interrupts. For level triggered interrupts, this bit is set when an MSI interrupt has been issued by the I/OxAPIC into the system fabric (noting that if BME bit is clear or when the mask bit is set, no new MSI interrupts cannot be generated and this bit cannot transition from 0 to 1 in those conditions). It is reset (if set) when an EOI message is received from a local APIC with the appropriate vector number, at which time the level interrupt input corresponding to the entry is resampled causing one more MSI interrupt (if other enable bits are set) and causing this bit to be set again.
13	RW	0	Interrupt Input Pin Polarity (IP): 0=active high; 1=active low. Strictly, speaking this bit has no meaning in MCH since the Assert/Deassert_INTx messages are level in-sensitive. Most OS'es today support only active low interrupt inputs for PCI devices. Given that, the OS is expected to program a 1 into this register. (Note: The "internal" virtual wire signals in the MCH need to be active low; i.e. 0=asserted and 1=deasserted.)
12	RO	0	Delivery Status: 0 - No activity for this interrupt. 1 - Edge: Event has occurred but interrupt message has not be delivered. Once delivered this bit will clear. 1- Level: Input is active. Cleared when input is cleared. Stays active regardless of delivered or not.
11	RW	0	**Destination Mode (DSTM):** 0 – Physical 1 – Logical
10:8	RW	0	**Delivery Mode (DELM):** This field specifies how the APICs listed in the destination field should act upon reception of the interrupt. Certain Delivery Modes will only operate as intended when used in conjunction with a specific trigger mode. The encodings are: 000 – Fixed: Trigger Mode can be edge or level. 001 – Lowest Priority: Trigger Mode can be edge or level. 010 – SMI/PMI 011 – Reserved 100 – NMI 101 – INIT 110 – Reserved 111 – ExtINT NOTE: Fixed or Lowest Priority can be either edge or level, and the trigger mode must be examined. For all other legal values, ignore trigger mode, and assume edge.
7:0	RW	0	Vector (VCT): This field contains the interrupt vector for this interrupt

3.10.4.6 RTH[0:23]: Redirection Table Entries High DWORD

Index:	11h, 13h, 15h, 17h, 19h, 1Bh, 1Dh, 1Fh, 21h, 23h, 25h, 27h, 29h, 2Bh, 2Dh, 2Fh, 31h, 33h, 35h, 37h, 39h, 3Bh, 3Dh, 3Fh		
Bit	**Attr**	**Default**	**Description**
31:24	RW	00h	**Destination ID (DID):** They are bits [19:12] of the MSI address.
23:16	RW	00h	**Extended Destination ID (EDID):** These bits become bits [11:4] of the MSI address. (Not required for IA32 support.)
15:00	RO	0000h	Reserved

3.11 Intel® VT-d Memory Mapped Registers

This chapter describes both the PCI Configuration space and CSRCFG space registers.

3.11.1 Intel VT-d Memory Mapped Register Map

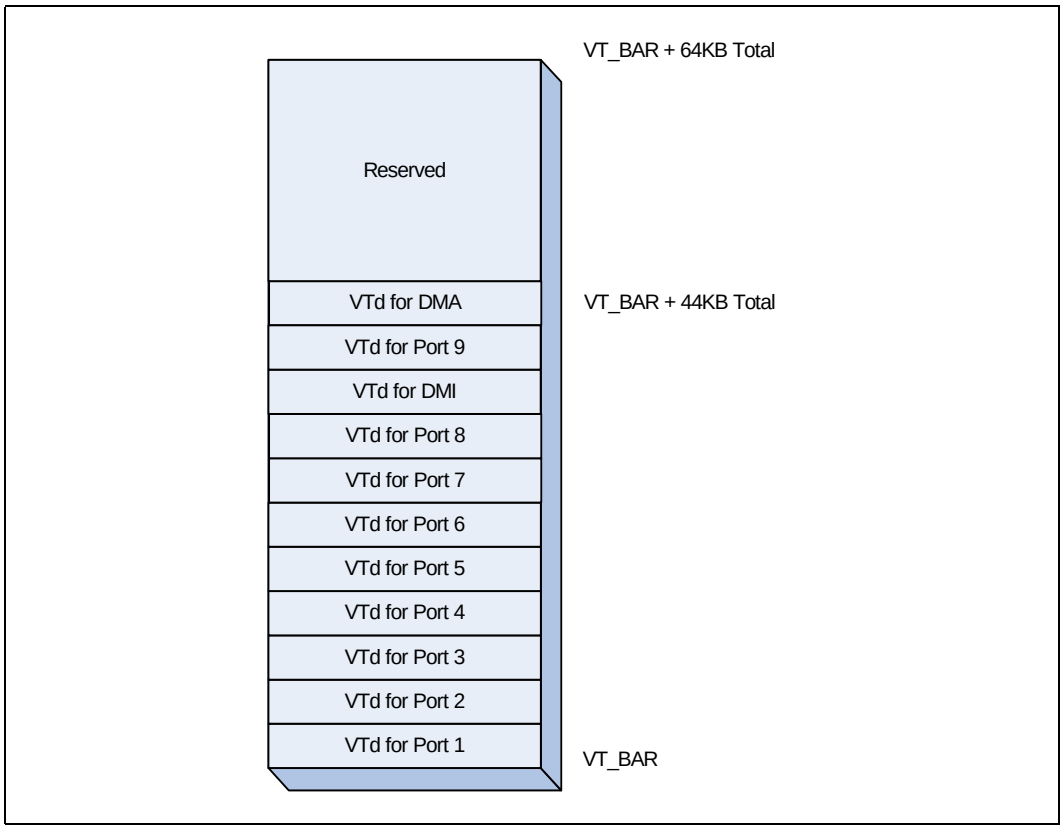
Table 3-54. Intel VT-d Memory Mapped Registers

VTD_VERSION				00h					80h
EXT_VTD_CAP				04h					84h
VTD_CAP				08h					88h
				0Ch					8Ch
GLOBALCMD				10h					90h
GLOBALSTS				14h					94h
ROOTENTRYADD				18h					98h
				1Ch					9Ch
CONTEXTCMD				20h					A0h
				24h					A4h
CTXCMD				28h					A8h
				2Ch					ACh
				30h					B0h
FLTSTS				34h					B4h
FLTEVTCTRL				38h					B8h
FLTEVTDATA				3Ch					BCh
FLTEVTADDR				40h	FLTRECL				C0h
				44h					C4h
				48h	FLTRECU				C8h
				4Ch					CCh
				50h					D0h
				54h					D4h
FLTLOG				58h					D8h
				5Ch					DCh
				60h	IOTLBINV				E0h
PROT_MEM_EN				64h					E4h
PROT_LOW_MEM_BASE				68h	INVADD				E8h
PROT_LOW_MEM_LIMIT				6Ch					ECh
PROT_HIGH_MEM_BASE				70h	VTDCONFIG				F0h
				74h					F4h
PROT_HIGH_MEM_LIMIT				78h					F8h
				7Ch					FCh

3.11.2 Intel VT-d Memory Mapped Registers

The Intel® VT-d registers described in this section are required individually for each Intel® QuickData Technology Device remap engine. There is one set of these registers exposed per x4 root port, one for the ESI port and one for the Intel® QuickData Technology Device. These registers are all addressed using aligned DWORD or aligned QWORD accesses, starting from the 4 k aligned base address in the VTCSRBASE register. Any combination of BEs is allowed within a DWORD or QWORD access. The Intel® QuickData Technology Device remap engine registers corresponding to the root port, Port 1, occupy the first 4 K of offset starting from the base address. The Intel® QuickData Technology Device remap engine registers corresponding to the root port, Port 2, occupy the second 4 K of offset starting from the base address and so on. The ESI port's remap registers occupy the 9th 4 K of offset starting from the base. Figure 3-6 shows the base address of the various remap engines as offsets from the VT_BAR address.

Figure 3-6. Base Address of Intel VT-d Remap Engines



3.11.2.1 VTD_VERSION: Version Number Register

Offset: 00h			
Bit	**Attr**	**Default**	**Description**
31:8	RV	0h	Reserved
7:4	RO	1h	Major Revision
3:0	RO	0h	Minor Revision

3.11.2.2 VTD_CAP: VT-d Chipset Capabilities Register

Offset: 08h			
Bit	**Attr**	**Default**	**Description**
63:56	RV	0	Reserved
55	RO	0	**Intel® QuickData Technology Device Read Draining:** MCH does not support any hardware based draining
54	RO	if (VTd for Intel® QuickData Technology Device) {1} else {0} endif	**Intel® QuickData Technology Device Write Draining:** MCH does not support any hardware based draining for root ports but it supports hardware based write draining for Intel® QuickData Technology device and Intel® QuickData Technology Intel VT-d engine.
53:48	RO	09h	Reserved
47:40	RO	01h	Number of Fault Recording Registers: Only one register supported for MCH.
39	RWO	0	**PSI: Page Selective Invalidation Support** This field indicates support for domain/device specific page selective invalidation. MCH support domain specific PSI
38	RO	0	Reserved
37:34	RO	0	**SPS: Super Page support** This field indicates super page size supported by hardware. SG does not support super pages.
33:24	RO	0Ch	Fault Recording Register Offset: Fault registers are at offset 0xC0h
23	RO	0	**Spatial Separation:** N/A for MCH
22	RO	0	Reserved
21:16	RO	30h	**MGAW**: This register is set by MCH based on the setting of the VTCTRL. GPA_LIMIT register. Modification of the VTCTRL.GPA_LIMIT field will change this field's default value. The default value of GPA_Limit is 48 bit of GPA Width.
15:13	RV	0h	Reserved
14:13	RO	0h	**SPS**: N/A for MCH since it does not support super pages
12:8	RO	4h	**SAGAW:** Supported Adjusted Guest Address Width This 5-bit field indicates the supported adjusted guest address width (which in turn represents the level of page-table walks) supported by the hardware implementation. 0: 30-bit AGAW (2-level page table) 1: 39-bit AGAW (3-level page table) 2: 48-bit AGAW (4-level page table) 3: 57-bit AGAW (5-level page table) 4: 64-bit AGAW (6-level page table) MCH supports only 4 level page walks
7	RWO	0	**TCM: Cache Mode** This field indicates if the hardware caches not present or erroneous entries in the context-cache and IOTLB. SG does not cache invalid pages.
6	RO	1	**No PHMR Support**: MCH does not supports protected high memory range
5	RO	1	**No PLMR Support:** MCH does not supports protected low memory range
4	RO	0	**RWBF**: N/A for MCH
3	RO	0	**Advanced Fault Logging**: MCH does not support advanced fault logging

Offset: 08h			
Bit	**Attr**	**Default**	**Description**
2:0	RO	010b	**Number of Domains Supported:** MCH supports 256 domains with 8 bit domain ID

3.11.2.3 EXT_VTD_CAP: Extended VT-d Capability Register

Offset: 10h			
Bit	**Attr**	**Default**	**Description**
63:32	RV	0	Reserved
31:24	RO	0	Number of Invalidation Units: MCH supports only one and so this is set to 0
23:18	RV	0	Reserved
17:8	RO	Eh	Invalidation Unit Offset: MCH's invalidation unit resides at offset
7:1	RV	0	Reserved
0	RO	1	Root Entry Table must be in cacheable region

3.11.2.4 GLBCMD: Global Command Register

Offset: 18h			
Bit	**Attr**	**Default**	**Description**
31	RW	0	**Translation Enable:** Software writes to this field to request hardware to enable/disable Intel® QuickData Technology Device-remapping hardware. 0: Disable Intel® QuickData Technology Device-remapping hardware 1: Enable Intel® QuickData Technology Device-remapping hardware Hardware reports the status of the translation enable operation through the TES field in the Global Status register. Before enabling (or re-enabling) Intel® QuickData Technology Device-remapping hardware through this field, software must: • Setup the Intel® QuickData Technology Device-remapping structures in memory • Flush the write buffers (through WBF field), if write buffer flushing is reported as required. • Set the root-entry table pointer in hardware (through SRTP field). • Perform global invalidation of the context-cache and global invalidation of IOTLB • If advanced fault logging supported, setup fault log pointer (through SFL field) and enable advanced fault logging (through EAFL field). There may be active Intel® QuickData Technology Device requests in the platform when software updates this field. Hardware must enable or disable remapping logic only at deterministic transaction boundaries, so that any in-flight transaction is either subject to remapping or not at all.

Offset: 18h			
Bit	**Attr**	**Default**	**Description**
30	RW	0	**Set Root Table Pointer:** Software sets this field to set/update the root-entry table pointer used by hardware. The root-entry table pointer is specified through the Root-entry Table Address register. Hardware reports the status of the root table pointer set operation through the RTPS field in the Global Status register. The root table pointer set operation must be performed before enabling or re-enabling (after disabling) Intel® QuickData Technology Device remapping hardware. After a root table pointer set operation, software must globally invalidate the context cache followed by global invalidate of IOTLB. This is required to ensure hardware uses only the remapping structures referenced by the new root table pointer, and not any stale cached entries. While Intel® QuickData Technology Device-remapping hardware is active, software may update the root table pointer through this field. However, to ensure valid in-flight Intel® QuickData Technology Device requests are deterministically remapped, software must ensure that the structures referenced by the new root table pointer are programmed to provide the same remapping results as the structures referenced by the previous root table pointer. Clearing this bit has no effect.
29	RO	0	**Set Fault Log Pointer:** N/A to MCH
28	RO	0	**Enable Advanced Fault Logging:** N/A to MCH
27	RO	0	**Write Buffer Flush:** N/A to MCH
26:0	RO	0	Reserved

3.11.2.5 GLBSTS: Global Status Register

Offset: 1Ch			
Bit	**Attr**	**Default**	**Description**
31	RO	0	**Translation Enable Status:** When set, indicates that translation hardware is enabled and when clear indicates the translation hardware is not enabled.
30	RO	0	**Set Root Table Pointer Status:** This field indicates the status of the root- table pointer in hardware. This field is cleared by hardware when software sets the SRTP field in the Global Command register. This field is set by hardware when hardware finishes the set root-table pointer operation (by performing an implicit global invalidation of the context-cache and IOTLB, and setting/updating the root-table pointer in hardware with the value provided in the Root-Entry Table Address register).
29	RO	0	**Set Fault Log Pointer Status:** N/A to MCH
28	RO	0	**Advanced Fault Logging Status:** N/A to MCH
27	RO	0	**Write Buffer Flush Status:** N/A to MCH
26:0	RV	0	Reserved

3.11.2.6 ROOTENTRYADD: Root Entry Table Address Register

Offset: 20h			
Bit	**Attr**	**Default**	**Description**
63:12	RW	0	**Root Entry Table Base Address**: 4 K aligned base address for the root entry table. MCH does not utilize bits 63 : 43 and checks for them to be 0. Software specifies the base address of the root-entry table through this register, and enables it in hardware through the SRTP field in the Global Command register. Reads of this register returns value that was last programmed to it.
11:0	RV	0	Reserved

3.11.2.7 CTXCMD: Context Command Register

Offset: 28h			
Bit	**Attr**	**Default**	**Description**
63	RW	0	Invalidate Context Entry Cache (ICC): Software requests invalidation of context-cache by setting this field. Software must also set the requested invalidation granularity by programming the CIRG field. Software must read back and check the ICC field to be clear to confirm the invalidation is complete. Software must not update this register when this field is set. Hardware clears the ICC field to indicate the invalidation request is complete. Hardware also indicates the granularity at which the invalidation operation was performed through the CAIG field. Software must not submit another invalidation request through this register while the ICC field is set. Software must submit a context cache invalidation request through this field only when there are no invalidation requests pending at this Intel® QuickData Technology Device-remapping hardware unit. Since information from the context-cache may be used by hardware to tag IOTLB entries, software must perform domain-selective (or global) invalidation of IOTLB after the context cache invalidation has completed.
62:61	RW	0	Context Invalidation Request Granularity (CIRG): When requesting hardware to invalidate the context-entry cache (by setting the ICC field), software writes the requested invalidation granularity through this field. Following are the encoding for the 2-bit IRG field. 00: Reserved. Hardware ignores the invalidation request and reports invalidation complete by clearing the ICC field and reporting 00 in the CAIG field. 01: Global Invalidation request. MCH supports this. 10: Domain-selective invalidation request. The target domain-id must be specified in the DID field. MCH supports this. 11: Device-selective invalidation request. The target SID must be specified in the SID field, and the domain-id (programmed in the context-entry for this device) must be provided in the DID field. MCH does not support this and alias this request to a domain-selective invalidation request. Depending on the invalidation granularities supported by a hardware implementation, an invalidation request may be processed by performing invalidation at a coarser granularity. Hardware indicates completion of the invalidation request by clearing the ICC field. At this time, hardware also indicates the granularity at which the actual invalidation was performed through the CAIG field.
60:59	RO	0	Context Actual Invalidation Granularity (CAIG): Hardware reports the granularity at which an invalidation request was processed through the CAIG field at the time of reporting invalidation completion (by clearing the ICC field). The following are the encoding for the 2-bit CAIG field. 00: Reserved. This is the value on reset. 01: Global Invalidation performed. MCH sets this in response to a global invalidation request. 10: Domain-selective invalidation performed using the domain-id that was specified by software in the DID field. MCH sets this in response to a domain-selective or device-selective invalidation request. 11: Device-selective invalidation performed. MCH does not set this value at all.
58:34	RV	0	Reserved
33:32	RO	0	Function Mask: Since MCH does not perform any device selective invalidation, this field is a don't care
31:16	RO	0	Source ID: MCH ignores this field
15:0	RW	0	Domain ID: Indicates the id of the domain whose context-entries needs to be selectively invalidated. S/W needs to program this for both domain and device selective invalidates. MCH ignores bits 15:8 since it supports only a 8 bit Domain ID.

3.11.2.8 FLTSTS: Fault Status Register

Offset: 34h			
Bit	**Attr**	**Default**	**Description**
31:16	RV	0	Reserved
15:8	ROS	0	**Fault Record Index:** This field is valid only when the Primary Fault Pending field is set. This field indicates the index (from base) of the fault recording register to which the first pending fault was recorded when the Primary Fault pending field was set by hardware.
7:2	RV	0	Reserved
1	ROS	0	**Primary Fault Pending**: This field indicates if there are one or more pending faults logged in the fault recording registers. Hardware computes this field as the logical OR of Fault (F) fields across all the fault recording registers of this Intel® QuickData Technology Device-remap hardware unit. 0: No pending faults in any of the fault recording registers 1: One or more fault recording registers has pending faults. The fault recording index field is updated by hardware whenever this field is set by hardware. Also, depending on the programming of fault event control register, a fault event is generated when hardware sets this field.
0	RW1CS	0	**Primary Fault Overflow:** Hardware sets this bit to indicate overflow of fault recording registers

3.11.2.9 FLTEVTCTRL: Fault Event Control Register

Offset: 38h			
Bit	**Attr**	**Default**	**Description**
31	RW	1	Interrupt Message Mask: 1: Hardware is prohibited from issuing interrupt message requests. 0: Software has cleared this bit to indicate interrupt service is available. When a faulting condition is detected, hardware may issue a interrupt request (using the fault event data and fault event address register values) depending on the state of the interrupt mask and interrupt pending bits.
30	RO	0	**Interrupt Pending:** Hardware sets the IP field whenever it detects an interrupt condition. Interrupt condition is defined as when an interrupt condition occurs when hardware records a fault through one of the Fault Recording registers and sets the PPF field in Fault Status register. If the PPF field was already set at the time of recording a fault, it is not treated as a new interrupt condition. The IP field is kept set by hardware while the interrupt message is held pending. The interrupt message could be held pending due to interrupt mask (IM field) being set, or due to other transient hardware conditions. The IP field is cleared by hardware as soon as the interrupt message pending condition is serviced. This could be due to either (a) Hardware issuing the interrupt message due to either change in the transient hardware condition that caused interrupt message to be held pending or due to software clearing the IM field. (b) Software servicing the interrupting condition through clearing the Fault (F) field in all the Fault Recording registers with faults, causing the PPF field in Fault Status register to be evaluated as clear.
29:0	RV	0	Reserved

3.11.2.10 FLTEVTDATA: Fault Event Data Register

Offset: 3Ch			
Bit	**Attr**	**Default**	**Description**
31:16	RV	0	Reserved
15:0	RW	0	Interrupt Data

3.11.2.11 FLTEVTADDR: Fault Event Address Register

Offset: 40h			
Bit	**Attr**	**Default**	**Description**
31:20	RO	FEEh	Interrupt Address: The interrupt address is interpreted as the address of any other interrupt from a PCI Express port. Software is not allowed to change this field
19:2	RW	0	**Interrupt Address:** The interrupt address is interpreted as the address of any other interrupt from a PCI Express port.
1:0	RV	0	Reserved

3.11.2.12 FLTEVTUPADDR: Fault Event Upper Address Register

Offset: 44h			
Bit	**Attr**	**Default**	**Description**
31:0	ROS	0	Only 32 bit address supported and this field is RO 0.

3.11.2.13 FLTLOG: Fault Log Register

Offset: 58h			
Bit	**Attr**	**Default**	**Description**
63:9	RO	0	N/A for MCH since MCH does not support advanced fault logging
8:2	RV	0	Reserved
1:0	RO	0	N/A for MCH since MCH does not support advanced fault logging

3.11.2.14 PMEN : Protected Memory Enable Register

Offset: 64h			
Bit	**Attr**	**Default**	**Description**
31	RW	0	Enable Protected Memory as defined by the PROT_LOW(HIGH)_BASE and PROT_LOW(HIGH)_LIMIT registers Note: The MCH does not support high and low protected memory regions.
30:1	RV	0	Reserved
0	RO	0	Protected Region Status: This bit is set by MCH whenever it has completed enabling the protected memory region per the rules stated in the Intel VT-d spec

3.11.2.15 PROT_LOW_MEM_BASE : Protected Memory Low Base Register

Offset: 68h			
Bit	**Attr**	**Default**	**Description**
31:24	RW	0	**Protected Low Memory Base:** 16 MB aligned base address of the low protected dram region Note: The MCH does not support high and low protected memory regions.
23:0	RV	0	Reserved

3.11.2.16 PROT_LOW_MEM_LIMIT : Protected Memory Low Limit Register

Offset: 6Ch			
Bit	**Attr**	**Default**	**Description**
31:24	RW	0	**Protected Low Memory Limit:** 16MB aligned base address of the low protected dram region Note: The MCH does not support high and low protected memory regions.
23:0	RV	0	Reserved

3.11.2.17 PROT_HIGH_MEM_BASE : Protected Memory High Base Register

Offset: 70h			
Bit	**Attr**	**Default**	**Description**
63:24	RW	0	16 MB aligned base address of the high protected dram region Note: The MCH does not support high and low protected memory regions.
23:0	RV	0	Reserved

3.11.2.18 PROT_HIGH_MEM_LIMIT : Protected Memory Limit Base Register

Offset: 78h			
Bit	**Attr**	**Default**	**Description**
63:24	RW	0	16 MB aligned limit address of the high protected dram region Note: The MCH does not support high and low protected memory regions.
23:0	RV	0	Reserved

3.11.2.19 FLTRECL: Fault Record Lower Register

Offset: C0h			
Bit	**Attr**	**Default**	**Description**
63:12	ROS	0	**GPA:** 4 k aligned GPA for the faulting transaction. Valid only when F field is set
11:0	RV	0	Reserved

3.11.2.20 FLTRECU: Fault Record Upper Register

Offset: C8h			
Bit	**Attr**	**Default**	**Description**
63	RW1CS	0	Fault (F): Hardware sets this field to indicate a fault is logged in this fault recording register. The F field is set by hardware after the details of the fault is recorded in the PADDR, SID, FR and T fields. When this field is set, hardware may collapse additional faults from the same requestor (SID). Software writes the value read from this field to clear it.
62	ROS	0	Type: Type of the first faulted Intel® QuickData Technology Device request 0: Intel® QuickData Technology Device write 1: Intel® QuickData Technology Device read request This field is only valid when Fault (F) bit is set.
61:40	RV	0	Reserved
39:32	ROS	0	Fault Reason: Reason for the first translation fault. See Intel VT-d spec for details. This field is only valid when Fault bit is set.
31:16	RV	0	Reserved
15:0	ROS	0	**Source Identifier**: Requester ID of the Intel® QuickData Technology Device request that faulted. Valid only when F bit is set

3.11.2.21 IOTLBINV: IOTLB Invalidate Register

Offset: E8h			
Bit	**Attr**	**Default**	**Description**
63	RW	0	Invalidate IOTLB cache (IVT): Software requests IOTLB invalidation by setting this field. Software must also set the requested invalidation granularity by programming the IIRG field. Hardware clears the IVT field to indicate the invalidation request is complete. Hardware also indicates the granularity at which the invalidation operation was performed through the IAIG field. Software must read back and check the IVT field to be clear to confirm the invalidation is complete. When IVT field is set, software must not update the contents of this register (and Invalidate Address register, i if it is being used), nor submit new IOTLB invalidation requests.
62:60	RW	0	IOTLB Invalidation Request Granularity (IIRG): When requesting hardware to invalidate the I/OTLB (by setting the IVT field), software writes the requested invalidation granularity through this IIRG field. Following are the encoding for the 3-bit IIRG field. 000: Reserved. Hardware ignores the invalidation request and reports invalidation complete by clearing the IVT field and reporting 00 in the AIG field. An invalid programming error is reported through the error-status register. 001: Global Invalidation request. MCH supports this. 010: Domain-selective invalidation request. The target domain-id must be specified in the DID field. MCH supports this 011: Reserved 100: Device-page-selective invalidation request. The target address, mask and invalidation hint must be specified in the Invalidate Address register, the domain-id must be provided in the DID field, and the device requestor-id must be provided in SID field. MCH aliases this to "011" i.e. it performs a domain-page-selective invalidation on this request as well. 101-111 - Reserved. MCH ignores the invalidation request and completes the invalidation the invalidation request by clearing the IVT field and reporting 000 in the IAIG field. At this time, hardware also indicates the granularity at which the actual invalidation was performed through the IAIG field.

Offset: E8h			
Bit	**Attr**	**Default**	**Description**
59:57	RO	0	IOTLB Actual Invalidation Granularity (IAIG): Hardware reports the granularity at which an invalidation request was proceed through the AIG field at the time of reporting invalidation completion (by clearing the IVT field). The following are the encoding for the 3-bit IAIG field. 000: Reserved. This is the value on reset. 001: Global Invalidation performed. MCH sets this in response to a global IOTLB invalidation request. 010: Domain-selective invalidation performed using the domain-id that was specified by software in the DID field. MCH sets this in response to a domain-specific or page-specific IOTLB invalidation request. 011: Domain-page selective invalidation performed. MCH sets this in response to domain/device-page selective invalidation requests. 100-111: Reserved
56:50	RV	0	Reserved
49	RO	0	Drain Reads: MCH does not support this feature
48	Reg ? 10 : RW RO	Reg ? 10 : 1 0	**Drain Writes:** This bit allows software Intel® QuickData Technology Device to indicate to the hardware if write requests need to be drained on an IOTLB invalidation. Drain writes on an IOTLB invalidation is only supported for the Intel® QuickData Technology Device.
47:32	RW	0	**Domain ID**: Domain to be invalidated and is programmed by software for both page and domain selective invalidation requests. MCH ignores the bits 47:40 since it supports only an 8 bit Domain ID.
31:0	RV	0	Reserved

3.11.2.22 VTDCONFIG: VTD Configuration Register

Offset: F0h			
Bit	**Attr**	**Default**	**Description**
31:8	RO	0h	Reserved
7:4	RW	Fh	LRU_TIMER: VTD Least Recently Used Timer This field sets the number of 64 core clock ticks before a valid cache entry is demoting to a lower LRU priority bucket. There are 4 priority buckets.
3:1	RO	0h	Reserved
0	RO	0h	VTD_TLB_MODE: VTD MODE This bits sets the Mode of operations for the VTD Engine in MCH. MCH will only operate in Mode 0.

§

4 System Address Map

This chapter describes the system address maps in memory space, I/O space and PCI configuration space.

4.1 Memory Map

The Intel® 5400 chipset platform supports 38 bits of physical memory for maximum of 38-bit system address with up to 128 GB of physical memory support.

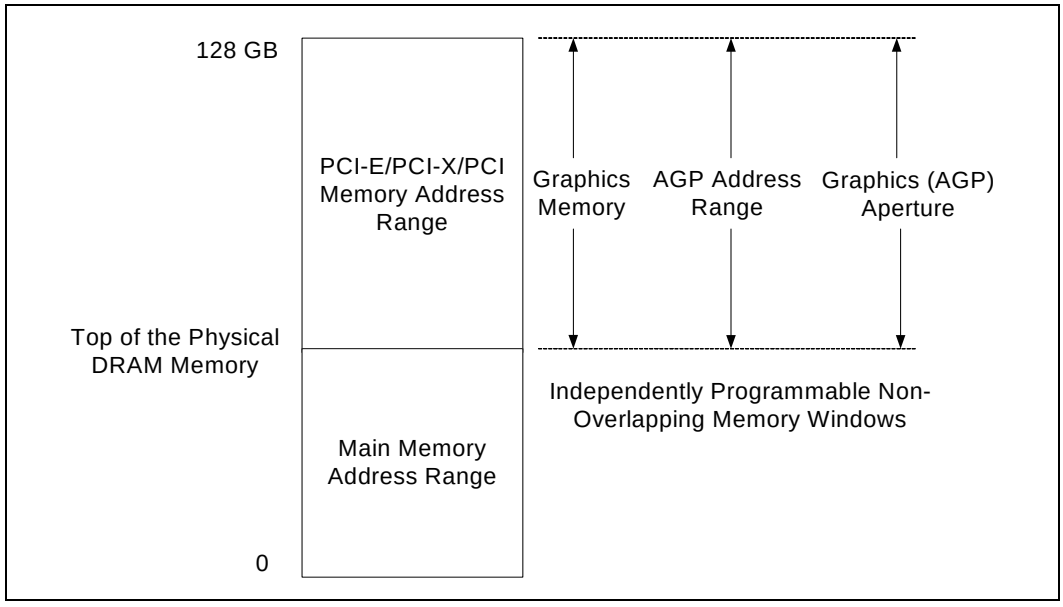
Other address spaces supported by the Intel® 5400 chipset are:

- 38-bit local address supported over the FB-DIMM channels for physical memory space.
- 32 and 64 bit address bit formats supported for PCI Express interfaces.

The chipset treats accesses to various address ranges in different ways. There are fixed ranges like the compatibility region below 1 MB, and variable ranges like the memory mapped I/O range. The locations of these ranges in the memory map are illustrated in Figure 4-2.

The Intel® 5400 chipset memory map includes a number of programmable ranges. All of these ranges must be unique and non-overlapping as shown in Figure 4-1. There are no hardware interlocks to prevent problems in the case of overlapping ranges. Accesses to overlapped ranges may produce indeterminate results. For example, setting HECBASE to all zeros will overlap the MMCFG region and the compatibility region resulting in unpredictable results.

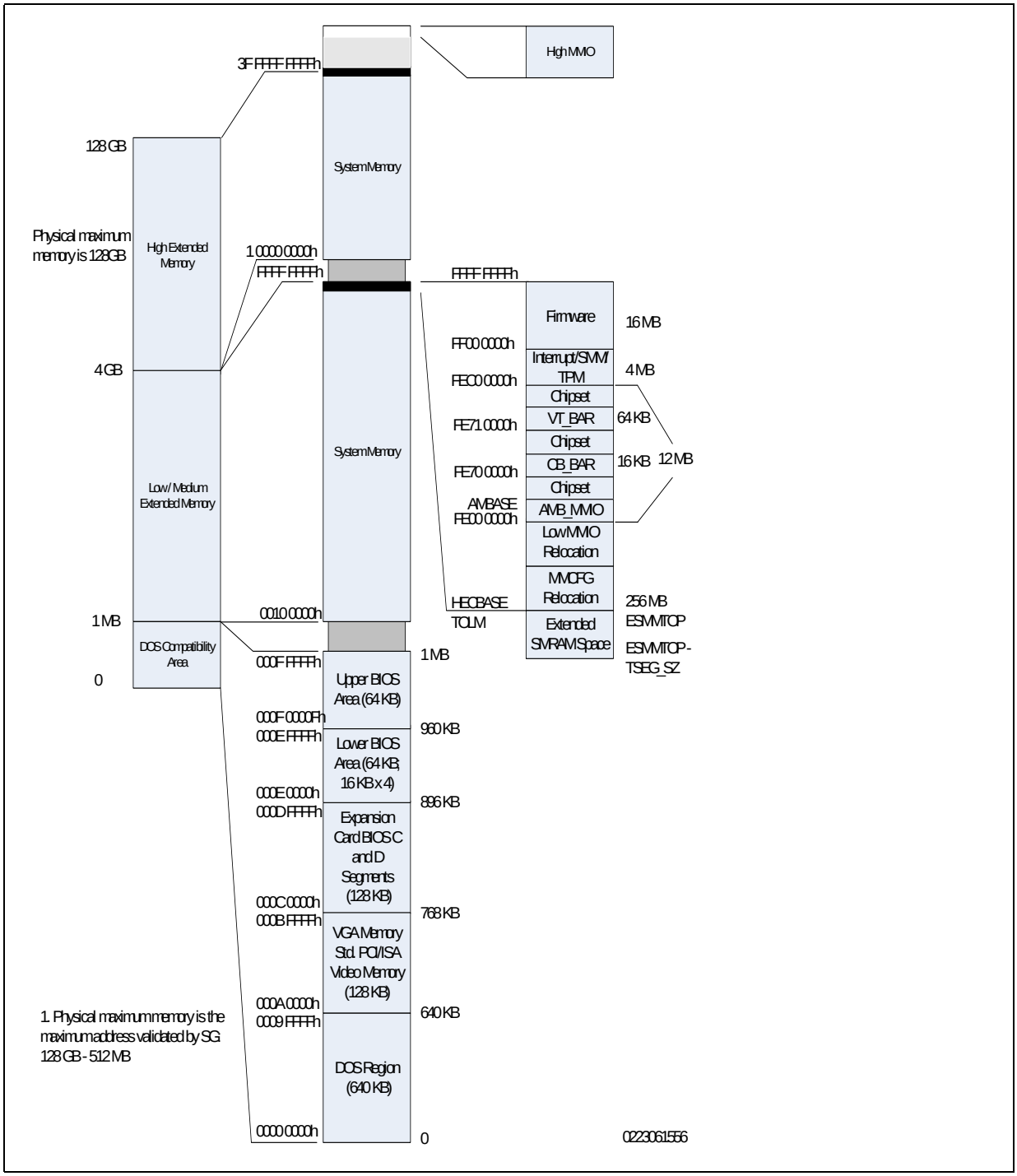
Figure 4-1. System Memory Address Map



4.1.1 32/64-bit addressing

For inbound and outbound writes and reads, the Intel® 5400 chipset 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 address format. It is the responsibility of the software to ensure that the relevant bits are programmed for 64-bits based on the OS limits.

Figure 4-2. Detailed Memory System Address Map



4.2 Compatibility Area

This is the range from 0 - 1 MB (0 0000h to F FFFFh). Requests to the compatibility area are directed to main memory, the Compatibility Bus (ESI), or the VGA device. Any physical DRAM addresses that would be addressed by requests in this region that are mapped to the Compatibility Bus (ESI) and are not recovered.

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 or VGA or other non-DRAM areas.

Addresses below 1 M that are mapped to memory are accessible by the processors and by any I/O bus. The address range below 1 M is divided into five address regions. These regions are:

- 0 - 640 KB MS-DOS Area.
- 640 – 768 KB Video Buffer Area.
- 768 – 896 KB in 16-KB sections (total of eight sections) - Expansion Card BIOS, Segments C and D.
- 896 - 960 KB in 16-KB sections (total of four sections) - Lower Extended System BIOS, Segment E.
- 960 KB–1 MB memory (BIOS Area) - Upper System BIOS, Segment F.

There are fifteen memory segments in the compatibility area. Thirteen of the memory ranges can be enabled or disabled independently for both read and write cycles.

Table 4-1. Memory Segments and Their Attributes

Memory Segments	Attributes	Comments
000000h–09FFFFh	Fixed: mapped to main DRAM	0 to 640 KB – DOS Region
0A0000h–0BFFFFh	Mapped to ESI, x16 graphics port	Video Buffer (physical DRAM configurable as SMM space)
0C0000h–0C3FFFh	WE RE	Add-on BIOS
0C4000h–0C7FFFh	WE RE	Add-on BIOS
0C8000h–0CBFFFh	WE RE	Add-on BIOS
0CC000h–0CFFFFh	WE RE	Add-on BIOS
0D0000h–0D3FFFh	WE RE	Add-on BIOS
0D4000h–0D7FFFh	WE RE	Add-on BIOS
0D8000h–0DBFFFh	WE RE	Add-on BIOS
0DC000h–0DFFFFh	WE RE	Add-on BIOS
0E0000h–0E3FFFh	WE RE	BIOS Extension
0E4000h–0E7FFFh	WE RE	BIOS Extension
0E8000h–0EBFFFh	WE RE	BIOS Extension
0EC000h–0EFFFFh	WE RE	BIOS Extension
0F0000h–0FFFFFh	WE RE	BIOS Area

4.2.1 MS-DOS Area (0 0000h–9 FFFFh)

The MS-DOS area is 640 KB in size and is mapped to main memory controlled by the MCH.

4.2.2 Legacy VGA Ranges (A 0000h–B FFFFh)

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 port, or optionally it can be mapped to main memory (it must be mapped to SMM space). Mapping of this region is controlled by the VGA steering bits. At power on this space is mapped to the ESI port.

Priority for VGA mapping is constant in that the MCH consistently decodes internally mapped devices first. The MCH positively decodes internally mapped devices. 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 Intel® 5400 chipset MCH.BCTRL configuration registers associated with the PCI Express port, then transactions in this space are sent to that PCI Express port.

The VGAEN bit can only be set in one and only one of the Intel® 5400 chipset MCH.BCTRL registers. If any VGAEN bits are set, all the ISAEN bits must be set. If the VGAEN bit of a PCI Express port x in the Intel® 5400 chipset MCH is set and BCTRL[x].VGA16bdecode is set to zero, then ISAEN bits of all peer PCI Express ports with valid I/O range (PCICMD.IOAE = 1, IOLIMIT >= IOBASE) in the MCH must be set by software. Otherwise, it is a programming error due to the resulting routing conflict.

If the VGAEN bit of a PCI Express port x in the MCH is set, and BCTRL[x].VGA16bdecode is set to one, and if there is another PCI Express port y (x != y) with valid I/O range including the lowest 4 K I/O addresses (PCICMD[y].IOAE = 1, IOLIMIT[y] >= IOBASE[y] = 0000h), BCTRL[y].ISAEN bit must be set to one by software. Otherwise, it is a programming error.

This region is non-cacheable.

Compatible SMRAM Address Range (A 0000h–B FFFFh)

The legacy VGA range may also be used for mapping SMM space. The SMM range (128 KB) 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_N signal determines where accesses to the SMM Range are directed. SMMEM_N is a special FSB message bit that uses multiplexed address bit FSBxA_N[7]. SMMEM_N is valid during the second half of the FSB request phase clock. (the clock in which FSBxADS_N is driven asserted).

SMMEM# asserted directs the accesses to the memory and SMMEM# deasserted directs the access to the PCI Express bus where VGA has been mapped.

When compatible SMM space is enabled, SMM-mode processor accesses to this range are routed to physical system DRAM at this address. Non-SMM-mode processor accesses to this range are considered to be to the video buffer area as described above. Graphics port and ESI originated cycles to enabled SMM space are not allowed and are considered to be to the Video Buffer Area.

Monochrome Adapter (MDA) Range (B 0000h–B 7FFFh)

The Intel® 5400 chipset does not support this range.

4.2.3 Expansion Card BIOS Area (C 0000h–D FFFFh)

This 128-KB ISA Expansion Card BIOS covers segments C and D. This region is further divided into eight, 16-KB segments. Each segment can be assigned one of four read/write states: read only, write only, read/write, or disabled. Typically, these blocks are mapped through the MCH and are subtractively decoded to ISA space. Memory that is disabled is not remapped.

Read and write transactions may be directed to different destinations with in the range C 0000h to D FFFFh. Historically, these blocks were used to shadow ISA device BIOS code. For the Intel® 5400 chipset, 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 Intel® 5400 chipset MCH.PAM registers. There is a PAM register for each sub-range that defines the routing of reads and writes.

Table 4-2. PAM Settings

PAM [5:4]/1:0]	Write Destination	Read Destination	Result
00	ESI	ESI	Mapped to ESI Port
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 port (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 Intel® 5400 chipset 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 C and D segments MUST be programmed to send accesses to DRAM.

4.2.4 Lower System BIOS Area (E 0000h–E FFFFh)

This 64-KB area, from E 0000h to E FFFFh, is divided into four, 16-KB segments. Each segment can be assigned independent read and write attributes through the Intel® 5400 chipset MCH.PAM registers. This area can be mapped either the ESI port (Intel 631xESB/632xESB I/O Controller Hub) or to main memory. Historically this area was used for BIOS ROM. Memory segments that are disabled are not remapped elsewhere.

The power-on default for these segments is to map them to the ESI port (Intel 631xESB/632xESB I/O Controller Hub). Software should not set cacheable memory attributes for any of these ranges unless both read and write transactions are mapped to main memory. Chipset functionality is not guaranteed if this region is cached.

For locks to this region, the Intel® 5400 chipset will complete them, but does not guarantee the atomicity of locked access to this range when writes and reads are mapped to separate destinations. If inbound transactions are expected, the E segment MUST be programmed to send these transactions to DRAM.

4.2.5 Upper System BIOS Area (F 0000h–F FFFFh)

This area is a single, 64-KB segment, from E 0000h - F FFFFh. This segment can be assigned read and write attributes through the Intel® 5400 chipset MCH.PAM registers. The power-on default is set to read/write disabled with transactions forwarded to the ESI port (Intel 631xESB/632xESB I/O Controller Hub). By manipulating the read/write attributes, the MCH can "shadow" BIOS into the main system memory. When disabled, this segment is not remapped.

For locks to this region, the Intel® 5400 chipset will complete them, but does not guarantee the atomicity of locked access to this range when writes and reads are mapped to separate destinations. If inbound transactions are expected, the F segment MUST be programmed to send these transactions to DRAM.

4.3 System Memory Area

The low/medium memory regions range from 1 MB to 4 GB. It consists of sub-regions for Firmware, Processor memory mapped functions, and Intel® 5400 chipset specific registers.

To support High Performance Computing systems, there are two non-coherent memory low ranges (one for each FSB). These ranges, when enabled, will not allocate SF and its coherence will not be checked. The size is programmable with minimum of 1MB. The range is programmable between 0x10_0000 and ESMMTOP-TSEG_SZ

The Extended Memory Area covers from 10 0000h (1 MB) to FFFF FFFFh (4 GB-1) address range and it is divided into the following regions:

- Main System Memory from 1 MB to the Top of Memory; 4 GB system memory.
- PCI Memory space from the Top of Memory to 4 GB with two specific ranges:
- APIC Configuration Space from FEC0 0000h (4 GB–20 MB) to FECF FFFFh and FEE0 0000h to FEEF FFFFh
- High BIOS area is from 16 MB to 4 GB - 1

Main System DRAM Address Range (0010 0000h to Top of System Memory)

The address range from 1 MB to the top of system memory is mapped to system memory address range controlled by the MCH. The Top of Main Memory (TOLM) is limited to 4-GB DRAM. All accesses to addresses within this range will be forwarded by the MCH to the system memory.

The MCH provides a maximum system memory address decode space of 4 GB. The MCH does not remap APIC memory space. The MCH does not limit system memory address space in hardware.

4.3.1 System Memory

See Section 4.3.9.

4.3.2 15 MB - 16 MB Window (ISA Hole)

The Intel® 5400 chipset does not support the legacy ISA hole between addresses F0 0000h - FF FFFFh. All transactions to this address range are treated as system memory.

4.3.3 Extended SMRAM Space (TSEG)

SMM space allows system management software to partition a region in main memory to be used by system management software. This region is protected for access by software other than system management software. When the SMM range is enabled, memory in this range is not exposed to the Operating System. The Intel® 5400 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. Intel® 5400 chipset uses the SMM enable and range registers to determine where to route the access.

Extended SMRAM Space is different than the SMM space defined with in the VGA address space, A 0000h - B FFFFh. This region is controlled by the Intel® 5400 chipset registers Intel® 5400 chipset MCH.EXSMRC.TSEG_SZ and Intel® 5400 chipset MCH.EXSMRTOP.ESMMTOP. The TSEG SMM space starts at ESMMTOP - TSEG_SZ and ends at ESMMTOP. This region may be 512 KB, 1 MB, 2 MB, or 4 MB 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 Top Of Low Memory (TOLM), defined by the TOLM register.

Intel® 5400 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.

In order to make cacheable SMM possible, the chipset must accept EWB's and must absorb IWB data regardless of the condition of the SMMEM# pin. The Intel® 5400 chipset 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 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.

4.3.4 Memory Mapped Configuration (MMCFG) Region

There is one relocatable memory mapped configuration region in the Intel® 5400 chipset MCH. The processor bus address defines the particular configuration register to be accessed. This configuration mechanism is atomic.

The memory mapped configuration region is compatible with the PCI Express enhanced configuration mechanism. The MMCFG region is a 256 MB window that maps to PCI Express registers on both the Intel® 5400 chipset and the south bridge.

The location of this MMCFG window is defined by the Intel® 5400 chipset MCH.HECBASE register. The HECBASE register could also be accessed through a fixed location. The default value of Intel® 5400 chipset MCH.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 4 GB. 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 4 GB Low/Medium Memory limit and not overlapping above 4 GB 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.

QWord (64 bits) accesses to this MMCFG address range are not supported except the sub-ranges that are mapped to Intel VT-d and other integrated devices configuration registers.

4.3.5 Low Memory Mapped I/O (MMIO)

This is the first of two Intel® 5400 chipset memory mapped I/O ranges. The low memory mapped I/O range is defined to be between Top Of Low Memory, (TOLM) and FE00 0000h. This low MMIO region is further subdivided between the PCI Express and ESI 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 4-3. Low Memory Mapped I/O[1]

I/O Port	MCH Base	MCH Limit
ESI	N/A[2]	N/A[2]
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
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
PEX8 Memory	MBASE8	MLIMIT8
PEX8 Prefetchable Memory	PMBASE8	PMLIMIT8
PEX9 Memory	MBASE9	MLIMIT9
PEX9 Prefetchable Memory	PMBASE9	PMLIMIT9

Notes:
1. This table assumes Intel® 5400 chipset MCH.PMLU and Intel® 5400 chipset MCH.PMBU are 0's. Otherwise, the prefetchable memory space will be located in high MMIO space.
2. 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® 5400 chipset MCH 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 MCH.PMLU and MCH.PMBU registers are greater than 0, then the corresponding prefetchable region will be located in the high MMIO range instead.

4.3.6 Chipset Specific Range

The address range FE00 0000h - FEBF FFFFh region is reserved for chipset specific functions.

FE00 0000h - FE01 FFFFh: This range (with size of 128 KB for four FB-DIMM channels; 16 Advanced Memory Buffer (AMB) per channel, 2 KB per AMB), is used for accessing AMB registers. These registers can only be accessed through memory mapped register access mechanism as MMIO. Notice that they are not accessible through CF8/CFC or MMCFG which are used 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 0000h - FE6F FFFFh: This range is used for fixed memory mapped Intel® 5400 chipset 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:

- Four Scratch Pad Registers
- Four Sticky Scratch Pad Registers
- Four Boot flag registers
- HECBASE register for MMCFG
- AMBASE register for AMB memory mapped registers

These registers are described in the Chapter 3, “Register Description.”

FE70 0000h - FE70 3FFFh: FE70_0000 - FE70_3FFF: Recommended range for CB_BAR MMIO. The integrated Intel® QuickData Technology Device device has a 16 KB MMIO space with a default range from 0 to 0000_3FFF. This range is enabled by MAEN bit of the PCICMD register (bit 1) in the integrated Intel® QuickData Technology Device device (dev. 15)and could be relocated by programming CB_BAR register. This recommended range could be used as a private MMIO space by software.

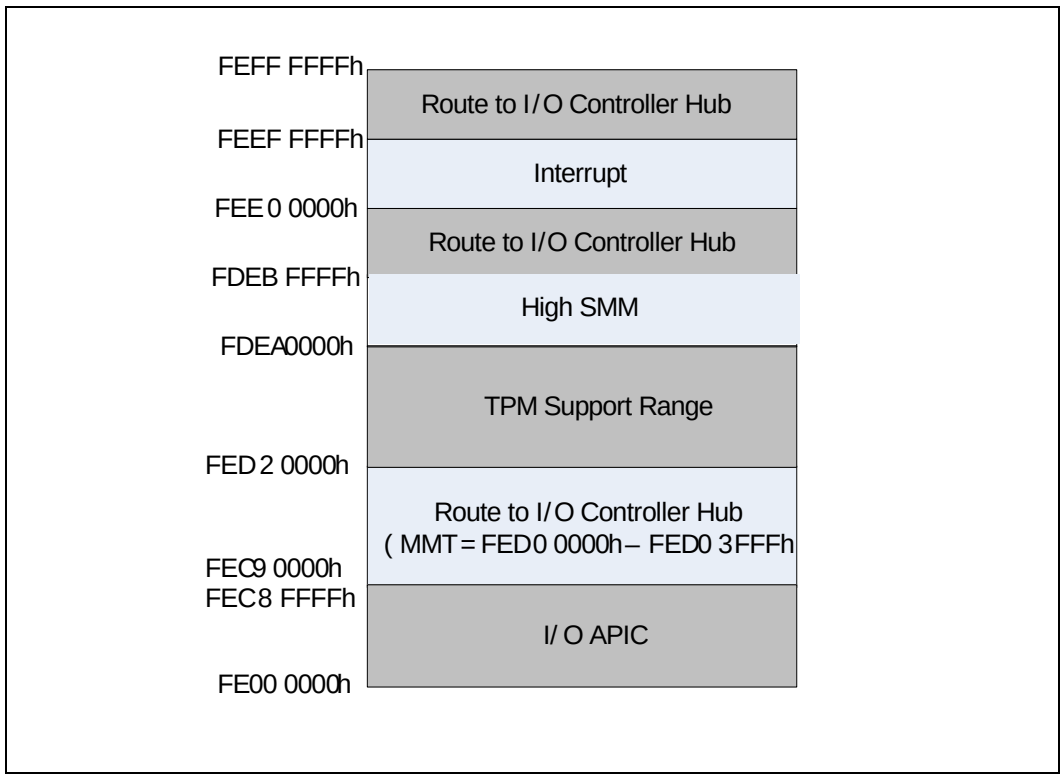
FE71 0000h - FE71 FFFFh: VT_BAR MMIO range. The integrated VTd devices have a 64 KB MMIO space with a default range from FE71_0000 to FE71_FFFF. This range could be relocated by programming VT_BAR register. This range is enabled by ENBVTBAR bit of VT_BAR register and could be used as a private MMIO space by software. All inbound transaction to this range will be completer aborted by Intel® 5400 chipset.

The Intel® 5400 chipset MCH will master abort requests to the remainder of this region unless they map into one of the relocatable regions such as MMCFG. The mechanism for this range can be the same as it is for the memory mapped configuration accesses.

4.3.7 Interrupt/SMM/TPM Region

This 4 MB range is used for processor specific applications. This region lies between FEC0 0000h and FEFF FFFFh and is split into four 1 MB segments.

Figure 4-3. Interrupt /SMM Region



This region is used to support various processor and system functions. These functions include I/O APIC control range which is used to communicate with I/O APIC controllers located on PXH, IXH, and Intel 631xESB/632xESB I/O Controller Hub devices. **The high SMM range is not supported in the MCH**. Transactions directed to this range are redirected to physical memory located in the compatible (legacy) SMM space; 0A 0000h - 0B FFFFh. The interrupt range is used to deliver interrupts. Memory read or write transactions from the processor are illegal.

4.3.7.1 I/O APIC Controller Range

This address range FEC0 0000h to FEC8 FFFF is used to communicate with the IOAPIC controllers in the PXH, IXH, or Intel 631xESB/632xESB I/O Controller Hub devices.

The APIC ranges are hard coded. Reads and writes to each IOAPIC region should be sent to the appropriate ESI or PCI Express port as indicated below if EXSMRC.APICDIS = 0. If EXSMRC.APICDIS is set to 1, then this range is treated as a regular address range and accessing requests will be sent to ESI.

Table 4-4. I/O APIC Address Mapping (Sheet 1 of 2)

IOAPIC0 (ESI)	FEC0 0000h to FEC7 FFFFh
IOAPIC1 (PEX1)	FEC8 0000h to FEC8 0FFFh
IOAPIC2 (PEX2)	FEC8 1000h to FEC8 1FFFh

Table 4-4. I/O APIC Address Mapping (Sheet 2 of 2)

IOAPIC3 (PEX3)	FEC8 2000h to FEC8 2FFFh
IOAPIC4 (PEX4)	FEC8 3000h to FEC8 3FFFh
IOAPIC5 (PEX5)	FEC8 4000h to FEC8 4FFFh
IOAPIC6 (PEX6)	FEC8 5000h to FEC8 5FFFh
IOAPIC7 (PEX7)	FEC8 6000h to FEC8 6FFFh
IOAPIC8 (PEX8)	FEC8 7000h to FEC8 7FFFh
IOAPIC9 (PEX9)	FEC8 8000h to FEC8 8FFFh
IOAPIC10 (integrated IOAPIC)	FEC8 9000h to FEC8 9FFFh
Reserved (Intel 631xESB/632xESB I/O Controller Hub for master abort)	FEC8 A000h to FEC8 FFFFh

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.

Refer to Section 3.10.1.5, "MBAR: IOxAPIC Memory Base Address Register" on page 246 and Section 3.10.1.9, "ABAR: I/OxAPIC Alternate BAR" for the accessing the integrated IOxAPIC controller using its MBAR and ABAR registers. MBAR range is enabled by Memory_Space_Enable bit of PCICMD register in the integrated IOAPIC controller device (dev. 18) and is defined at 4 KB boundary. ABAR range is enabled by ABAR_Enable bit of IOAPIC_ABAR register and is defined at 4 KB boundary.

Inbound accesses to this memory range are not allowed.

4.3.7.1.1 TPM Support Range

The MCH 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 Intel 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. The 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 4-5, "TPM Support Range Address Routing" on page 278.

Table 4-5. 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 through FED4_0FFF will be routed to ESI.

Locked transaction to the FED4_0xxx from FSB will be master aborted. Note: There will be no Intel Trusted Execution Technology support in the MCH although Intel Trusted Execution Technology transaction to ESI is used internally to communicate TPM access.

4.3.7.2 Interrupt Range

Requests to the address range FEE0 0000h to FEEF FFFFh 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 interrupt requests from the PCI Express devices in the form of memory writes are converted by the MCH to processor bus interrupt requests.

4.3.7.3 Reserved Ranges

The Intel® 5400 chipset MCH will master abort requests to the addresses in the interrupt/reserved range (FEC0 0000h - FEFF FFFFh) which are not specified. This can be done by sending the request to the compatibility bus (ESI) to be master aborted.

4.3.7.4 Firmware Range

The Intel® 5400 chipset platform allocates 16 MB of firmware space from FF00 0000h to FFFF FFFFh. Requests in this range are directed to the Compatibility Bus. The Intel 631xESB/632xESB I/O Controller Hub will route these to its FWH interface. This range is accessible from any processor bus.

To support High Performance Computing systems, there are two non-coherent memory high ranges (one for each FSB). These ranges, when enabled, will not allocate SF and its coherence will not be checked. The size is programmable with minimum of 256 MB. The range is programmable above the 4 GB address line.

4.3.8 High Extended Memory

This is the range above 4 GB. The range from 4 GB to MCH.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 1 TB).

4.3.8.1 System Memory

See Section 4.3.9.

4.3.8.2 High MMIO

The high memory mapped I/O region is located above the top of memory as defined by MCH.MIR[1].LIMIT. These MCH.PMBU and MCH.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[1].LIMIT and it falls within the MCH.PMBU+PMBASE and MCH.PMLU+PMLIMIT range, it should be routed to the appropriate PCI Express port. For accesses above MIR[1].LIMIT (and above 4 GB) that are not in a high MMIO region, they should be master aborted.

4.3.8.3 Extended Memory

The range of memory just below 4 GB from TOLM to 4 GB (Low MMIO, Chipset, Interrupt/SMM/Intel Trusted Execution Technology) does not map to memory. If the DRAM memory, behind the TOLM to 4 GB range, is not relocated, it will be unused.

On LH, the memory reclamation was done by adding a range above top of memory before MMIO was allocated (called Extended Memory), and mapping the newly added range into the memory that would have been lost. On Intel® 5400 chipset, this range doesn’t have any meaning because hardware reclamation mechanism is different. However, since both chipsets handle the reclamation with hardware, the only difference

to software is when there is more than 4 GB of memory + (Low MIO, Chipset, etc.), rather than using the REMAPLIMIT value of Cayuse to indicate the top of usable memory, Intel® 5400 chipset will use MIR[1].LIMIT to indicate the top of usable memory. (If there is less than 4 GB of memory + (Low MMIO, etc.), both chipsets use TOLM to indicate the top of memory.) Note that ESMMTOP cannot be greater than TOLM; otherwise Chipset functionality is not guaranteed.

4.3.9 Main Memory Region

4.3.9.1 Application of Coherency Protocol

The Intel® 5400 chipset MCH applies the coherency protocol to all accesses to main memory. Application of the coherency protocol includes snooping the other processor bus.

Two exceptions to this rule are the Expansion Card BIOS area, 0C 0000h - 0F FFFFh and the legacy SMM, 0A 0000h - 0B FFFFh, range. The Expansion Card BIOS area 0C 0000h - 0F FFFFh may not necessarily route both reads and writes to memory, the legacy SMM range, 0A 0000h - 0B FFFFh, may target non-memory when not in SMM mode. The coherency protocol is not applied to these two exceptions.

4.3.9.2 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 in the MCH.

The MCH.MIR.LIMIT registers will decode an access into a specific interleaving range. Within the interleaving range, the MCH.MIR.LIMIT register indicates which FB-DIMM memory branch the address is associated with.

It is a programming error if the highest MIR’s LIMIT is less than TOLM.

In addition, 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 FB-DIMM interleaving policies). The DMIR.LIMIT registers also decode the access into a specific interleaving range, which helps decode which rank the access will be directed to. MCH will also convert the address into the appropriate DDR address, RAS, and CAS signals.

4.4 Memory Address Disposition

The following section presents a summary of address dispositions for the Intel® 5400 chipset MCH.

4.4.1 Registers Used for Address Routing

Table 4-6 is a summary of the registers used to control memory address disposition. These registers are described in detail in Section 3.

Table 4-6. Intel® 5400 Chipset MCH Memory Mapping Registers

Name	Function
MIR[2:0]	Memory Interleaving Registers (FB-DIMM 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
IOAPIC_MBAR/ABAR	MMIO window memory/alternate base address registers for integrated IOAPIC controller device.
VT_BAR	MMIO window base address register for Intel VT device
MBASE (dev 1-9)	Base address for memory mapped I/O to PCI Express ports 1-9
MLIMIT (dev 1-9)	Limit address for memory mapped I/O to PCI Express ports 1-9
PMBASE (dev 1-9)	Base address for memory mapped I/O to prefetchable memory of PCI Express ports 1-9[1]
PMLIMIT (dev 1-9)	Limit address for memory mapped I/O to prefetchable memory of PCI Express ports 1-9
PMBU (dev 1-9)	Prefetchable Memory Base (Upper 32 bits) - Upper address bits to the base address of prefetchable memory space. If the prefetchable memory is below 4 GB, this register will be set to all 0's.
PMLU (dev 1-9)	Prefetchable Memory Limit (Upper 32 bits) - Upper address bits to the limit address of prefetchable memory space. If the prefetchable memory is below 4 GB, this register will be set to all 0's.
PCICMD (dev 1-9)	MSE (Memory Space Enable) bit enables the memory and pre-fetchable ranges.

Notes:

1. The chipset treats memory and prefetchable memory the same. These are just considered 2 apertures to the PCI Express port.

4.4.2 Address Disposition for Processor

The following tables define the address disposition for the Intel® 5400 chipset MCH. Table 4-7 defines the disposition of outbound requests entering the MCH on the processor bus. Table 4-11 defines the disposition of inbound requests entering the MCH on an I/O bus. For address dispositions of PCI Express/ESI devices, please refer to the respective product specifications for the PXH or Intel 631xESB/632xESB I/O Controller Hub.

Table 4-7. Address Disposition for Processor (Sheet 1 of 2)

Address Range	Conditions	MCH Behavior
DOS	0 to 09FFFFh	Coherent Request to Main Memory. Route to main memory according to MCH.MIR registers. Apply Coherence Protocol.
SMM/VGA	0A0000h to 0BFFFFh	see Table 4-9, "SMM Memory Region Access Control from Processor" and Table 4-10, "Decoding Processor Requests to SMM and VGA Spaces".
C and D BIOS segments	0C0000h to 0DFFFFh and PAM=11	Non-coherent request to main memory. Rout to appropriate FB-DIMM device according to MCH.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	0E0000h to 0FFFFFh and PAM=11	Non-coherent request to main memory. Rout to appropriate FB-DIMM device according to MCH.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 Intel® 5400 chipset MCH.MIR registers. Coherence protocol is applied. Note: the extended SMRAM space is within this range.
Extended SMRAM Space	ESMMTOP-TSEG_SZ <= Addr < ESMMTOP	see Table 4-9 and Table 4-10.
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+256 MB	Convert to a configuration access and route according to the Configuration Access Disposition.
Intel® 5400 chipset 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 such as CB_BAR or VT_BAR ranges or to the FB-DIMM based on the context.
	FE00_0000h to FEBF_FFFFh AND (NOT a valid SG memory mapped register address or NOT a valid AMB targeted 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 4-4, "I/O APIC Address Mapping".
Intel 631xESB/632xESB I/O Controller Hub/ Intel 631xESB/632xESB I/O Controller Hub timers	FEC9_0000h to FED1_FFFF	Issue request to ESI.
TPM support range	FED2_0000 to FED9_FFFF	see Table 4-5, "TPM Support Range Address Routing" on page 278

Table 4-7. Address Disposition for Processor (Sheet 2 of 2)

Address Range	Conditions	MCH Behavior
Interrupt	interrupt transaction to FEE0_0000h to FEEF_FFFFh (not really memory space)	Route to appropriate FSB(s).
	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 MCH.MIR registers. Coherence protocol is applied.
High MMIO	PMBU+PMBASE <= Addr <= PMLU+PMLIMIT	Route request to appropriate PCI Express port
All others	All Others (subtractive decoding)	Issue request to ESI.

4.4.2.1 Access to SMM Space (Processor Only)

Accesses to SMM space are restricted to processors, inbound transactions are prohibited. Inbound transactions to enabled SMM space are not allowed and the MCH will set MCH.EXSMRAMC.E_SMERR bit. The following table defines when a SMM range is enabled. All the enable bits: G_SMRAME and TSEG_EN are located in the MCH.EXSMRC register.

Table 4-8. Enabled SMM Ranges

Global Enable G_SMRAME	TSEG Enable TSEG_EN	Legacy SMM Enabled?	HIGH SMM Enabled?	Extended SMRAM Space (TSEG) Enabled?
0	X	No	No	No
1	0	Yes	No	No
1	1	Yes	No	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 4-9 defines the access control of SMM memory regions from processors.

Table 4-9. SMM Memory Region Access Control from Processor (Sheet 1 of 2)

G_SMRAME	D_LCK	D_CLS	D_OPEN	SMMEM#	Code Access to SMM Memory[1]	Data Access to SMM Memory[2]
0[3]	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[4]	x	1	1	x	illegal settings	illegal settings
1	1	1	0	1	yes	no (legacy SMM) yes (H_SMM, TSEG)

Table 4-9. SMM Memory Region Access Control from Processor (Sheet 2 of 2)

G_SMRAME	D_LCK	D_CLS	D_OPEN	SMMEM#	Code Access to SMM Memory[1]	Data Access to SMM Memory[2]
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:
1. BRLC
2. Data access transaction other than BRLC
3. For access to TSEG region (address range between ESMMTOP - TSEG_SZ and ESMMTOP), Intel® 5400 chipset MCH will route to identical system memory by definition (as TSEG is not enabled).
4. It is a programming error if D_CLS and D_OPEN are both set to 1, Intel® 5400 chipset MCH's behavior is undefined. Intel® 5400 chipset MCH could master abort SMM access.

The MCH 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 MCH.SMRAMC, MCH.EXSMRC, and MCH.BCTRL registers. Table 4-10 defines Intel® 5400 chipset 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 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 Table 4-11.

Table 4-10. Decoding Processor Requests to SMM and VGA Spaces

SMM region	Transaction Address Range	SMM Memory Address Range	SMM Access Control[1]	G_SMRAME	H_SMRAME	T_EN	EWB/IWB	Routing
Legacy VGA/SMM [2]	A_0000h to B_FFFFh	A_0000h to B_FFFFh	x	0	x	x	x	to the VGA-enabled port (in BCTRL); otherwise, ESI[3]
			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 Log B25 error

Notes:
1. SMM memory access control, see Table 4-9.
2. Software must not cache this region.
3. One and only one BCTRL can set the VGAEN; otherwise, send to ESI.

MCH

4.4.3 Inbound Transactions

In general, inbound 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 Intel® QuickData Technology Device is treated as an I/O device, thus accesses initiated by the Intel® QuickData Technology Device are considered as inbound accesses. Intel® 5400 chipset MCH will master abort the IB transaction targeting at enabled TSEG SMM range

For all table entries where an access is forwarded to ESI to be master aborted, if an access comes from ESI, the Intel® 5400 chipset MCH ESI may master abort a transaction without forwarding it back to the ESI.

Table 4-11. Address Disposition for Inbound Transactions (Sheet 1 of 2)

Address Range	Conditions	MCH Behavior
DOS	0 to 09FFFFh	Coherent Request to Main Memory. Route to main memory according to Intel® 5400 chipset MCH.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[1]
C, D, E, and F BIOS segments	0C0000h to 0FFFFFh and PAM=11[2]	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 FB-DIMM according to Intel® 5400 chipset MCH.MIR registers.
Low/Medium Memory	10_0000 <= Addr < ESMMTOP - TSEG_SZ	Coherent Request to Main Memory. Route to main memory according to Intel® 5400 chipset MCH.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 Master aborted by MCH. 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+256 MB	Inbound MMCFG access is not allowed and will be aborted.
Intel® 5400 chipset specific	FE00_0000h to FEBF_FFFFh AND valid SG memory mapped register address	Inbound MMCFG access is not allowed and will be aborted.
	FE00_0000h to FEBF_FFFFh AND NOT a valid SG 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 4-4
Intel 631xESB/632xESB I/O Controller Hub / Intel 631xESB/632xESB I/O Controller Hub timers	FEC9_0000h to FED1_FFFF	Issue request to ESI.
TPM support range	FED2_0000 to FED9_FFFF	see Table 4-5, "TPM Support Range Address Routing" on page 278

Table 4-11. Address Disposition for Inbound Transactions (Sheet 2 of 2)

Address Range	Conditions	MCH Behavior
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 Intel® 5400 chipset MCH.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:

1. One and only one BCTRL can set the VGAEN; otherwise, send to ESI for master abort.
2. Other combinations of PAM's are not allowed if inbound accesses to this region can occur. Just like Cayuse, chipset functionality is not guaranteed.

4.5 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 4.5.1, 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.

4.5.1 Special I/O Addresses

There are two classes of I/O addresses that are specifically decoded by the Intel® 5400 chipset 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.

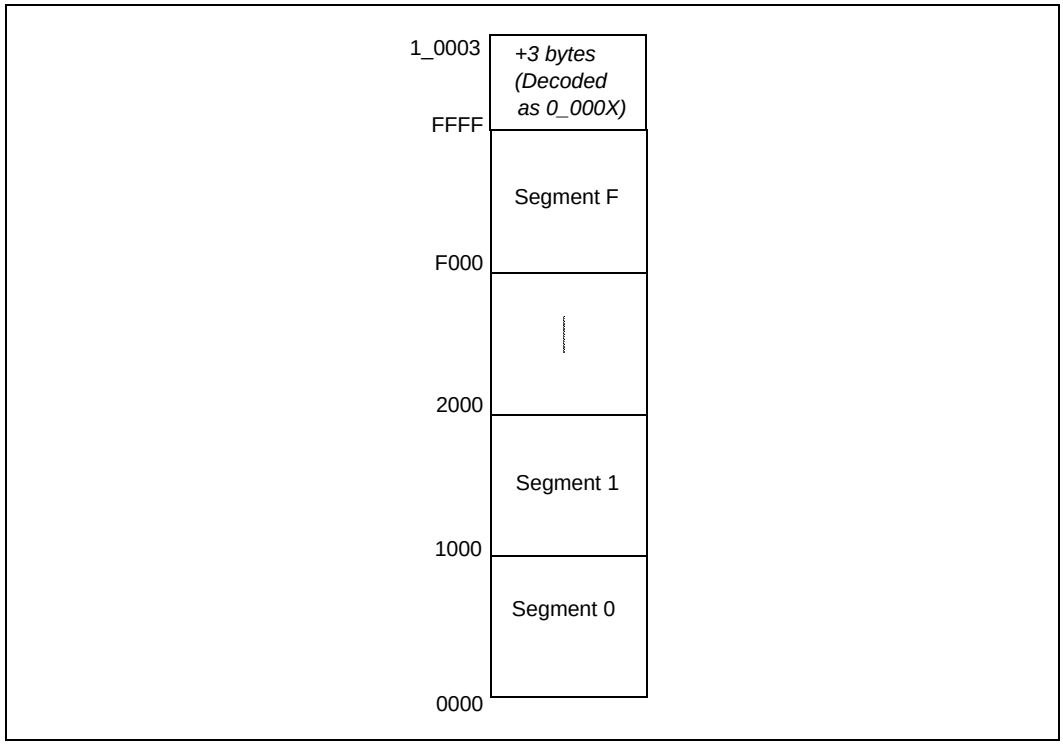
Historically, the 64 K I/O space actually was 64 K+3 bytes. For the extra three bytes, A#[16] is asserted on FSB. The Product Name decodes only A#[15:3] when the request encoding indicates an I/O cycle. Therefore first three byte I/O accesses with A#[16] asserted are decoded as if they were accesses to the first three bytes starting from I/O addresses 0 (wrap-around the 64 KB line). A[16] is not forwarded by Intel® 5400 chipset MCH.

At power-on, all I/O accesses are mapped to the ESI.

4.5.2 Outbound I/O Access

The Intel® 5400 chipset MCH 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, must be programmed to send to the ESI for compatibility.

Figure 4-4. System I/O Address Space



4.5.2.1 Outbound I/O Accesses Routing

The MCH 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 PCICMD[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 PCICMD[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 MCH 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 PCICMD[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 block.
4. I/O defined by IOBASE/IOLIMIT: If PCICMD[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).

4.5.3 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 transaction for configuration access to the Intel® 5400 chipset MCH 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.

4.6 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 4.3.4. Individual register maps are in the registers chapters of the Intel® 5400 chipset MCH Component Specifications.

If a CPU issues a zero length configuration cycle accessing the Intel® 5400 chipset MCH's internal configuration space registers or the CB_BAR/AMB Memory mapped area, then it will be completed on the FSB "in order" with no data.

4.7 Intel VT-d Address Map Implications

Intel VT-d applies only to inbound memory transactions. Inbound I/O and configuration transactions are not affected by Intel VT-d. Inbound I/O, configuration and message decode and forwarding happens the same whether Intel VT-d is enabled or not. For memory transaction decode, the host address map in Intel VT-d corresponds to the address map discussed earlier in the chapter and all addresses after translation are subject to the same address map rule checking (and error reporting) as in the non-Intel VT-d mode. There is not a fixed guest address map that the MCH Intel VT-d hardware can rely up on (except that the guest domain addresses cannot go beyond the guest address width specified in the context-entry contents) i.e. it is OS dependent. The MCH converts all incoming memory guest addresses to host addresses and then applies the same set of memory address decoding rules as described earlier.

§

5 Functional Description

This chapter describes each of the MCH interfaces and functional units including the Dual Independent Bus (DIB) processor Frontside Bus (FSB) interface, the PCI Express ports, system memory controller, power management, and clocking.

5.1 Processor Front Side Buses

The MCH supports two Dual-Core Intel Xeon Processor 5100 series, Quad-Core Intel Xeon Processor 5300 series, Quad-Core Intel Xeon Processor 5400 series and Dual-Core Intel Xeon Processor 5200 series are based on 65 nm and 45 nm processes in a 771-land, FC-LGA4 package. The processors are based upon a 32-bit Intel® Xeon processor supporting Intel® 64 Architecture based on Intel® Core™ Microarchitecture.

The MCH supports 1066 MT/s, 1333 MT/s and 1600 MT/s FSB which is a quad-pumped bus running off a 266/333/400 MHz clock, and a point to point DIB processor system bus interface. Each processor FSB supports peak address generation rates of 533/667/800 Million Addresses/second. Both FSB data buses are quad pumped 64-bits which allows peak bandwidths inbound of 8 GB/s (1066 MT/s) and 10 GB/s (1333 MT/s) and out bound of 17 GB/s and 21 GB/s, respectively. The MCH supports 38-bit host addressing, decoding up to 128 GB of the processor's memory address space. Host-initiated I/O cycles are decoded to PCI, PCI Express, ESI interface or MCH configuration space. Host-initiated memory cycles are decoded to PCI, PCI Express, ESI or system memory.

Unless otherwise specified, the term processor in this document refers jointly to the Quad-Core Intel Xeon Processor 5300 series, Quad-Core Intel Xeon Processor 5400 series and Dual-Core Intel Xeon Processor 5200 series.

5.1.1 FSB Overview

The MCH is the only priority agent for two point to point, independent, processor front side buses (FSB). These two buses are referred to as Dual Independent Buses (DIB). The MCH maintains coherency across these two buses. The MCH may complete deferrable transactions with either defer-replies or in-order responses. Intel® 5400 chipset contains an internal Snoop-Filter 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 38 bit address bus, a 64 bit data bus, and associated control signals. The FSB utilizes 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 FSB data bus is quad pumped (4X) and supports peak bandwidths. Parity protection is applied to the data bus.

Interrupts are also delivered via the FSB.

5.1.2 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{0/1}DBI_N[3:0] signals for FSB0 and FSB1, respectively indicate if the corresponding 16 bits of data are inverted on the bus for each quad pumped data phase.

Table 5-1. FSB{0/1}DBI_N[3:0] Data Bit Correspondence

FSB0DBI_N[3:0]	Data Bits
FSB{0/1}DBI_N[0]	FSB{0/1}D_N[15:0]
FSB{0/1}DBI_N[1]	FSB{0/1}D_N[31:16]
FSB{0/1}DBI_N[2]	FSB{0/1}D_N[47:32]
FSB{0/1}DBI_N[3]	FSB{0/1}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{0/1}DBI_N[3:0] 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{0/1}DBI_N[3:0] to determine if the corresponding data segment should be inverted.

5.1.3 FSB Interrupt Overview

The processor 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 Intel® 5400 chipset platform, FSB interrupts may originate from the processor on the system bus, or from a downstream device on the Enterprise South Bridge Interface (ESI). In the later case, the MCH drives the Interrupt Message Transaction onto the system bus.

In the Intel® 5400 chipset the Intel 631xESB/632xESB I/O Controller Hub contains I/OxAPICs, 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.

5.1.3.1 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 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.

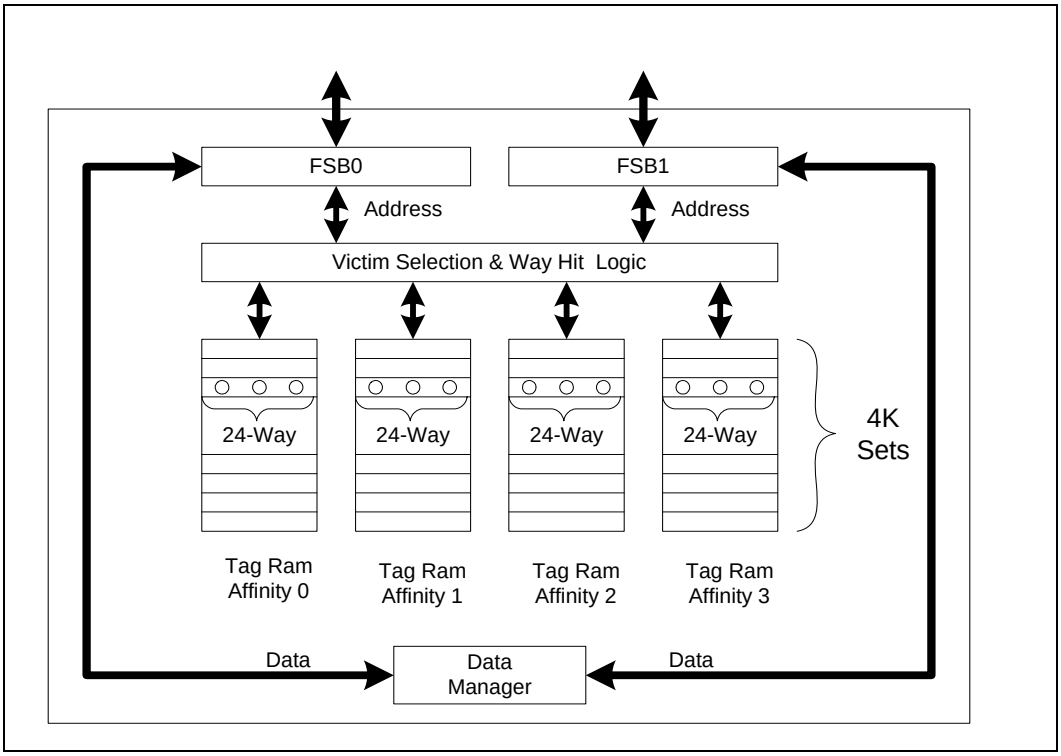
5.2 Snoop Filter

The Snoop Filter (SF) eliminates traffic on the snooped frontside bus of the processor being snooped. By removing snoops from the snooped bus, the full bandwidth is available for other transactions. Supporting concurrent snoops effectively reduces performance degradation attributable to multiple snoop stalls.

The SF is composed of four affinity groups each containing 4 K sets of x24-way associative entries. The overall SF size is 24 MB. Each affinity group supports a Active-way management algorithm. Lookups are done on a 96 way lookup, full 24-way per set for 4 sets for hit/miss checks.

The SF is organized as shown in Figure 5-1.

Figure 5-1. Snoop Filter



The snoop filter is organized in four parts referred to as the Tag Ram Affinity Groups, Affinity[3:0]. Each Affinity Group is associated with each last level cache. Under normal conditions a snoop is competed with a 1 snoop stall penalty. When the processors request simultaneous snoops the first snoop is completed with a one snoop stall penalty, the second snoop requires a 2 snoop stall penalty.

During SF access arbitration, processor 0 is given priority over processor 1. Thus simultaneous snoops are resolved with a 1 snoop stall penalty for processor 0 and a 2 snoop stall penalty for processor 2.

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 an entry, update/add an entry, or invalidate an entry in the snoop filter.

The SF has the following features:

- Snoop Filter tracks total of 24 MB of processor L2 cache lines, this is equivalent to: (24 * (2^{20})byte) / 64 byte CL = 393,216 cache lines.
- The SF is configured in 4 K sets organized as a 4 DID Affinity x 24 Way x 4 K Set-Associativity array. This is equivalent to (2^{12} Sets) x 24 Way x 4 DID = 393,216 tag entries
- 4 x 24 Affinity Set-Associativity will allocate/evict entries within the 24-way corresponding to the assigned affinity group if the SF look up is a miss. Each SF look up will be based on 96-way (4x24 ways) look up.
- The size of the snoop filter Tag RAM is: 4096 sets * 4 affinities * 24 ways * 33 bits/affinity/ set/way = 1,622,016 bytes
- The size of the snoop filter Victim Ram is: 4096 sets * 4 affinities * 8 bits = 16,384 bytes
- The size of the snoop filter Random ROM is: 1024 addresses * 16 bits = 2,048 bytes
- The Snoop Filter is operated at 2x of Intel® 5400 chipset core frequency, i.e. 533 MHz to provide 267 MLUU/s (where a Look-Up-Update operation is a read followed by a write operation to the tag).
  - The maximum lookup and update bandwidth of the Snoop Filter is equal to the max request bandwidth from both FSB's. The lookup and update bandwidth from I/O coherent transactions have to share the bandwidth with both FSBs per request weighted-round-robin arbitration.
  - The SF lookup latency is four SF-clocks or two Intel® 5400 chipset core clocks to support single snoop stall in idle condition (single request issued from either bus). If both bus are making requests simultaneously, the snoop-filter will always select bus 0 first. In such scenario, bus 0 request will have one snoop-stall and bus 1 request will have two snoop-stalls.
- Active Way / Invalid / E/M / Pseudo-Random replacement algorithm, with updates on lookups and invalidates, Invalid / Pseudo-Random replacement algorithm, with updates on lookups and invalidates.
- Tag entries support a 38-bit physical address space. The MCH supports an external address space of 38 bits as well.
- Stores coherency state (EM) and Bus[1:0] for each valid cache line in the system. The tracking algorithm utilizes conservative tracking (super-set tracking). The processor can silently down grade a line state from E to S/I or S to I without any action appearing on the FSB. Therefore, a line appearing in the SF as E states may actually missed in the corresponding processor caches. Conversely a SF S-line will never be found in E/M state in a processors L2 cache, or a SF miss will never be found in M/E/S state in a processors L2 cache. The following is the summery 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 share state
  - If Bus[1:0]=00, the entry is invalid.
  - If Bus[1:0]=01, the FSB0 processor(s) has ownership of the line.
  - If Bus[1:0]=10, the FSB1 processor(s) has ownership of the line.
  - If Bus[1:0]=11, both buses have ownership and the line must be shared by both FSB processors (EM must be 0).
  - EM||Bus[1:0] =111 is a reserved definition.

- ECC coverage, with correction of single bit errors, detection of double bit errors (SEC-DED).
  - Invalid/pRandom Array does not implement ECC or parity protection. A bit failure will result in the selection of the wrong victim entry and may have a minimal impact on performance. However, the coherency engine will resolve the conflict and guarantee correctness.

5.3 System Memory Controller

The MCH masters four First Generation Fully-Buffered DIMM (FB-DIMM1) memory channels. Up to four DIMMs can be connected to each channel (up to sixteen DIMMs for the entire array). 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 key features of the FB-DIMM memory interface are summarized in the following list.

- Four Fully Buffered DDR (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 four Fully Buffered - DDR2 DIMMs (FB-DIMM1).
- Supports up to 16 dual-ranked FB-DDR2 8GB DIMMs, i.e. 128 GB of physical memory
- The FB-DIMM link speed is at 6x the DDR 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@1333 MT/s) and A 4.8 GHz FB-DIMM link Supports DDR2-800 (FSB@1600 MT/s)
- The MCH will comply with the FB-DIMM specification definition of a host and will be compatible with any FB-DIMM-compliant DIMM.
- Special single channel, single DIMM operation mode (Branch 0, Channel 0, Slot 0 position only).
- All memory devices must be DDR2.

Table 5-2 and Figure 5-4 present system memory capacity as a function of DRAM device capacity and MCH operating mode.

Table 5-2. Minimum System Memory Configurations & Upgrade Increments

DRAM Technology	Smallest System Configuration - One DIMM	Smallest Upgrade Increment - Two DIMM
512 Mb	512 MB	1024 MB
1024 Mb	1024 MB	2048 MB
2048 Mb	2048 MB	4096 MB
4096 Mb	4096 MB	8192 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 5-3. Maximum 16 DIMM System Memory Configurations

DRAM Technology x8 Single Rank	Maximum Capacity
512 Mb	8 GB
1024 Mb	16 GB
2048 Mb	32 GB
4096 Mb	128 GB

Note: The *Maximum Capacity e* 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 5-4. Maximum 16 DIMM System Memory Configurations

DRAM Technology x4 Dual Rank	Maximum Capacity
512 Mb	32 GB
1024 Mb	128 GB
2048 Mb	128 GB
4096 Mb	128 GB

Note: The *Maximum Capacity* columns represent the system memory available when all DIMM slots are populated with identical x4 Double Rank (x4DR) DIMMs using the DRAM Technology indicated.

5.3.1 Memory Population Rules

DIMM population rules depend on the operating mode of the MC. When operating in non-mirrored mode the minimum memory upgrade increment is two identical DIMMs per branch (DIMMs must be identical with respect to size, speed, and organization). Non-mirrored mode has an exceptional mode that operates with a single DIMM which is discussed in the following section.

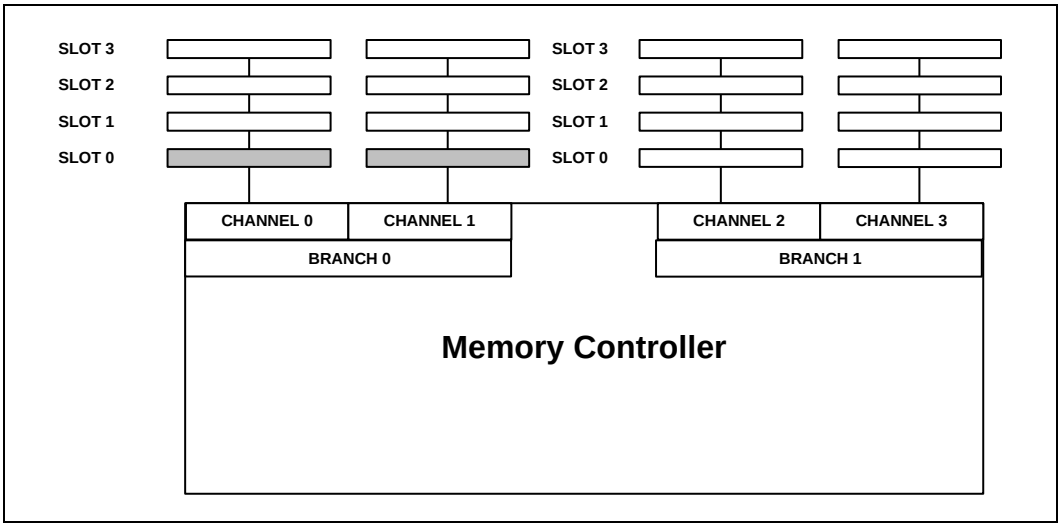
5.3.1.1 Memory Upgrades

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.

Figure 5-2 depicts the minimum two DIMM configuration. The populated DIMMs are depicted in gray (Slot 0 of Branch 0 populated).

Figure 5-2. Minimum Two DIMM Configuration



5.3.1.2 Enhanced 2-DIMM Memory Mode

When installing a total of 2 DIMMS improved performance can be obtained by populating 1 DIMM in Channel 0 of Branch 0 and the second DIMM in Channel 2 of Branch 1 as opposed to installing the 2 DIMMS in Channels 0 and 1 of Branch 0.

Figure 5-3 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 5-3. Next Two DIMM Upgrade Positions

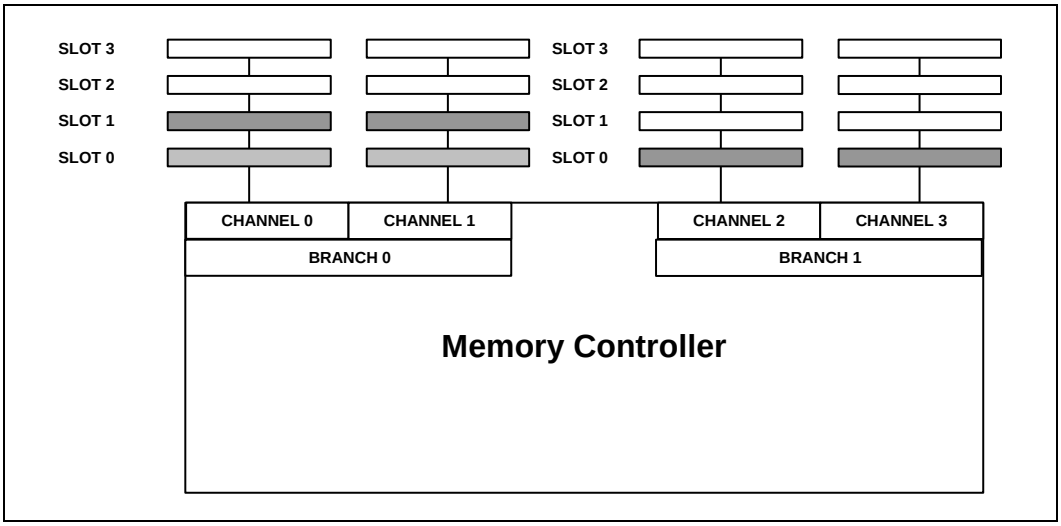
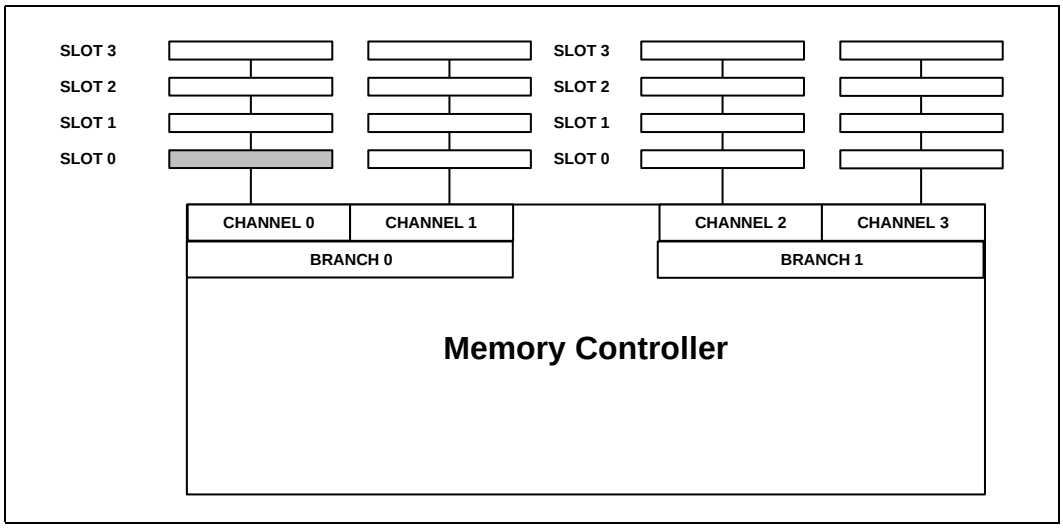


Figure 5-4 depicts a special single DIMM non-mirrored operation mode. This mode requires that the DIMM be placed in Branch 0, Channel 0, Slot 0. When upgrading from this mode the normal two DIMM memory upgrade rules are followed.

Figure 5-4. Depicts a Special Single DIMM Operation Mode



5.3.2 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 the dual needs for higher bandwidth and larger memory sizes.

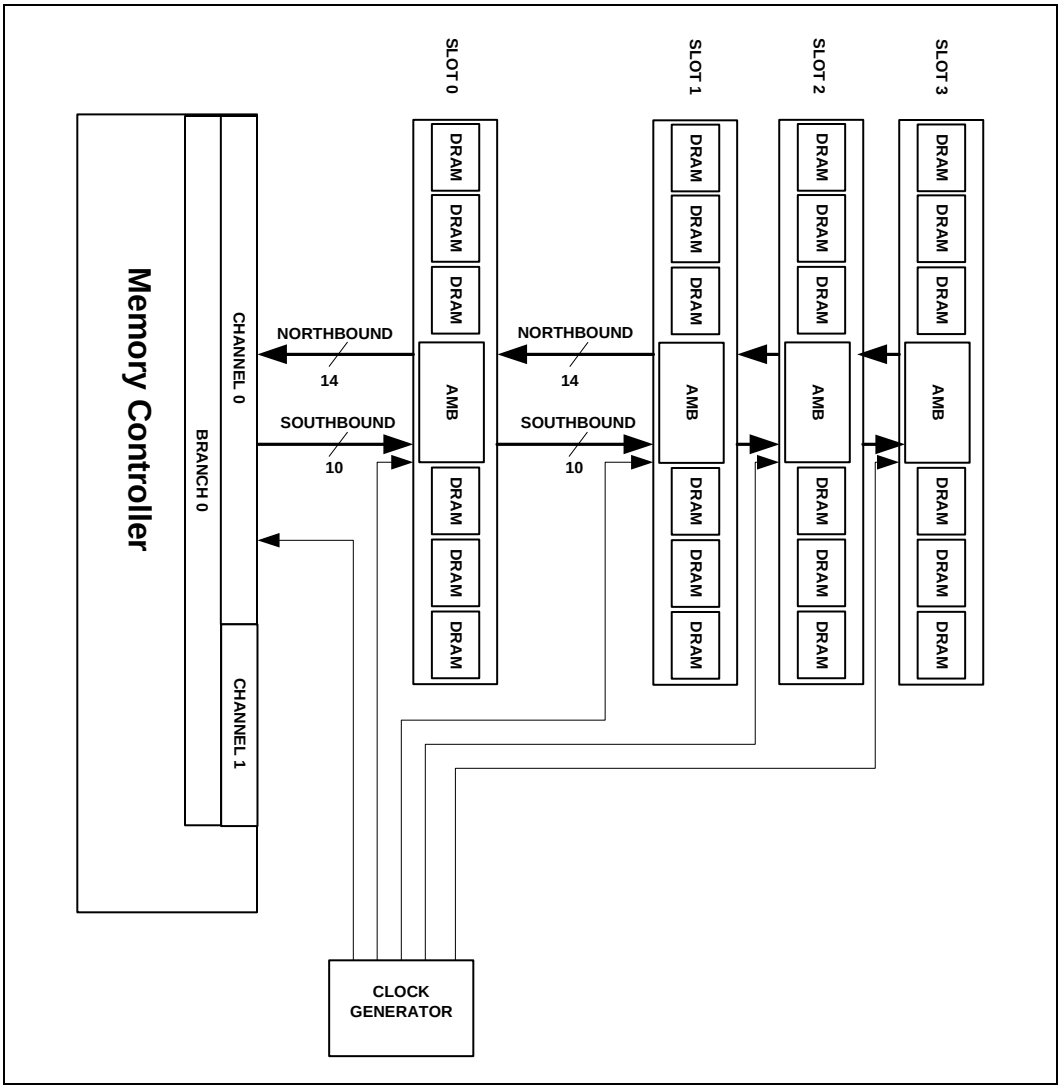
FB-DIMM memory DIMMs contain an Advanced Memory Buffer (AMB) device that serves as an interface between the point to point FB-DIMM Channel links and the DDR2 DRAM devices. Each AMB is capable of buffering up to two ranks of DRAM devices. Each AMB supports two complete FB-DIMM channel interfaces. The 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 5-5 depicts a single FB-DIMM channel with these three signal groups.

Figure 5-5. 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.

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.

5.3.3 FB-DIMM Memory Operating Modes

The MCH only supports non-mirrored mode operation.

5.3.3.1 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 128 GB is available. Normally when operating in non-mirrored mode both channels on a branch are operated in lock step, referred to as dual-channel mode. There is a single DIMM, single channel mode of operation referred to as single-channel mode.

5.3.3.1.1 Non-Mirrored Mode ECC

ECC is supported differently for each of these single- and dual-channel modes:

Dual-Channel Mode:

When branches operate in dual-channel mode, the MCH supports the 18 device DRAM failure correction code option for FB-DIMM. As applied by 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 DIMMs. This includes any pair of single bit errors.
- Detection of all permutations of two x4 DRAM failures.

Single-Channel Mode:

When the branch operates in single-channel mode, the MCH supports an 8-byte-over-32-byte Single Error Correct Double Error Detect, SECDED+ code. It is the same ECC code used in the dual-channel mode, but the number of devices over which the codeword is defined is halved, thereby reducing the SDDC properties to SECDED+. As applied by Intel® 5400 chipset, this code has the following properties:

- Detection of any two bits in error within 8 bytes of data
- Correction of any single bit in error within 8 bytes of data
- Correction of any single DRAM-data-bus-failure

5.3.3.2 Single Device Data Correction (SDDC) Support

The Intel 5400 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. 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 are as follows:

- Two Phase Code covering 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 5-5.

Table 5-5. 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 plus a single bit error (SBE), i.e. to avoid silent data corruption in the event of a particle induced error while correcting for a failed device, SG 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, SG detects a correctable error with a weight >1 and the "corrected" symbol does not match the "failed" component then SG will assume that the error is a multi-bit uncorrectable error and signal an "uncorrectable error"

5.3.3.3 Data Poisoning in Memory

Data Poisoning in memory is defined as all zeroes in the codeword (32B0 except for the least significant bytes being 0xFF00FF. The Intel® 5400 chipset MCH poisons a memory location based on the events described in Table 5-6.

Table 5-6. Memory Poisoning Table

Event	Correctable Error	UnCorrectable Error
Normal Memory Read	Correct Data to be given register Intel® 5400 chipset MCH logs M17 error. (Correctable Non-Mirrored demand data ECC Error) Correct Data to be written back to memory	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 re-silver or spare copy data ECC error)	If error persistent 1. Log and Signal M11 Error (Non Aliased uncorrectable re-silver or 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 re-silver or spare copy data ECC error)	Re-use Read to memory and signal a M11 error. (Non-aliased uncorrectable re-silver or spare copy data ECC error). If error persistent 1. Poison the new memory image.

5.3.4 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 backpressure 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 Intel® 5400 chipset MCH can be completely scrubbed every day (estimate). Error logs include RAS/CAS/BANK/RANK. 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.

5.3.5 Demand Scrubbing

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.

5.3.6 x8 Correction

5.3.6.1 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. A conflicting read or write request pending issue will be held until the scrub is either completed or aborted because it was uncorrectable.

5.3.6.2 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. 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

5.3.7 Memory DIMM Pair/Rank isolation

For any ECC error flag set in the FERR_NF_FB-DIMM register, the FB-DIMMChan_indx field indicates the FB-DIMM branch that caused the error.

Possible values are:

- FB-DIMMChan_indx = 0 for a branch 0 ECC error
- FB-DIMMChan_indx = 2 for a branch 1 ECC error

For uncorrectable errors, BIOS uses the DIMMISO register's RANK, DIMM, and PERSISTENT fields to identify the failing DIMM.

MCH attempts uncorrectable error DIMM isolation by comparing the value of the initial response with the value of the retried response. The data from each channel is compared separately. If either channel's initial data miscompares with its retried data, then the DIMMISO.PERSISTENT configuration bit is cleared and the appropriate DIMMISO.DIMM configuration bits are set. If both channels' initial and retried data match, then the DIMMISO.PERSISTENT configuration bit is set because the failing DIMM cannot be identified. In single-channel mode, the DIMM can always be identified because only one channel is in operation. In mirrored mode, the DIMM should almost always be identifiable (outside of an extreme platform excursion) because the initial and retried data are returned from different branches. In non-mirrored lockstep mode, the DIMM should usually be identifiable (outside of an extreme platform excursion) because onset of a multiple-DRAM malfunction should be gradual, initially appearing as a transient failure (e.g. multiple data strobes "glitch" on the initial request, but operate normally on the retry).

For correctable errors, BIOS uses the RECMEMA.RANK and REDMEMB.ECC_Locator configuration fields to identify the failing DIMM. The ECC_Locator indicates which x8 device (or pair of adjacent x4 devices) caused the error. If any bit [8:0] is set, a DIMM on the even channel caused the error. If any bit [17:9] is set, a DIMM on the odd channel caused the error. REDMEMA.SYNDROME can be used to determine specific x8 bits in error, but this is not required by BIOS.

After a mirrored branch is off-lined, BIOS could execute AMB MemBIST on the suspicious DIMM-pair to reproduce failures. This can be performed out-of-band through MCH SPD bus.

Figure 5-6. Non-Redundant Memory Read Recovery with All NB Lanes Operational

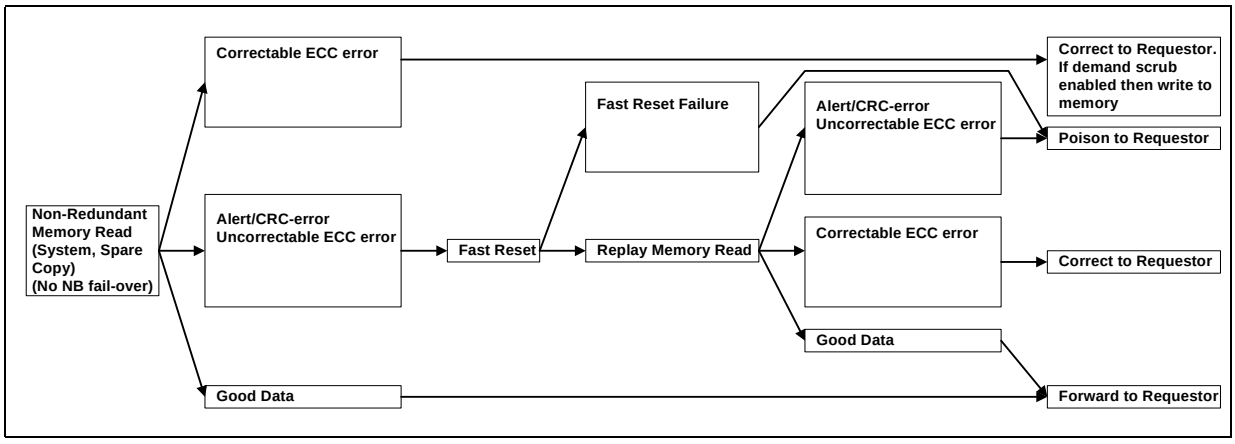


Figure 5-7. Non-Redundant Memory Read Recovery with a Faulty NB Lane

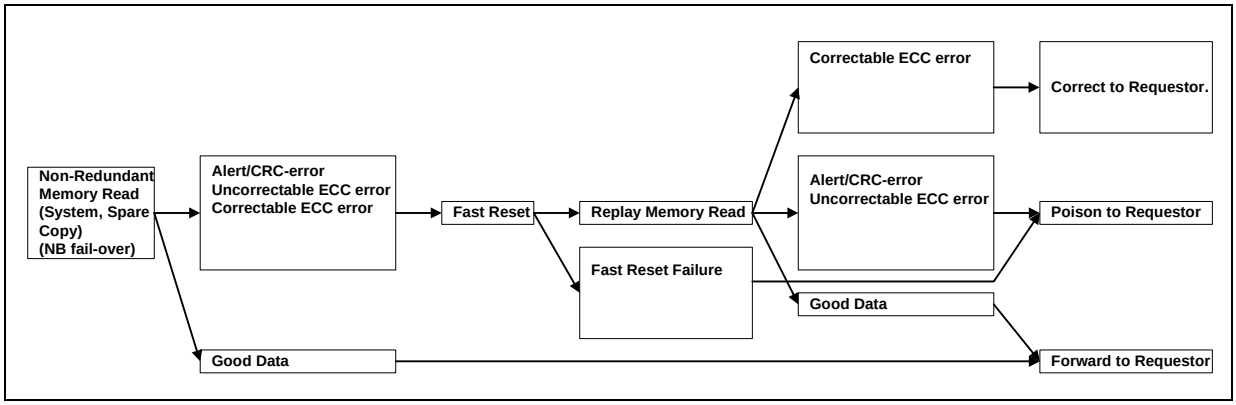


Figure 5-8. Configuration or Non-Redundant Memory Write Recovery

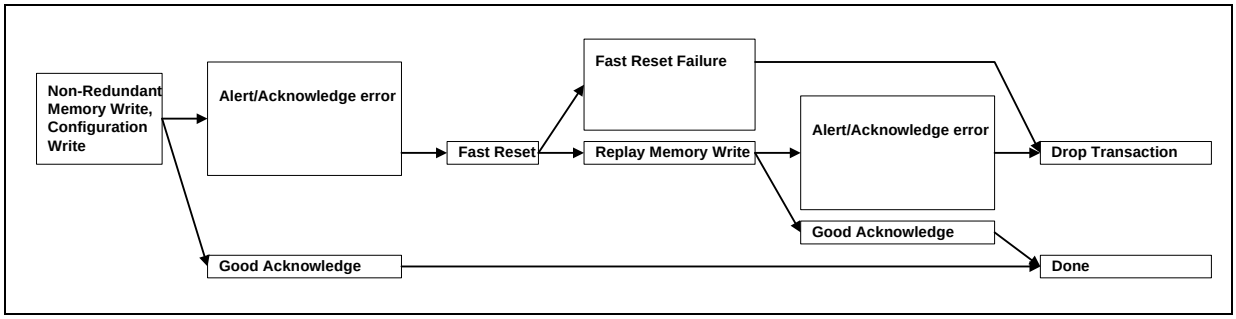


Figure 5-9. Memory Patrol Scrub Recovery

Figure 5-10. FB-DIMM Configuration Read Recovery

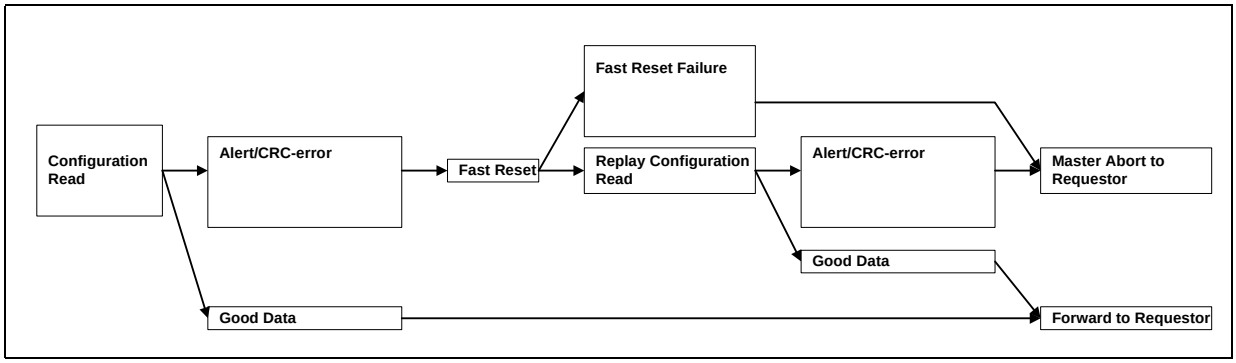


Figure 5-11. First Redundant Memory Read Recovery with All NB Lanes Operational

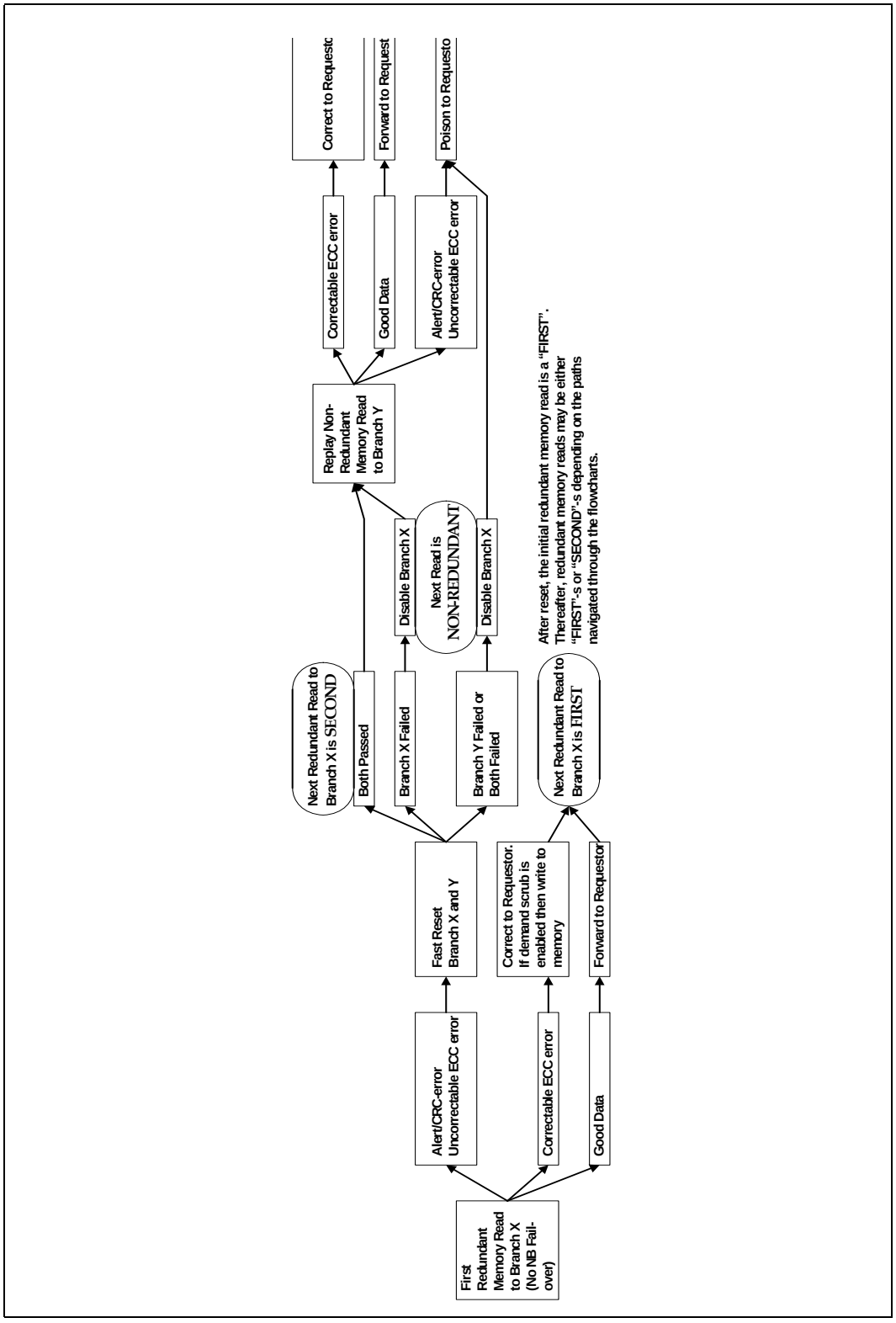


Figure 5-12. First Redundant Memory Read Recovery with a Faulty NB Lane

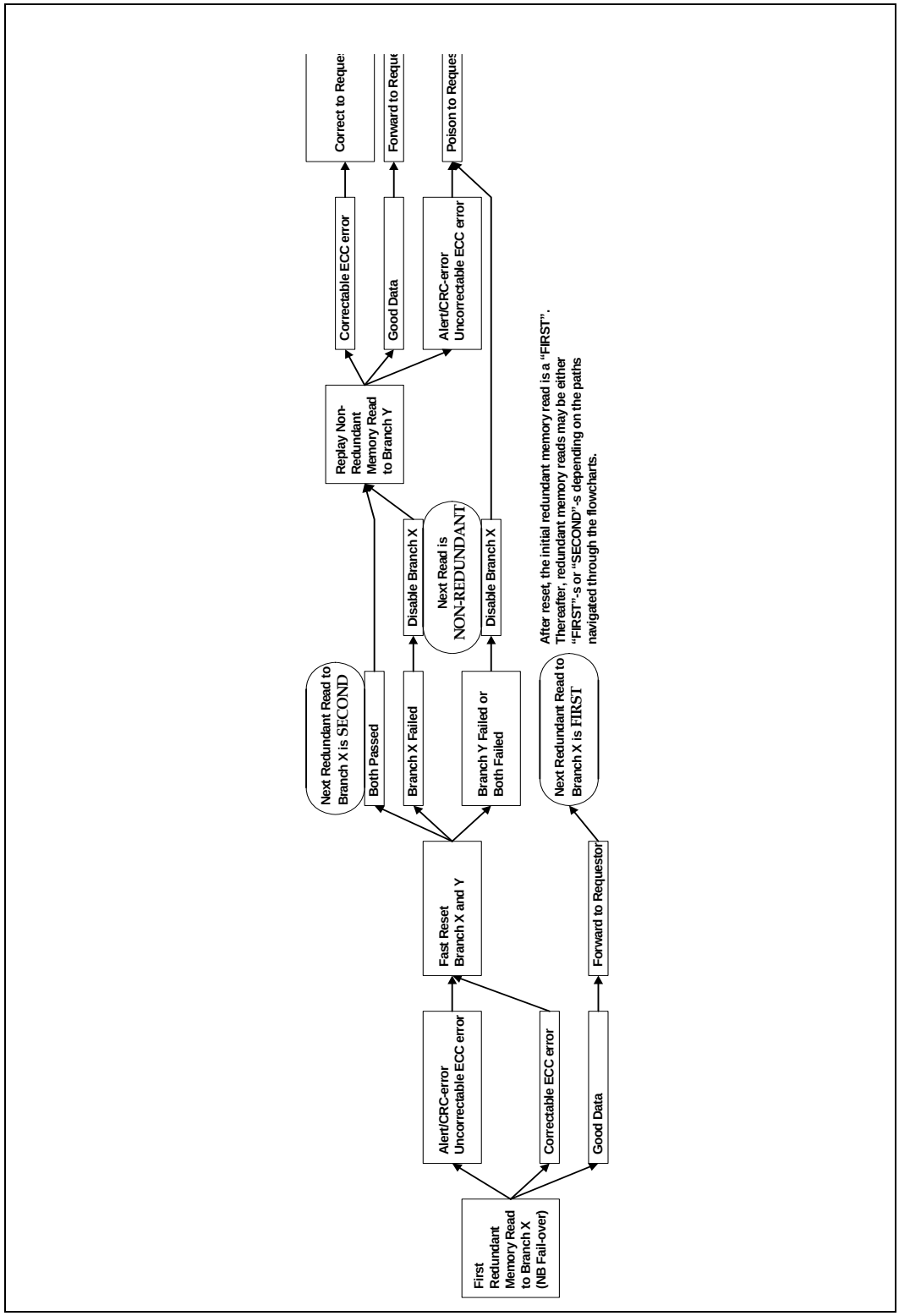


Figure 5-13. Second Redundant Memory Read Recovery with All NB Lanes Operational

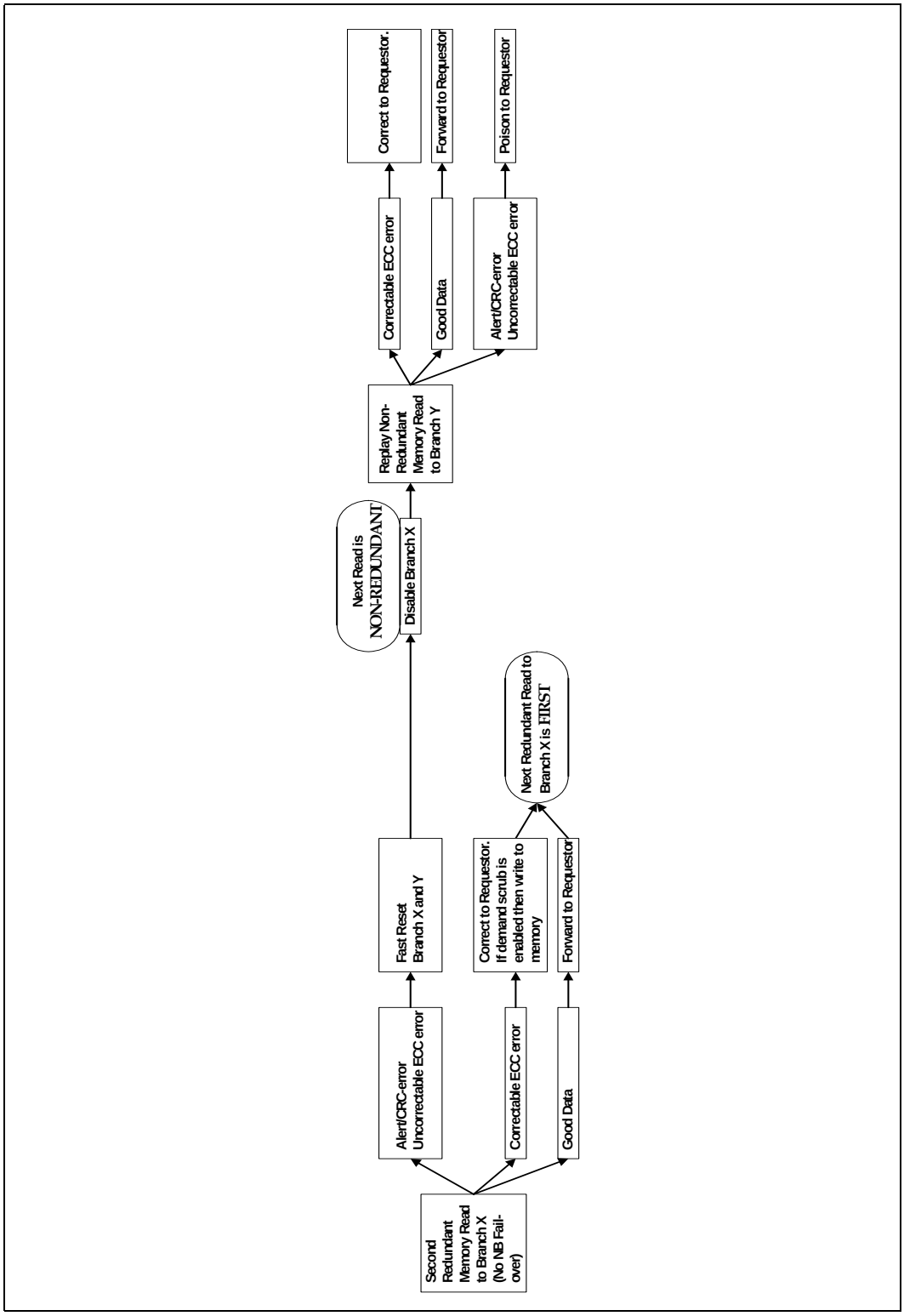


Figure 5-14. Second Redundant Memory Read Recovery with a Faulty NB Lane

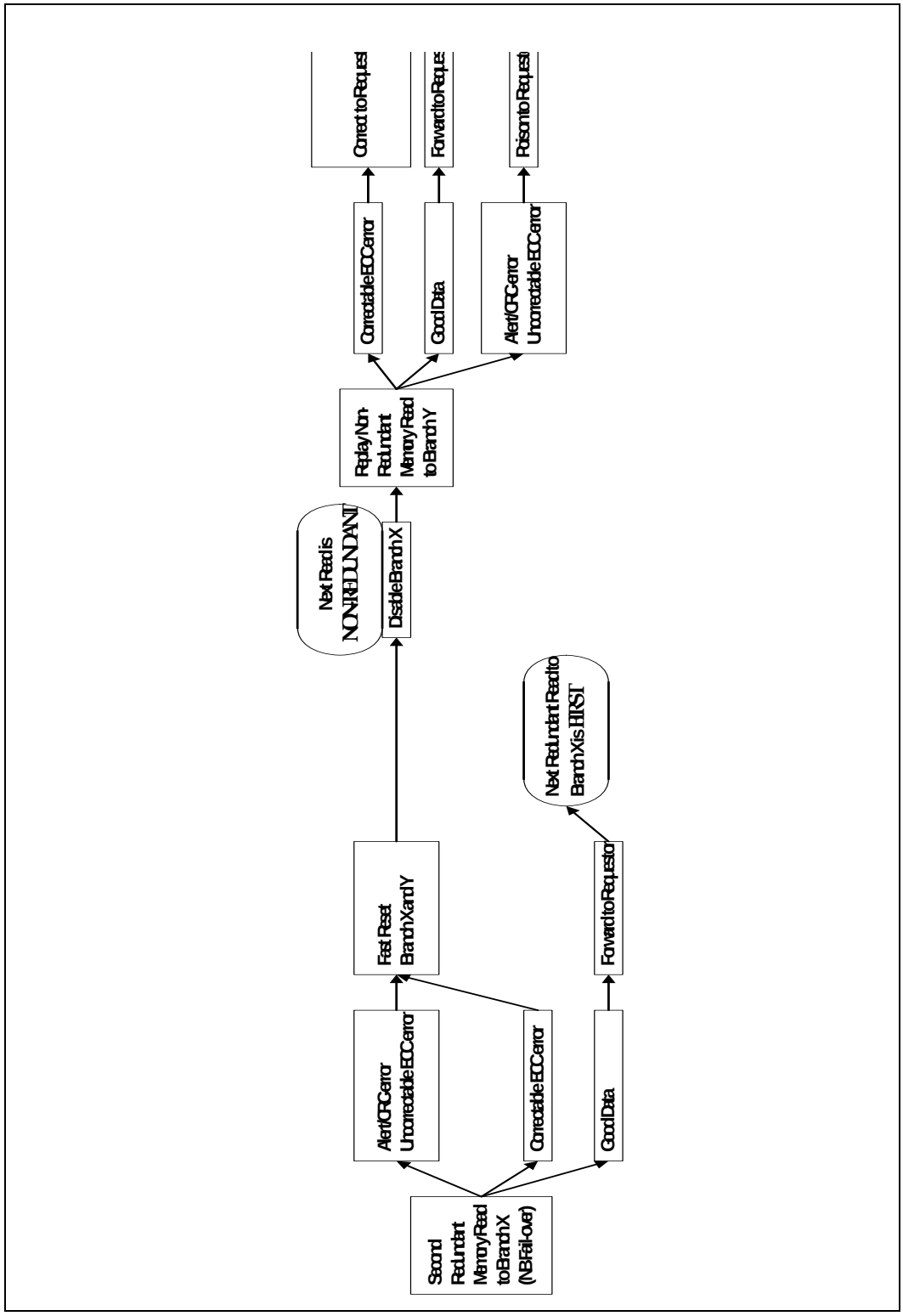
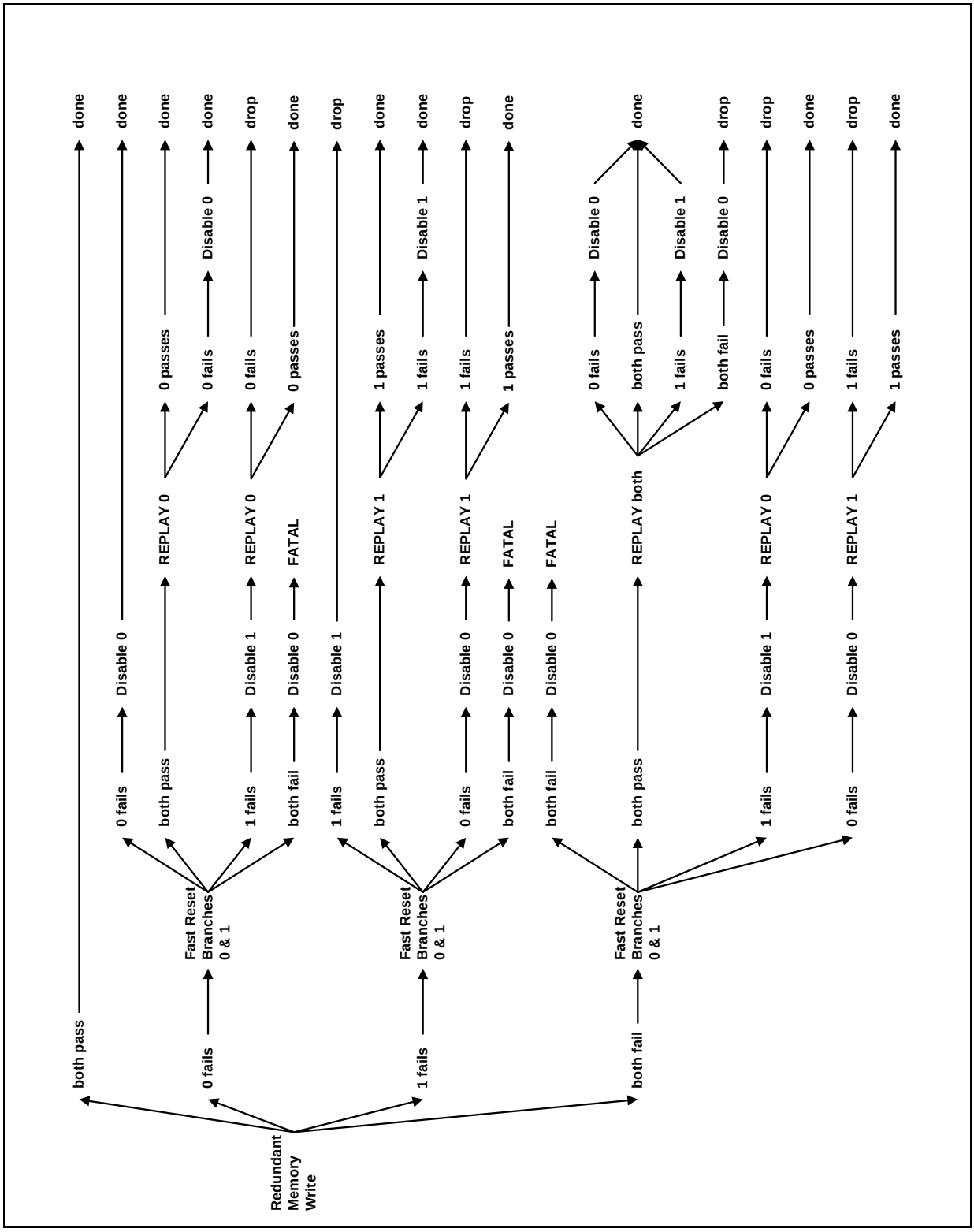


Figure 5-15. Redundant Memory Write Recovery



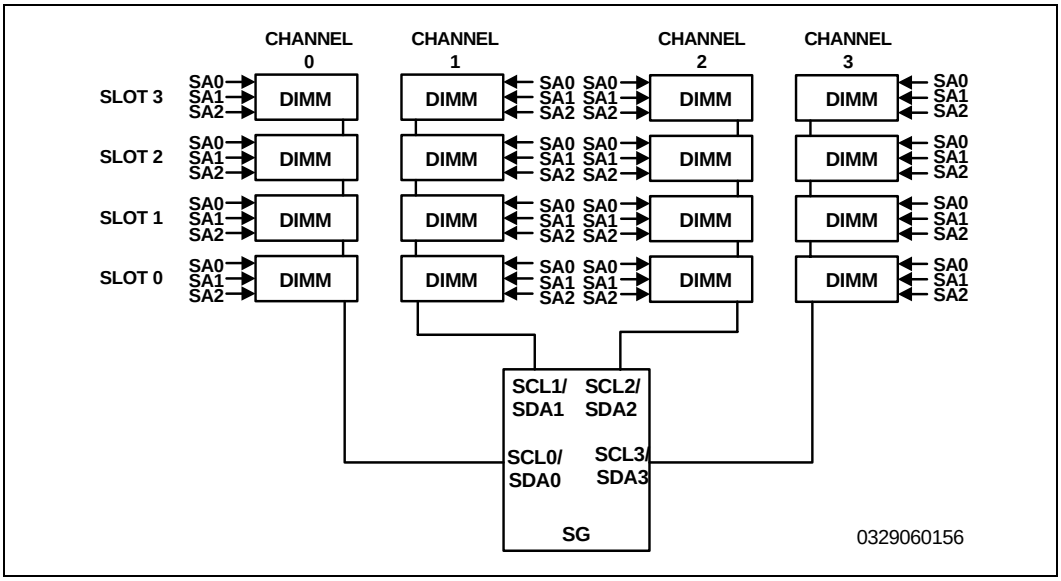
5.3.8 FB-DIMM Memory Configuration Mechanism

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 are 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 SPDxSMBCLK, SPDxSMBDATA pins on the MCH, see Figure 5-16. The x in SPDxSMB... denotes the SPD port number.

The 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 four SPD EEPROM's associated with each SPD bus. Therefore, the SPD interface is wired as indicated in Figure 5-16.

Figure 5-16. Connection of DIMM Serial I/O Signals



Board layout must map chip selects to SPD Slave Addresses as shown in Table 5-7.

Table 5-7. SPD Addressing (Sheet 1 of 2)

SPD Bus	FB-DIMM Channel	SLOT	Slave Address
0	0	0	0
		1	1
		2	2
		3	3

Table 5-7. SPD Addressing (Sheet 2 of 2)

SPD Bus	FB-DIMM Channel	SLOT	Slave Address
1	1	0	0
		1	1
		2	2
		3	3
2	2	0	0
		1	1
		2	2
		3	3
3	3	0	0
		1	1
		2	2
		3	3

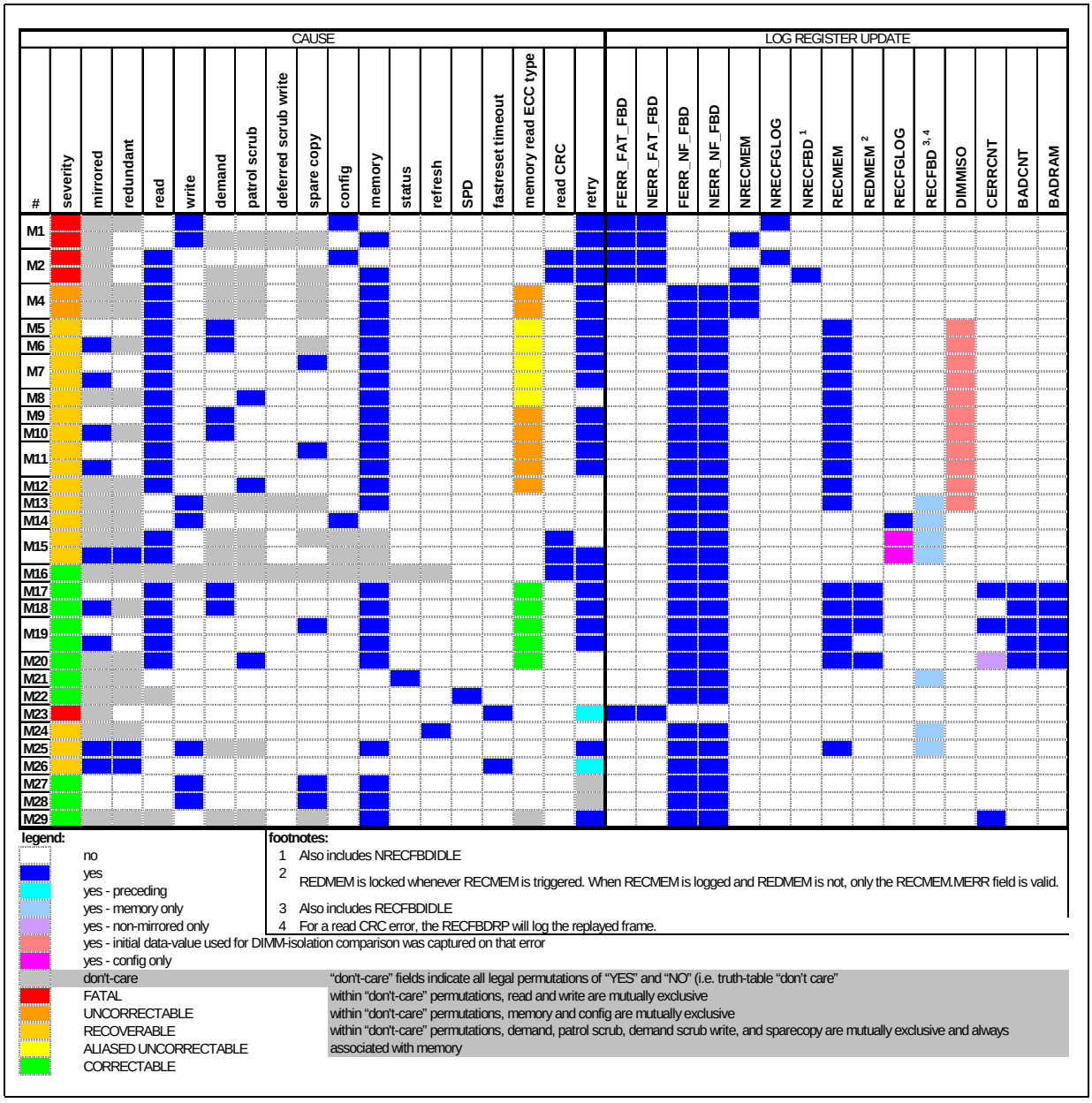
5.3.9 FB-DIMM Memory Failure Isolation Mechanisms

The MCH supports FB-DIMM fail-over mode and CRC generation on Northbound data. Successful transaction completion is signalled by the absence of alerts within a read round-trip. Bad CRC accompanies alerts. Alerts preempt read data. Detection of corrupted CRC or corrupted write acknowledge (idle) if enabled will initiate a FB-DIMM fail-over to the secondary data signal. In this case the A consecutive CRC/ack failure on the same transaction is fatal.

5.3.9.1 FB-DIMM Configuration Read Error

An erroneous configuration read return will be master aborted and return all 1’s. It will not be retried.

Figure 5-17. FB-DIMM Error Table



5.3.9.2 DIMM Failure Isolation

The failing DIMM may be isolated using information contained in several registers. ECC error flag bits are recorded in register FERR_NF_FB-DIMM, Section 3.9.22.3. This register records various error sources related to FB-DIMM memory transactions. When an error occurs the channel/branch information is recorded in the FB-DIMMChan_indx field.

The FB-DIMMChan_indx is a two bit field that records branch ECC errors. ECC errors are reported on a per branch basis (the LSB of this field has no relevance for ECC errors). For ECC errors the possible values for this field are:

FB-DIMMChan_indx = 0 Branch 0 ECC error

FB-DIMMChan_indx = 2 Branch 1 ECC error

Once the branch is determined the failing DIMM is determined, the rank and DIMM is determined from the RECMEMA.RANK and REDMEMB.ECC_Locator fields. The ECC_Locator indicates which x8 SDRAM device (or pair of adjacent x4 devices) caused the error. If any of the bits [8:0] is set, a DIMM on the even channel caused the error. If any of the bits [17:9] is set, a DIMM on the odd channel caused the error. See Table 3-50.

For uncorrectable errors the NRECMEMA.RANK register is used to identify the failing DIMM pair (lockstep channels).

5.3.9.3 ECC Code

When branches operate in dual-channel mode, the MCH supports the 18 device DRAM failure correction code (SDDC aka SECC) option for FB-DIMM. 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 2 wire faults on the DIMMs. This includes any pair of single bit errors.
- Detection of all permutations of 2 x4 DRAM failures.

When the branch operates in single-channel mode, the MCH supports an 8-byte-over-32-byte Single Error Correct Double Error Detect, SECDED+ code. It is the same ECC code used in the dual-channel mode, but the number of devices over which the codeword is defined is half, thereby reducing the SDDC properties to SECDED+. A single wire fault in the same device is replicated across all symbols reducing the effectiveness of the error correction. In the single DIMM mode (e.g. nine x8 devices), single wire fault (stuck at) errors or permanent full device faults can not be corrected. As applied by Intel® 5400 chipset, this code has the following properties:

- Detection of any two bits in error within 8 bytes of data
- Correction of any single bit in error within 8 bytes of data
- Correction to any single DRAM data bus wire failure

5.3.9.4 Inbound ECC Code Layout for Dual-Channel 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 5-18 illustrates the ECC code layout for branch 0. The figure shows how the symbols are mapped on the FB-DIMM 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 FB-DIMM channel. The layout for branch 1 is the same.

The bits of Data Symbol 0 (DS0) are traced from DRAM to FB-DIMM 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 FB-DIMM 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 5-18. Code Layout for Single-Channel Branches

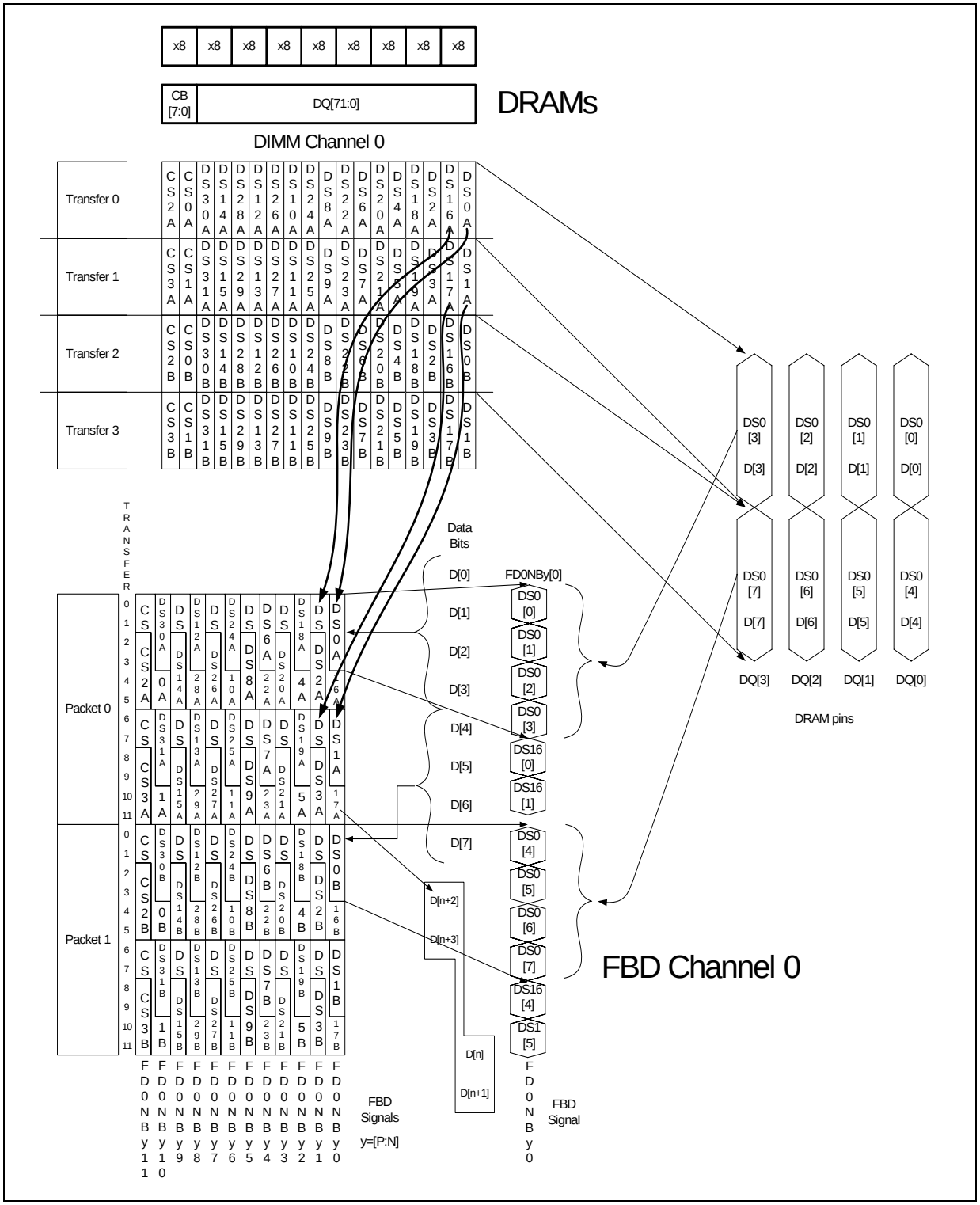
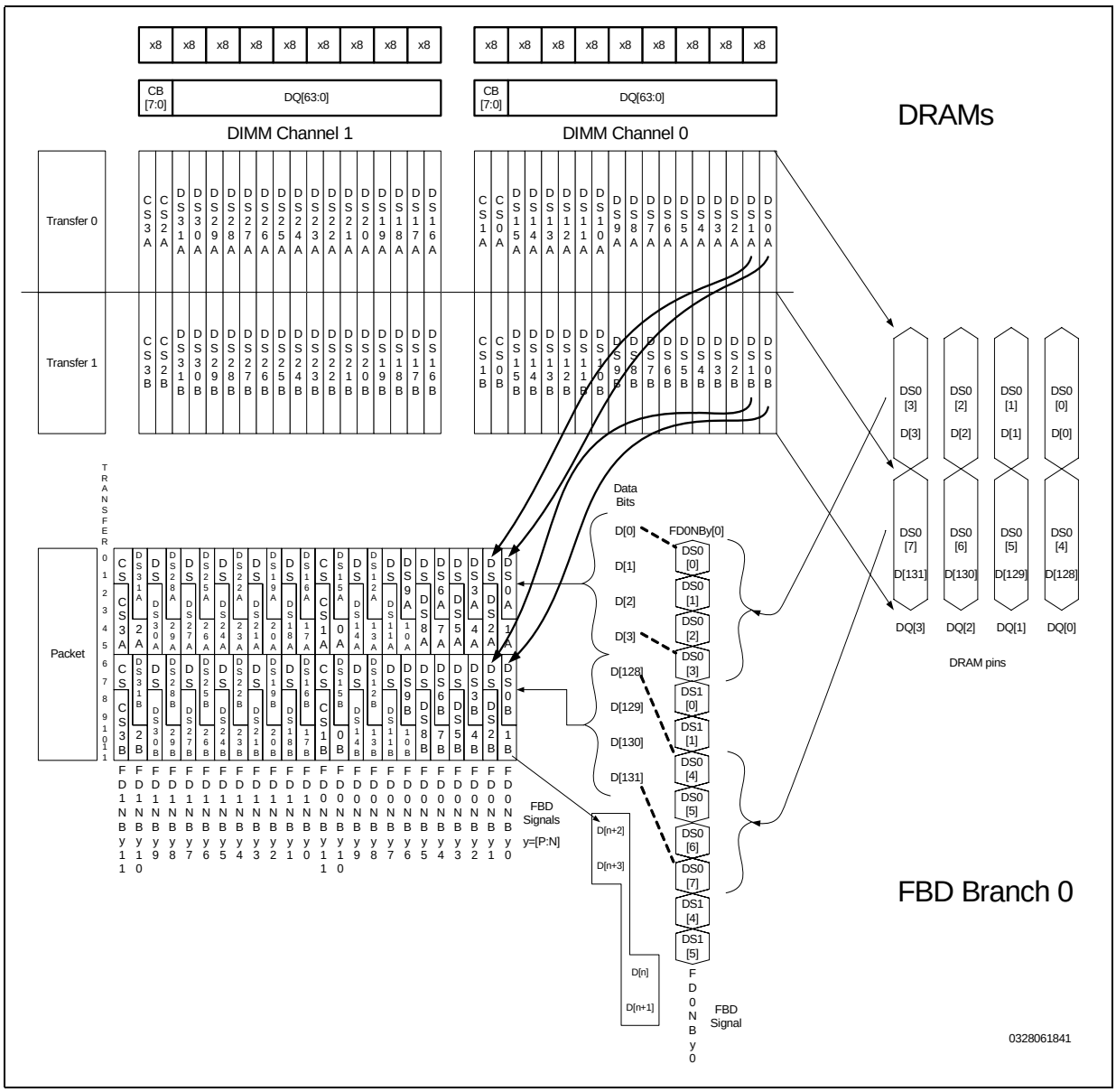


Figure 5-19. Code Layout for Dual-Channel Branches



5.3.9.5 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.

5.3.10 DDR2 Protocol

5.3.10.1 Posted CAS

Posted CAS timing is used, which requires that $t_{AL} = t_{RCD} - 1$.

5.3.10.2 Refresh

Regardless of the number of DIMMs installed, each rank will get a minimum of one refresh every period defined by the DRT.TREF configuration register field. The refreshes cycle through all (8 or 16) DIMM ranks.

The DIMM enters self-refresh mode during an FB-DIMM fast reset.

5.3.10.3 Access Size

All memory accesses are 64B.

5.3.10.4 Transfer Mode

For lockstep operation, 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.

For single-channel operation, the DIMM is programmed to use a burst-length of 64 bytes (8 transfers) across the channel. The Mode Register of the DIMM must be programmed for a burst length of 8, and interleave mode.

5.3.10.5 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.

5.3.11 FB-DIMM Memory Thermal Management

The Intel® 5400 chipset memory controller implements an adaptive throttling methodology 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. Closed Loop Thermal Activate Throttle Control when the temperature of the FB-DIMMs increases beyond a certain threshold.
   b. Open Loop Global Activate Control to limit requests when the number of activates crosses an event threshold events in large time window
2. Electrical Throttling to prevent silent data corruption by limiting the number of activates per rank in a small sliding window

5.3.11.1 Closed Loop Thermal Activate Throttle Control

Closed loop thermal activate throttling control uses the temperature of the FB-DIMM temperature sensor located in the AMB to determine when to throttle. FB-DIMM (AMB temperature) is returned each sync packet. A thermal throttle period is defined as window consisting of 1,344 cycles (42*32). The throttling logic in the memory controller uses this information to limit the number of activates to any DIMM within a throttling window based on temperature threshold crossing algorithm described later.

Every 42 frames the host controller is required to send a sync[1] packet, which returns a status packet from the AMBs along with temperature information. The AMB component has two temperature threshold points, T_{low} (programmed into the GB.TEMPLO register, a.k.a. T1) and T_{mid} (programmed into the GB.TEMPMID register, a.k.a. T2), and the current temperature of the GB with respect to these thresholds are returned in the status packet. In addition, the sync and status packets guarantees that enough transitions occur on each lane to maintain proper bit lock.

1. The sync packet may be dispatched by the MCH at an interval less than 42 frames depending on the gear ratios, timing, circuit and other parameters. For example, in the case of the core running at 266 MHz and the DDR2 clock at 333 MHz (4:5), the MCH can send a SYNC every 32@266 MHz=40 DDR2 333 MHz clocks. This meets the minimum 42 clock requirement of the FB-DIMM protocol for sync packet generation frequency.

In each 42 frame period:

Frames 1-40 are used for normal DRAM traffic: The A slot for DRAM commands and B/C for write data, as necessary.

Frame 41 is used for configuration commands: If a configuration read, it will appear in the A slot. If a configuration write it will appear in B/C (which is the only choice).

Frame 42 is used by the Sync packet and occupies the A, B, and C slots.

FB-DIMM thermal information is returned in the Sync packet as an encoded 2 bit field. The encoding of this field described in the following table.

Table 5-8. AMB Thermal Status Bit Definitions

S[2:1]	**Thermal Trip:** This field indicates various thermal conditions of the AMB as follows:
00	Below TEMPLO
01	Above TEMPLO
10	Above TEMPMID and falling
11	Above TEMPMID and rising

This data is a duplication of the contents of the AMB.FBDS0 register discussed in the Intel® 6400/6402 Advanced Memory Buffer External Design Specification (EDS), Revision 2.0. These 2 bits are returned for each AMB during in the Status packet.

The TEMPLO threshold is generally used to inform the host to accelerate refresh events. The TEMPMID threshold is generally used to inform the host that a thermal limit has been exceeded and that thermal throttling is needed.

There are separate counters associated with each of the 2 lockstep FB-DIMM pairs in a given branch (one counter per FB-DIMM pair per branch). When any of the counters reaches its limit (as specified by the THRTSTS.THRMTHRT register field for a given branch), 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, to allow the page to close.

5.3.11.2 Sequence of Actions During Throttling

When throttling begins during a given throttling window, the following actions take place:

Stop new DRAM commands

Wait "X" clocks for DRAM commands to complete. Where "X" is the worst case delay as defined below

Assert CKE low

Wait for throttling window to expire

Just before end of activation throttle window (about 3 clocks before for the CKE setup), Assert CKE high

Once the branch has been throttled, the memory controller sends a broadcast CKE for each DIMM command to take the CKE low on all DIMMs of the branch. This command is sent after the proper time has elapsed so that the outstanding transfers complete properly on the DRAMs. When activation throttling starts, CKE must not go low on the DRAMs until the last command has completed in the DRAMs. The worst case is an activate immediately followed by a posted CAS. A fixed time from the last command is used by the Intel 54000 Chipset MCH corresponding to the worst case delay (X) defined by

$$X = Max(worst\_case\_round\_trip\_delay, M * TRFC)$$

$$M = \begin{cases} 1.25 & \text{if Core to FBD clock ratio is 5:4} \\ 1 & \text{Otherwise} \end{cases}$$

with a suitable guard band[1] to protect any data loss. The TRFC parameter (Refresh to Activate Command delay) is factored into the equation since a refresh could be just underway when the last activate was about to be issued. The "1.25" scaling factor is to account for the 5:4 gearing ratio required for a FSB frequency of (333 MHz/400 MHz) and FBD/ DDR clock frequency of (266MHz)

During the time that CKE is low, no DRAM commands should be sent on the channel. However, Non-DRAM commands such as Configuration register and SYNC are required to be sent during this period.

When the throttle window is about to expire, a CKE command is sent to take all CKEs high. This must be done at least 3 clocks before the first command.

1. The worst case round trip delay is expected to be in the 10-20 clock range for a posted CAS command and the Intel 5400 chipset MCH RTL can be microarchitected appropriately.

5.3.11.3 CKE State Near End of Activation Throttling Window

If the throttling begins very close to the end of the window, then the assertion of CKE low command would be delayed beyond the end of the throttle window. To prevent this occurrence, the memory controller logic does not observe a throttle event in the last few clocks of the window, or assert a CKE low command.

If the activation throttle is set to begin within Y clocks before the end of the window, the memory controller skips the asserting CKE low step, where Y is X + 6[1] (and the number "6" is derived from 3 clocks for the CKE low to high minimum, plus another 3 clocks for the CKE high until first command after the throttling window.

5.3.11.4 Refresh Handling During Throttling

The MCH ensures that refreshes, which are lost during the activation throttle period (possibly up to 2), are made up at the end of the period. Double refresh rates to the DIMMs should be carried out when needed regardless of the setting of the MC.THRMHUNT bit. This is particularly important for open loop throttling when the temperature could rise beyond 85'C.

5.3.11.5 Throttling Parameters for Activation Throttling.

The current throttling parameters for each branch are stored in the THRMTHRT register field defined in Section 3.9.4. All activation throttling parameters in the THRMTHRT registers are 8-bits wide, and provide increments of 4 activations per throttle window (1344 clocks). Three levels of throttling limits are defined.

- THRTLOW: A base throttling level that is applied when the temperature is in the low range (below T_{low}) and the THRTSTS.GBLTHRT*[2] bit is not set by the Global Throttling Window logic. See Section 3.9.4
- THRTMID: A mid level throttling level that 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 Global Throttling Window logic. See Section 3.9.9.
- THRTHI: The highest level of throttling. When MC.THRMODE=1, this level is applied whenever the temperature is above T_{mid}. When MC.THRMODE=0, this level is the ceiling of the hunting algorithm of the closed loop throttling. The temperature being above T_{mid} has priority over the Global Throttling Window throttling (the higher throttling level takes precedence). See Section 3.9.13.

The MC.THRMHUNT bit must be enabled for the temperature to have any influence on the throttle parameters. If MC.THRMHUNT=0, only the GBLTHRT bit from the Global Throttle Window, when enabled can change the THRMTHRT register field.

1. Intel 5400 chipset MC design needs to adjust the value based on the latest JEDEC recommendation for CKE low to high transition.
2. GBLTHRT* is an internal combinatorial signal before it is latched in the THRSTS.GBLTHRT register field to enable the open loop throttling logic to use the latest value of the signal.

Figure 5-20. Thermal Throttling with THRMHUNT=1

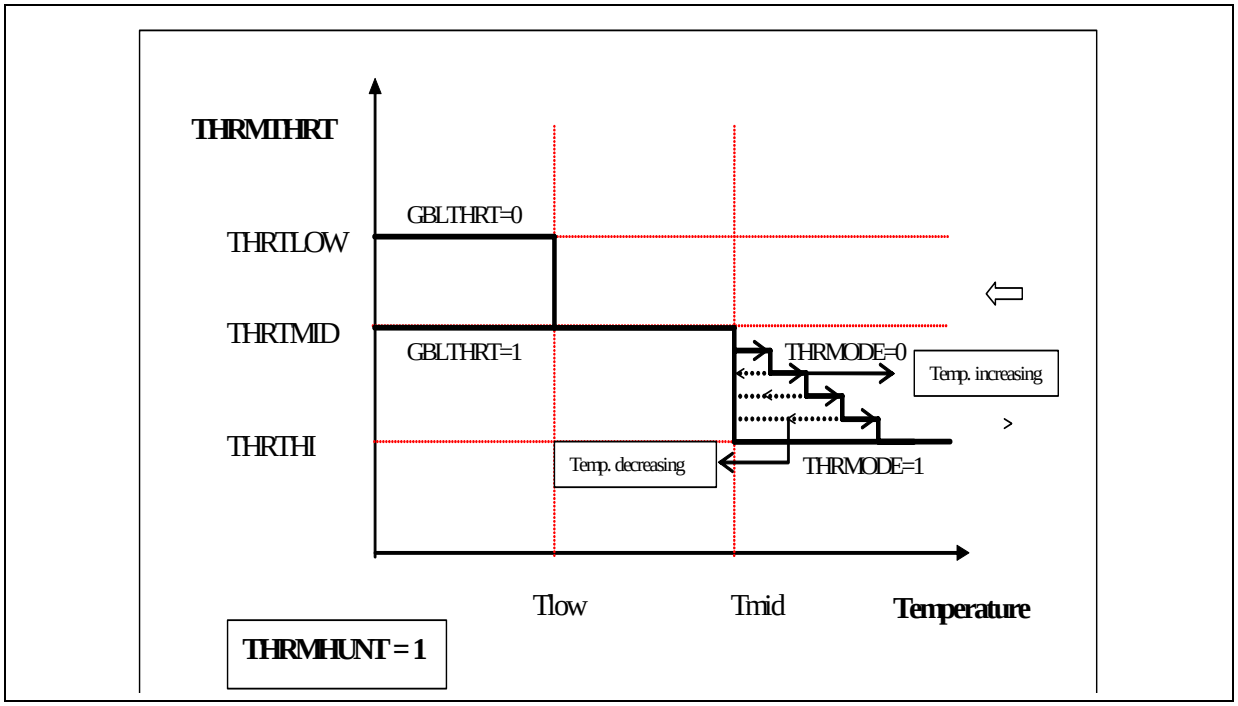
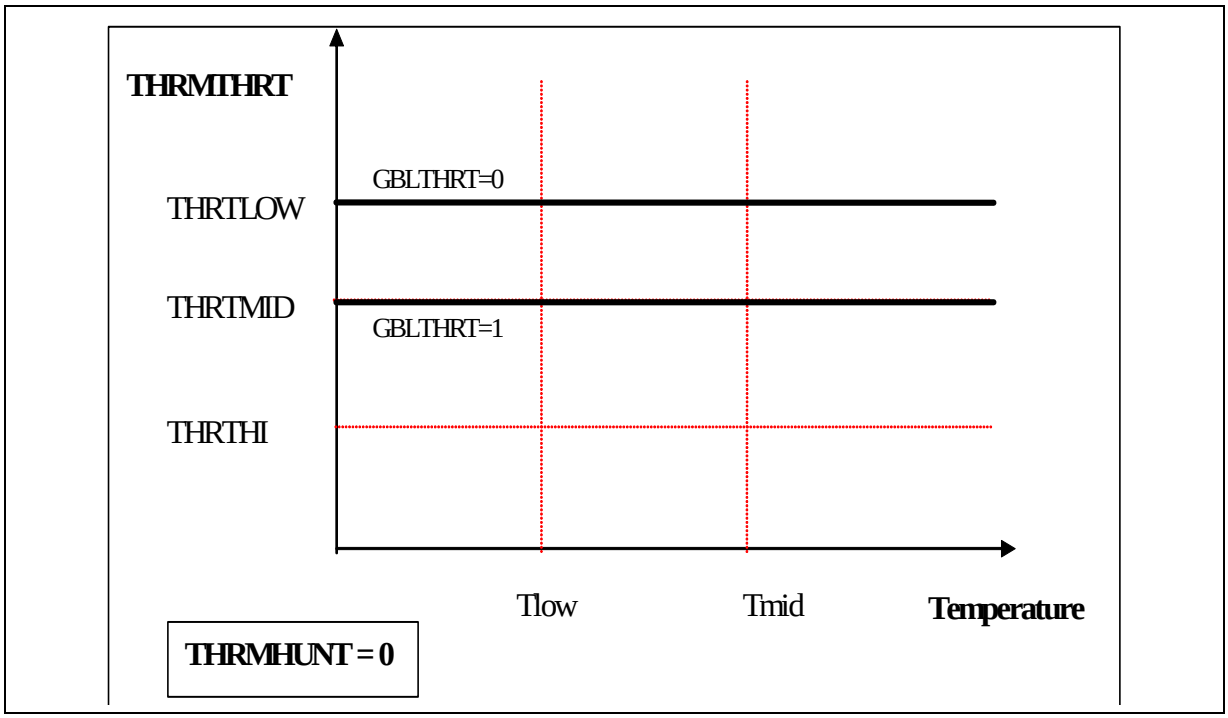


Figure 5-21. Thermal Throttling with THRMHUNT=0



5.3.11.6 Closed Loop Activation Throttling Policy

Individual DIMMs flag their thermal state in the FB-DIMM status return. When the MC.THRMHUNT configuration bit is set, memory reads and writes (summed together) will be regulated on a per-DIMM-pair basis according to the following algorithm described in Figure 5-22.

Intel 5400 chipset MCH provides a greater degree of thermal throttling if there is a sudden temperature spike between from T_{low} to T_{mid} by setting the THRMTHRT register to THRTMID as a starting point when MC.THRMODE=0.

Once this point is reached, if temperature increased further during the next global window, then THRMTHRT register will be adjusted by the equation THRMTHRT= MAX(THRMTHRT -2, THRTHI). See staircase effect in Figure 5-20. If temperature decreased but is still greater than T_{mid}, then the THRMTHRT will retain its last value. This provides some degree of hysteresis control to allow the DIMMs to cool further before THRMTHRT jumps back to a larger number (i.e less throttling) at the junction when the temperature reached T_{mid}. Refer to the dotted line in Figure 5-20. This scheme helps in reducing the thermal power by limiting the number of activates. See Figure 5-22 for further details.

1. Staircase Conditioning [THRTCTRL.THRMODE=0]: This method is employed when THRTCTRL.THRMODE=0 and temperature crosses above Tmid. The THRMTHRT registries capped to THRTMID (starting point) and it uses a linearly increasing (less aggressive) throttling algorithm to reduce activations and balance performance and power envelope when temperature rises and falls around Tmid point. Once THRTMID is reached, if temperature increases further during the next global window, then THRMTHRT register will be adjusted by the equation THRMTHRT= MAX(THRMTHRT -2, THRTHI). This produces the staircase effect as shown in Figure 5-20, “Thermal Throttling with THRMHUNT=1” .
   — If temperature decreases subsequently but is still greater than Tmid, then the THRMTHRT will retain its last value. This provides some degree of hysteresis control to allow the DIMMs to cool further before THRMTHRT jumps back to a larger number (i.e less throttling) at the junction when the temperature reached Tmid. This scheme helps in reducing the thermal power by limiting the number of activates.
2. Step Conditioning (brute force) [THRTCTRL.THRMODE=1]: This method is employed when THRTCTRL.THRMODE=1 and temperature crosses Tmid. The THRMTHRT register is capped to THRTHI and it provides a greater degree of throttling by allowing fewer activates to the memory allowing the DIMM to cool down quicker but at the expense of performance. This can be used to control sudden temperature surges that moves the envelope from below Tlow to above Tmid. and stays there for a long period.

During the global window, the Intel 5400 chipset MCH will broadcast one configuration write to the DIMMs AMB.UPDATED registers. This write will not be re-played or re-sent.

Note: A channel fault could drop an AMB.UPDATED write. If the temperature increased during the previous global window, but had not actually increased during the current global window, then THRMTHRT will un-necessarily decrease. If the temperature had not increased during the previous global window, but had actually increased during the current global window, then THRMTRHT will remain unresponsive to the temperature increase for one global throttling window. The situation will rectify itself in the next global throttling window.

1. If there is a sudden temperature spike between from below Tlow to above Tmid by setting the THRMTHRT register to THRTMID as a starting point when THRTCTRL.THRMODE=0. If temperature rose from above Tlow to above Tmid, then the THRMTHRT will use THRTMID value if THRTCTRL.THRMODE=0; otherwise it will use THRTHI if THRTCTRL.THRMODE=1.

Figure 5-22. Thermal Throttling Activation Algorithm

```
THRMTHRT = THRTLOW (Initialize to base-level Activations)

if (Global_Timer expires)
{
    if (MC.GTW_MODE == 1)   // Choose window size based on mode setting
    {   Global_Timer = 4*1344       // Validation & Debug Mode
    }
    else
    {   Global_Timer = 0.65625*2^25 = 16384*1344
        // make global throttling window an integral multiple of the closed loop window
    }

    if (THRTCTRL.THRMHUNT == 1)
    {
        for (each DIMM-pair [m] on each branch [n])  /* m=0..3, n=0..1 */
        {     if (temperature of any DIMM[i]  >= Tmid) /* 0 <= i <= 3 */
              {     if (THRTCTRL.THRMODE == 0)
                          if (THRMTHRT > THRTMID)
                          {   /* This will cap the start  point to THRTMID if there is a  */
                              /*   spike in the GB Temperature from Tlow to Tmid & beyond */
                              /*   Provides better throttling and control */
                              THRMTHRT = THRTMID
                          }
                          else       /* Staircase roll down may happen for subsequent samplings
                          {    if (the temperature of any DIMM which is above Tmid, increased)
                               {     THRMTHRT = max(THRMTHRT - 2, THRTHI)
                               } /* Otherwise retain last THRMTHRT value */
                          } /* end of THRMTHRT > THRTMID check */
                    }
                    else
                    {     THRMTHRT = THRTHI
                    }  /* end of THRMODE==0 check */
              }
              else if ((temperature of any DIMM[i]  >= Tlow[i])
                                  &&  (temperature of all DIMM's[i]  < Tmid[i]))
              {     THRMTHRT = THRTMID
              }
              else if (temperature of all DIMM's[i] < Tlow[i]))
              {     if (GBLTHRT == 1)
                    {     THRMTHRT = THRTMID
                    }
                    else
                    {     THRMTHRT = THRTLOW
                    }
              }
        }
    }
    else
    {    if (GBLTHRT == 1)
         {     THRMTHRT = THRTMID
         }
          else
         {     THRMTHRT = THRTLOW
         }
    }
}
```

Figure 5-23. FB-DIMM Bandwidth as a Function of Closed Loop Thermal Throttling

THRMTHRT Reg Value	Activates	% BW allowed	BW per DIMM GB/s	sys BW, 1 DIMM/ch	sys BW 2 DIMM/ch	sys BW 4 DIMM/ch
0	unlimited					
1	4	0.60%	0.03	0.13	0.25	0.51
2	8	1.19%	0.06	0.25	0.51	1.02
3	12	1.79%	0.10	0.38	0.76	1.52
4	16	2.38%	0.13	0.51	1.02	2.03
5	20	2.98%	0.16	0.63	1.27	2.54
6	24	3.57%	0.19	0.76	1.52	3.05
7	28	4.17%	0.22	0.89	1.78	3.56
8	32	4.76%	0.25	1.02	2.03	4.06
12	48	7.14%	0.38	1.52	3.05	6.10
16	64	9.52%	0.51	2.03	4.06	8.13
20	80	11.90%	0.63	2.54	5.08	10.16
24	96	14.29%	0.76	3.05	6.10	12.19
28	112	16.67%	0.89	3.56	7.11	14.22
32	128	19.05%	1.02	4.06	8.13	16.25
36	144	21.43%	1.14	4.57	9.14	18.29
40	160	23.81%	1.27	5.08	10.16	20.32
44	176	26.19%	1.40	5.59	11.17	
48	192	28.57%	1.52	6.10	12.19	
64	256	38.10%	2.03	8.13	16.25	
72	288	42.86%	2.29	9.14	18.29	
80	320	47.62%	2.54	10.16	20.32	
96	384	57.14%	3.05	12.19		
128	512	76.19%	4.06	16.25		
144	576	85.71%	4.57	18.29		
160	640	95.24%	5.08	20.32		
168	672	100.00%	5.33	21.33		

5.3.11.7 Global Activation Throttling Software Usage

In practice, the throttle settings for THRTMID are likely to be set by software such that the memory controller throttle logic will actually prevent the GBLACT limit from being exceeded and the result will often be that such that THRTLOW is used for a Global Throttle Window, at which time, the GBLACT.GBLACTLM is exceeded, causing the MC s to use a larger throttling period THRTMID for 16 (or 2) global[1] windows. During each of those global windows, GBLACT limit is not exceeded, because the throttling will prevent it from being exceeded. After 16 (or 2) global[2] throttling windows, it switches back to THRTLOW, and on the next global window GBLACT is again exceeded, causing another 16 (or 2) windows[2]. Hence, we can get a cumulative pattern of 16,1,16,1 (or 2,1,2,1) global[2] throttling windows and this prevents excessive heat dissipation in the FB-DIMMs by prolonging the throttle period.

1. The 2 window Global Throttling count will be chosen if MC.GTW_MODE=1.

It should be mentioned that the open and closed loop throttling control policies implemented on the Intel 5400 chipset MCH uses the internal core clocks for the calculating the windows and not the DDR clocks. Thus any software/BIOS should take this into account for manipulating the THRMTHRT registers when dealing with different FB-DIMM technologies and speeds.

5.3.11.8 Dynamic Update of Thermal Throttling Registers

In general, the Intel 5400 chipset registers should not updated dynamically during runtime as it may interfere with the internal state machines not designed exclusively for such changes and could result in a system hang/lock up. This requirement is relaxed (subject to validation) for the Intel 5400 chipset thermal throttling registers where it is desirable for BIOS or special OEM software in BMC to exercise dynamic control on throttling for open/closed loop algorithm implementation. The following examples are some of the potential areas of this usage model where dynamic change is needed to balance performance and acoustic levels in the system

Fan control for CPU temperature related system acoustics or other BMC related operations. Limit rouge activity by increasing memory throttling via throttle register updates to condition the system based on some event (excessive bandwidth or CPU activity)

Fan failure/breakdown. When this occurs, temperature conditioning can be provided by reducing the activity level in the DIMMs to a certain threshold until the failed fan can be repaired by the technician and service restored to normalcy

5.3.11.9 General Software Usage Assumptions

Under normal circumstances, it is expected that there is no change of throttling values once it is configured by BIOS during boot. The external Fan control and the BIOS settings of the OEM via BMC would ensure adequate cooling and maintain the DIMMs within the prescribed tolerance limits of the TDP. However, situations such as thermal virus or fan fail down condition might warrant the BIOS/SW to take preemptive action in adjusting the throttling to say 40-70% of the normal mode before it is cleared. This means that changes to throttling registers can happen at random intervals (infrequent) and the platform should be able to tolerate any transients changes that may result when the Intel 5400 chipset is updated with the new throttle values. These requirements are captured below.

5.3.11.10 Dynamic Change Operation Requirements for Closed Loop Thermal Throttling (CLTT)

In addition to all the conditions/requirements as stipulated in Section 5.3.11.11, the closed loop throttling which uses GB temperature feedback to adjust the throttling levels requires the following:

- Intel 5400 chipset Registers that are affected by dynamic updated include THRTMID (T2), THRTLOW (T1), THRTHI and THRTCTRL.THRMHUNT. (THRMHUNT=1 selects the closed loop mode)
- •When Temperature crosses T_{mid}, the CLTT switched to either the staircase function (if THRMCTRL.THRTMODE=0) or the single step function (THRMCTRL.THRMODE=1) as depicted in the right side of Figure 5-20. See also Figure 5-22. Note that CLTT is NOT history-based algorithm except in the staircase mode which uses the old value. In this case, the staircase function that always decrements the old value of THRMTHRT by 2. By design THRMTHRT can never be below THRTHI. If the new THRTHI is greater than THRMTHRT, then the algorithm will reset THRMTHRT to THRTHI & the staircase function can no longer be used since the bottom (aka

ceiling) of THRTHI has already been reached as defined by the following equation extracted from Figure 5-22.

THRMTHRT = max(THRMTHRT - 2, THRTHI)

5.3.11.11 Disabling Closed Loop Throttling

The following registers in Intel® 5400 chipset can be initialized to disable throttling:

- THRTCTRL.THRMHUNT = 0 /* This forces the Intel® 5400 chipset to ignore temperature for closed loop */
- THRTHI.THRHILM = 0 (or 168d)
- THRTMID.THRMIDLM = 0 (or 168d)
- THRTLOW.THRLOWLM = 0 (or 168d)

Figure 5-24. Global Activation Throttling BW allocation as a function of GBLACTLM for a 163841344 window with MC.GTW_Mode=0 (normal)**

GBLACT. GBLACTLM	# of Activates	% BW allowed	BW per DIMM GB/s	sys BW, 1 DIMM/ch	sys BW 2 DIMM/ch	sys BW 4 DIMM/ch
0	unlimited					
1	65536	0.60%	0.03	0.13	0.25	0.51
2	131072	1.19%	0.06	0.25	0.51	1.02
3	196608	1.79%	0.10	0.38	0.76	1.52
4	262144	2.38%	0.13	0.51	1.02	2.03
5	327680	2.98%	0.16	0.63	1.27	2.54
6	393216	3.57%	0.19	0.76	1.52	3.05
7	458752	4.17%	0.22	0.89	1.78	3.56
8	524288	4.76%	0.25	1.02	2.03	4.06
12	786432	7.14%	0.38	1.52	3.05	6.10
16	1048576	9.52%	0.51	2.03	4.06	8.13
20	1310720	11.90%	0.63	2.54	5.08	10.16
24	1572864	14.29%	0.76	3.05	6.10	12.19
28	1835008	16.67%	0.89	3.56	7.11	14.22
32	2097152	19.05%	1.02	4.06	8.13	16.25
36	2359296	21.43%	1.14	4.57	9.14	18.29
40	2621440	23.81%	1.27	5.08	10.16	20.32
44	2883584	26.19%	1.40	5.59	11.17	
48	3145728	28.57%	1.52	6.10	12.19	
64	4194304	38.10%	2.03	8.13	16.25	
72	4718592	42.86%	2.29	9.14	18.29	
80	5242880	47.62%	2.54	10.16	20.32	
96	6291456	57.14%	3.05	12.19		
128	8388608	76.19%	4.06	16.25		
144	9437184	85.71%	4.57	18.29		
160	10485760	95.24%	5.08	20.32		
168	11010048	100.00%	5.33	21.33		

5.3.12 Electrical Throttling

Electrical throttling is a mechanism that limits the number of activates (burstiness) 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 3.9.23.7. 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 per rank electrical throttling for FB-DIMM is 4 activates per 37.5ns window (JEDEC consensus) and is summarized in Table 5-9 for various DIMM technologies.

Table 5-9. Electrical Throttle Window as a Function of DIMM Technology

DIMM Modes	Intel 54000 Chipset MCH Core: FB-DIMM clock Ratio	Electrical Throttle Window[1] (in core clocks per rank per DIMM pair per branch)
DDR533	1:1	10
	5:4	13
DDR667	1:1	13
	4:5	13 (conservative)
DDR800[2]	1:1	15
Conservative (safe mode)	All	20

Notes:

1. Maximum 4 activates per rank is allowed within the window.
2. This is not a supported technology for Intel 5400 chipset MCH and is tabulated for information/illustrative purposes only.

The MC.ETHROT configuration register field limits the number of activations per sliding electrical throttle window. The memory controller logic can implement the sliding electrical throttle window with a 20-bit shift register per rank in each DIMM pair per branch. This register records for the last 20 clocks, whether an activate was issued or not to that rank. The number of activates can then be summed up from the state of the shift register and compared with the respective limit as shown in Figure 5-21. If the limit is reached, then further activates to the rank are blocked until the count falls below the limit. The Electrical throttling logic in the MC masks off the end bits for the DIMM technologies that require fewer clocks. As an example, if the DIMM technology used is DDR667, then it can allow 4 activates within the last 13 clocks, the remaining 7 bits are masked (forced to 0) so they do not prevent activates.

5.4 Intel 5400 Chipset Behavior on Overtemp State in AMB

Overtemperature occurring in an AMB may lead to data corruption in the chipset.

- If EI is received by Intel 5400 chipset due to Overtemp detection in one of the AMBs, Intel 5400 chipset will capture random data that most likely will be interpreted as having a CRC or uncorrectable ECC error causing the link to go into a fast reset loop without data corruption.
- If the EI is interpreted as having both good CRC and good ECC, this could cause data corruption until a bad CRC/ECC frame is detected and the link enters the fast reset loop.

Note: An all 0 frame fits this case of good CRC and ECC. This is just as unlikely as any other random frame contents when interpreting EI.

5.5 Interrupts

The MCH 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 and/or the MCH of the chipset.

The legacy 8259 functionality is embedded in the Intel 631xESB/632xESB I/O Controller Hub component. The chipset 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. For details regarding the XAPIC interrupt architectures see the *RS - XAPIC External Architecture Specification, Revision 1.1*.

5.5.1 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 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, also known as APIC messages. 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).

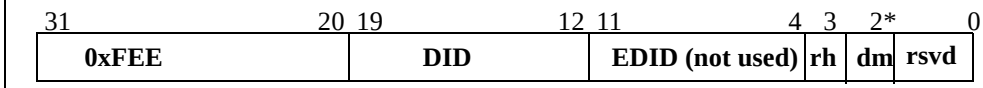
The XAPIC architecture provides for lowest priority delivery, through interrupt redirection by the chipset. If the redirectable hint bit is set in the XAPIC message, the chipset may redirect the interrupt to another processor. Note that redirection of interrupts can be to any processor on either Processor 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 5.5.4.

5.5.1.1 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 5-25 shows the address definition in IA32 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 5-25 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 5-25. 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 5-10. XAPIC Data Encoding

D[31:16]	D[15]	D[14]	D[13:11]	D[10:8]	D[7:0]
x	Trigger Mode	Delivery Status	x	Delivery Mode	Vector

5.5.2 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.

5.5.2.1 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 SG will drive A[12:11] to '00 for FSB0, and '01 for FSB1.

The SG will not rely on the cluster ID or any other fields in the APIC ID to route interrupts. The SG will ensure the interrupt is seen on both 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.

5.5.2.2 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). Processors used with the MCH operate in flat 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. The SG also has an equivalent bit in the External Task Priority Register (XTPR0) to indicate flat or cluster mode. This is set to flat mode by reset and must not be changed, since the processors used with the MCH operate in flat mode only.

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.

5.5.3 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 both 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. The MCH also has an embedded interrupt controller which will be discuss separately. Table 5-11 shows the routing rules used for routing XAPIC messages in a MCH platform. This table is valid for both broadcast and non-broadcast interrupts.

Table 5-11. 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 "Interrupt Redirection" on page 352) is performed by the MCH and is delivered to both FSBs.

5.5.4 Interrupt Redirection

The XAPIC architecture provides for lowest priority delivery through interrupt redirection by the SG. 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.

5.5.5 EOI

For XPF platforms using XAPIC, the EOI is a specially encoded processor bus transaction with the interrupt vector attached. Since the EOI is not directed, the SG 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.

Note: The MCH will translate the EOI on the FSB into an EOI TLP message type on the PCI Express/ESI ports.

5.5.6 I/O Interrupts

The Intel 631xESB/632xESB I/O Controller Hub components receive I/O interrupts with either dedicated interrupt pins or with writes to the integrated redirection table. The I/OxAPIC controller integrated within these components turns these interrupts into writes (following the format mentioned in Section 5.5.1.1) destined for the processor bus with a specific address. The MCH I/OxAPIC has dedicated interrupt virtual pins, but does not support writes to the integrated redirection table.

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 5.5.1.1. 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. To accomplish this, the PXH.MBAR must be programmed within the PCI Express device MMIO region.

Similarly, the integrated I/OxAPIC can be programmed through the MBAR memory region or the ABAR memory region. Refer to Section 4.3.7 for its assigned range.

5.5.6.1 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.

5.5.6.2 Hardware IRQ IOxAPIC Interrupts

Dedicated pin interrupts may be edge or level triggered. They are routed to IRQ pins on I/OxAPIC device such as the PXH, or Intel 631xESB/632xESB I/O Controller Hub. The I/OxAPIC 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 SG 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 I/O Hub or Intel 631xESB/632xESB I/O Controller Hub in the system. The INTA will return data that provides the interrupt vector.

5.5.6.3 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 2.3 . This appears as a 1 DWORD write on the PCI/PCI-X/PCI Express bus.

With PCI devices, there are two types of MSIs. 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 (**not supported by the embedded MCH IOxAPIC**) can be used in the event the OS doesn't support MSIs, but the BIOS does. In either way, the interrupt will appear as an inbound write to the SG over the PCI Express ports.

MSI is expected to be supported by the operating systems when the MCH is available. A 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.

5.5.6.4 Non-MSI Interrupts - "Fake MSI"

For interrupts coming through the MCH, 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 Intel 631xESB/632xESB I/O Controller Hub since this is always present in a system.

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 SG and is redirected down the appropriate PCI Express port following the SG 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 SG core, and the SG core architecture guarantees that the subsequent APIC message cannot pass any posted data already within the SG.

5.5.6.5 Message Signalled Interrupts Extension (MSI-X)

MSI-X defines a separate optional extension to basic MSI functionality. MSI-X supports a larger maximum number of vectors per function, the ability for software to control aliasing when fewer vectors are allocated than requested, plus the ability for each vector to use an independent address and data value, specified by a table that resides in memory space. Most of the other characteristics of MSI-X are identical to those of MSI.

While both MSI and MSI-X each support per-vector masking, it is a standard feature with MSI-x while it is an optional extension to MSI. MSI-X also supports a function mask bit, which when set masks all of the vectors associated with a function.

Per-vector masking is managed through a mask and pending bit pair per MSI vector or MSI-X Table entry. An MSI vector is masked when its associated mask bit is set. An MSI-X vector is masked when its associated MSI-X table entry mask bit or the MSI-X function mask bit is set. While a vector is masked, the function is prohibited from sending the associated message, and the function must set the associated pending bit whenever the function would otherwise send the message. When software unmasks a vector whose associated pending bit is set, the function must schedule sending the associated message, and clear the pending bit as soon as the message has been sent. A function is permitted to implement both MSI and MSI-X, but system software is prohibited from enabling both at the same time. MSI and MSI-X use separate and independent capability structures.

5.5.7 Interprocessor Interrupts (IPIs)

- Previous IA-32 processors use IPIs after reset to select the boot strap processor (BSP). Recent XPF processors do not use IPIs to select the BSP. A hardware arbitration mechanism is used instead.
- IA32 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 SG completes the interrupt normally and forwards the interrupt to the other bus.

- For redirectable XAPIC interrupts, the SG will generate an interrupt message to both processor buses SG 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 SG will forward the broadcast interrupt to the other processor bus.
- Interrupts are not deferred.

Note: The MCH does not retry interrupts.

5.5.7.1 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.

5.5.8 Chipset Generated Interrupts

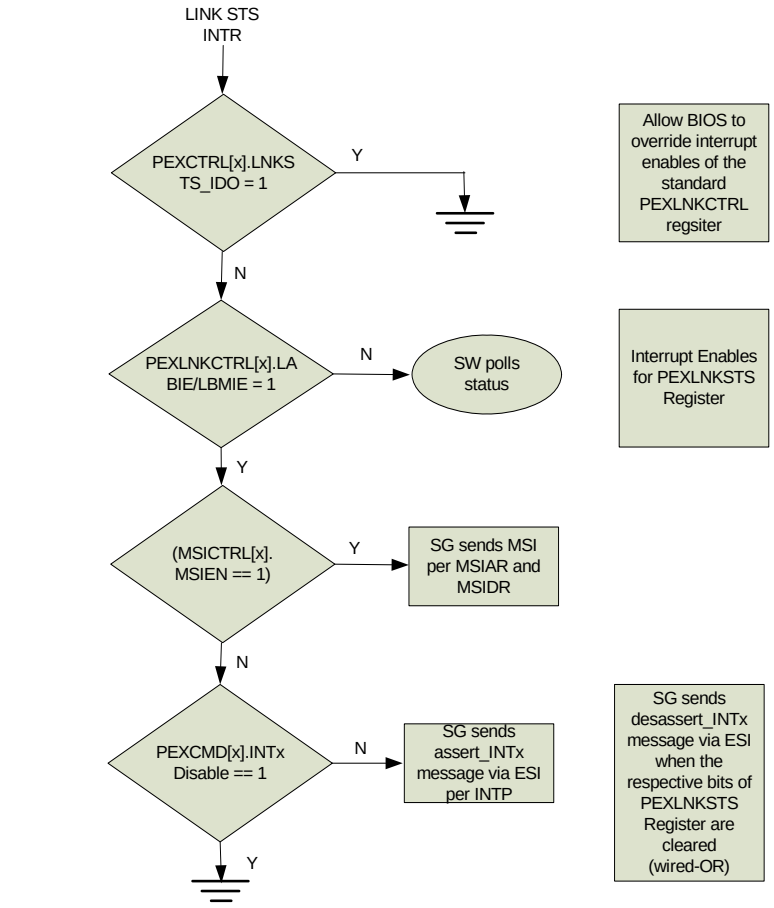
The MCH can trigger interrupts for chipset errors. For these events, the chipset can be programmed to assert pins that the system can route to an APIC controller. The interrupts generated by the chipset are still being defined. The following is a preliminary list of interrupts that can be generated.

1. Chipset error - MCH asserts appropriate ERR pin, depending on severity. This can be routed by the system to generate an interrupt at an interrupt controller. (MCH pins ERR[2:0], MCERR). The ERR[0] pin denotes a correctable and recoverable error. The ERR[1] pin denotes an uncorrectable error from MCH. The ERR[2] pin denotes a fatal error output from MCH.
2. PCI Express error - 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 as described in the RAS chapter.
      - MSI: Handled like a normal MSI interrupt (see Section 5.5.6.3)
3. 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. 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 the ESI port.

For the case when MSI is enabled, any new event that sets these bits (LABS or LMBS) will cause an MSI message to be sent to the FSB for each occurrence. i.e. each bit is considered unique. Whereas in the case of Legacy interrupts, a wired-OR approach is used to mimic the level sensitive behavior and only one assert_intx (deassert_intx) is sent even when multiple interrupt generating bits of the register get set. See Figure 5-26 for Link Status Interrupt flow.

Figure 5-26. PCI Express Link Status Interrupt Flow



LNKSTS_IDO	LABIE/LBMIE	MSIEN	INTx Disable	Output
1	x	x	x	--
0	1	1	x	MSI
0	1	0	0	assert_intx
0	1	0	1	--
0	0	x	x	--

0328060959

The following table, Table 5-12 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 5-12. Chipset Generated Interrupts

Source	Signalling mechanism	MCH signal method	Expected SW mechanism
Chipset Error	MCH registers	ERR[2:0], MCERR	Any
PCI Express Error	PCI Express ERR_COR/UNC/FATAL message	ERR[2:0], MCERR	Any
Power Management Event (PME)	PCI Express PM_PME message	Assert_PMEGPE, Deassert_PMEGPE to ESI	SCI->ACPI

5.5.8.1 MCH Generation of MSIs

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.

MSIs can be generated by the MCH internally for PM_PME messages, RAS errors and I/OxAPIC legacy interrupt translations errors.

Notable MSI-X enhancements over MSI include:

1. Support for 2048 vectors (instead of 32).
2. Support of independent message address and data for each vector.
3. Support of per-vector masking.

5.5.8.2 MSI Ordering in 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 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.

As a case in point, reads to MCH PCI Express (internal) configuration spaces will not generally guarantee ordering of internal MSIs from a root port device as required since the MCH uses a configuration ring methodology which houses the registers for the various PCI Express ports, MC, DFX etc.) and it operates independently of the MSI/interrupt generation logic. Thus any configuration ring access targeting a PCI Express port registers will not necessarily order and align with the internal MSIs.

Solution: To mitigate this problem and enforce ordering of the MSIs, the MCH will implement a “pending MSI signal” that is broadcast from each MSI/(or APIC message) capable unit to the coherency engine and thereby block the configuration request (non-posted) till all the MSI gets committed. Software will ensure that it will block future MSI generation for that device when it issues the configuration read for that device.

The CE will block sending any completion with the new bit-slice bit set when any of the pending MSI wires is asserted. CE will not block other transactions or completions during the block. When the pending MSI wires are deasserted, CE will be able to send the configuration completions.

The MCH Coherency Engine (CE) will block processor initiated MCH configuration access completions (MMCFG or CFC/CF8) if there is a pending internally generated MSI (or APIC message) within the MCH.

The pending MSI signal will be deasserted after fetch-completion is asserted for the MSI from CE, i.e. global visibility is guaranteed on the FSB. Then release the configuration block and allow the configuration completion to flow through. This approach will order the MSI and then send the non-posted configuration for that device.

CE will add a bit-slice (one bit per table entry) to track processor initiated MCH configuration access in CE transaction table. Note: Inbound configuration access will not set this bit.

A defeature mode to control the MSI/NP_CFG ordering is defined in the COHDEF.DIS_MSI_NPCFG register field.

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.

5.5.9 Legacy/8259 Interrupts

8259 interrupt controller is supported in the MCH platforms. 8259 interrupt request is delivered using the interrupt group sideband signals LINT[1:0] (a.k.a. NMI/INTR) or through an I/OxAPIC using the message based interrupt delivery mechanism with the delivery mode set to ExtINT (111b). There can be only one active 8259 controller in the system.

The mechanism in which a PCI Express device requests an 8259 interrupt is a PCI Express inband message. (ASSERT_INTA/B/C/D, DEASSERT_INTA/B/C/D).

The target processor for the interrupt uses the interrupt acknowledge transaction to obtain the interrupt vector from the 8259 controller. The SG forwards the interrupt acknowledge to the Intel 631xESB/632xESB I/O Controller Hub where the active 8259 controller resides.

The SG will support PCI Express devices that generate 8259 interrupts (for example, during boot). 8259 interrupts from PCI Express devices will be sent in-band to the SG which will forward these interrupts to the Intel 631xESB/632xESB I/O Controller Hub.

The MCH has a mechanism to track inband 8259 interrupts from each PCI Express and assert virtual interrupt signals to the 8259 through the inband "Assert_(Deassert)_Intx" messages. This is done by a tracking bit per interrupt (A, B, C, D) in each PCI Express which are combined (OR'd) into virtual signals that are sent to the Intel 631xESB/632xESB I/O Controller Hub. Each interrupt signal (A, B, C, D) from each PCI Express is OR'ed together to form virtual INT A, B, C, and D signals to the Intel 631xESB/632xESB I/O Controller Hub (Assert_(Deassert)_IntA/B/C/D (assertion encoding)). When all of the tracking bits for a given interrupt (A, B, C, or D) are cleared from all PCI Express ports, the virtual signal A, B, C, or D is deasserted via the inband Deassert_Intx message.

For PCI Express hierarchies, interrupts will be consolidated at each level. For example, a PCI Express switch connected to a SG PCI Express port will only send a maximum of 4 interrupts at a time, regardless of how many interrupts are issued downstream.

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. MCH interrupt signals described in Section 5.5.8 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 MCH interrupt signals to the appropriate Intel 631xESB/632xESB I/O Controller Hub pin.

5.5.10 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. For more details on error handling, please refer to the RAS chapter.

5.5.11 Integrated I/OxAPIC

The integrated I/OxAPIC in the MCH converts legacy PCI Express interrupt messages into APIC messages/interrupts. The I/OxAPIC appears as a PCI endpoint device in the MCH configuration space. The I/OxAPIC has a 24 entry table that allows for 24 unique APIC messages/interrupts. T his table is programmed via either the MBAR memory region or the ABAR memory region (refer to Section 3.10.1.5, "MBAR: IOxAPIC Memory Base Address Register" , Section 3.10.1.9, "ABAR: I/OxAPIC Alternate BAR") for details of the MBAR and ABAR registers). In the MCH, there are 37 unique legacy interrupts possible (9 root ports * 4 + 1 for Intel® QuickData Technology Device) and these are mapped to the 24 entries in the I/OxAPIC as shown in Table 5-13. The distribution is based on guaranteeing that there is at least one unshared interrupt line (INTA) for each possible source of interrupt. Furthermore, as the PCIE sources combine to establish larger link widths, more unshared interrupt lines will become available. When a legacy interrupt asserts, an APIC message/interrupt is generated (if corresponding I/OxAPIC entry is unmasked, bus mastering is enabled, etc.) based on the information programmed in the corresponding I/OxAPIC table entry. Table 5-14 and Table 5-15 provide the format of the interrupt message generated by the I/OxAPIC based on the table values.

Table 5-13. Integrated I/OxAPIC Table Mapping to PCI Express Interrupts (Sheet 1 of 2)

I/OxAPIC Table Entry#	PCI Express Port/Intel® QuickData Technology Device#	PCI Express Virtual Wire Type[1]
0	1 (X4/8/16)	INTA
1	2 (X4)	INTA
2	3 (X4/8)	INTA
3	4 (X4)	INTA
4	5 (X4/8/16)	INTA

Table 5-13. Integrated I/OxAPIC Table Mapping to PCI Express Interrupts (Sheet 2 of 2)

I/OxAPIC Table Entry#	PCI Express Port/Intel® QuickData Technology Device#	PCI Express Virtual Wire Type[1]
5	6 (X4)	INTA
6	7 (X4/8)	INTA
7	8 (X4)	INTA
8	9 (X4)	INTA
9	15	INTA
10	1, <2>	INTB, <INTD>
11	3, <4>	INTB, <INTD>
12	5, <6>	INTB, <INTD>
13	7, <8>	INTB, <INTD>
14	2, <1>	INTB, <INTD>
15	4, <3>	INTB, <INTD>
16	6, <5>	INTB, <INTD>
17	8, <7>	INTB, <INTD>
18	1, <2>	INTC, <INTC>
19	3, <4>	INTC, <INTC>
20	5, <6>	INTC, <INTC>
21	7, <8>	INTC, <INTC>
22	9, <9>	INTB, <INTD>
23	9	INTC

Notes:

1. <> associates interrupt from a given device number (as shown in the 'PCI Express Port/Intel® QuickData Technology Device#' column) that is marked thus to the corresponding interrupt wire type (shown in this column) also marked such. For example, I/OxAPIC entry 11 corresponds to the Wire-OR of INTB message from Device#1 and INTD message from Device#2.

Table 5-14. Address Format of Interrupt Generated from Integrated I/OxAPIC

Bits	Description
31:20	FEEh
19:12	**Destination ID:** This will be the bits 63:56 of the I/O Redirection Table entry for the interrupt associated with this message.
11:4	**Enhanced Destination ID:** This will be the bits 55:48 of the I/O Redirection Table entry for the interrupt associated with this message.
3	**Redirection Hint**: This bit allows the interrupt message to be directed to one among many targets, based on chipset redirection algorithm. 0 = The message will be delivered to the agent (CPU) listed in bits 19:4 1 = The message will be delivered to an agent based on the IOH redirection algorithm and the scope the interrupt as specified in the interrupt address. The Redirection Hint bit will be a 1 if bits 10:8 in the Delivery Mode field associated with corresponding interrupt are encoded as 001 (Lowest Priority). Otherwise, the Redirection Hint bit will be 0.
2	**Destination Mode:** This is the corresponding bit from the I/O Redirection Table entry. 1=logical mode and 0=physical mode.
1:0	00

Table 5-15. Data Format of Interrupt Generated from Integrated I/OxAPIC

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

5.5.11.1 Integrated I/OxAPIC EOI Flow

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

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

In addition to EOI broadcasts, Intel® 5400 chipset also supports directed writes to its EOI register within the embedded I/OxAPIC from the host.

5.5.11.2 Integrated I/OxAPIC Misc Notes

The APIC specification has a feature where writes to the ‘Pin Assertion Register (PAR)’ in the I/OxAPIC memory space, generates an interrupt. This feature is NOT supported in the MCH, and the Pin Assertion Register is not implemented. Therefore, fake MSIs are not supported by the IOxAPIC in the MCH, and inbound accesses to this address range is not allowed.

The MCH only supports level type interrupts, if the Read/Write TM bit of a given Redirection Table Entry (RTE) is set to indicate Edge it could result in lost interrupts.

The MCH is defined to have active low input pin polarity and must be programmed accordingly or it could result in lost interrupts.

5.5.11.3 PCI Express INTx Message Ordering

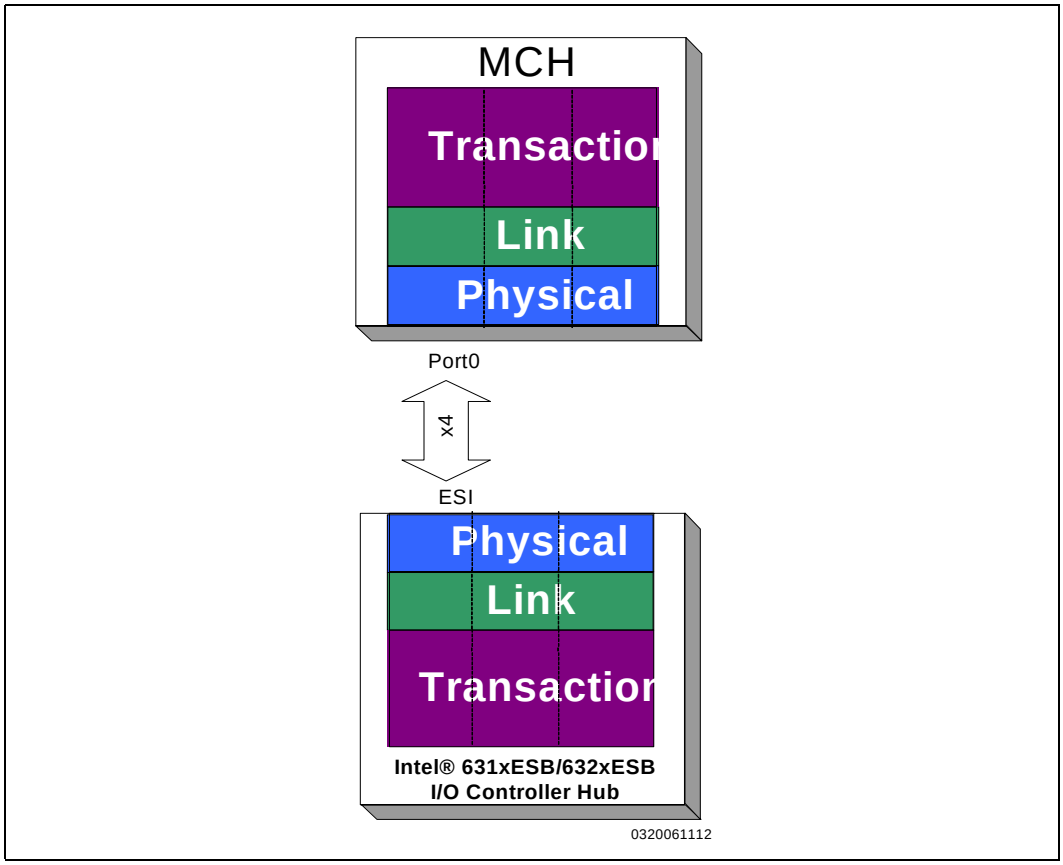
INTx messages on PCI Express are posted transactions and hence must follow the posted ordering rules in the MCH. For example, if INTx message is preceded by a memory write A, then the INTx message must push the memory write to a global ordering point (CE) before it is delivered to its destination (which could be the I/OxAPIC cluster that decides further action). This will guarantee that any APIC message/interrupt generated from the integrated I/OxAPIC (or from the I/OxAPIC in the platform components, if the integrated I/OxAPIC is disabled) will be ordered behind the memory write A, guaranteeing producer/consumer sanity.

However, for INTx interrupts generated by the root port itself, these are not considered posted from a spec perspective and are hence not subject to the same ordering requirements.

5.6 Enterprise South Bridge Interface (ESI)

The Enterprise South Bridge Interface (ESI) in the Intel® 5400 chipset north bridge is the chip-to-chip connection to the Intel 631xESB/632xESB I/O Controller Hub see Figure 5-27. The ESI is a specialized interchip interface based upon the *PCI-Express Base Specification*, Revision 1.1 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. 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 Intel® 5400 chipset north bridge.

Figure 5-27. MCH to Intel 631xESB/632xESB I/O Controller Hub Enterprise South Bridge Interface



The ESI port in the Intel® 5400 chipset north bridge may be combined with one additional PCI Express port, Port 9 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 2 GB/s (1GB/s each direction). When the ESI is pared with the additional x4 PCI Express link, the available bi-directional bandwidth to the Intel 631xESB/632xESB I/O Controller Hub is increased to 4 GB/s.

5.6.1 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 and I/O transactions. 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 IO ordering is maintained until the end of the transaction. At this point, the ordering requirement are satisfied for the inbound request and the next order dependant transaction can proceed. The MCH does not support peer-to-peer where the source and destination is the same PCI Express interface.

5.6.2 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[9:0] 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.

5.6.2.1 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 Intel® 6400/6402 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 I/O Hub at the earliest.

5.6.2.2 STPCLK Propagation

The Intel 631xESB/632xESB I/O Controller Hub has a sideband signal called STPCLK. This signal is used to place IA32 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 re-architect 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.

5.6.3 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.

5.6.4 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.

5.6.5 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.

5.6.6 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.

5.6.7 Error Handling

Table 5-36 describes the errors detected on ESI through the standard PCI Express and Advanced error reporting mechanism.

5.6.7.1 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. If the original request was a peer-to-peer non-posted transaction, the MCH returns just the completion status as required by the *PCI-Express Base Specification*, Revision 1.1.

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 depending on the message type. If the completion required bit is set, an Unsupported Special Cycle completion is returned.

5.6.7.2 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.

5.7 PCI Express

The PCI Express port 1 through port 8 are general purpose x4 PCI Express ports that may be used to connect to PCI Express devices. These x4 ports may be combined into high performance x8 PCI Express ports or up to two optimized high performance x16 graphics interface ports as well. The x16 ports contain several architectural enhancements to improve graphics performance. Port 1and Port 2, Port 3 and Port 4, Port 5 and Port 6, Port 7 and Port 8 are combinable to form single x8 ports. Ports 1-4 and ports 5-8 are combinable to form the two x16 ports. The possible configurations of the PCI Express ports are depicted in Figure 5-29 and the high performance x16 ports are specifically depicted in Figure 5-30.

PCI Express Port 9 is a dedicated x4 port and can not be combined with any other ports.

5.7.1 PCI Express Ports

The Intel® 5400 chipset MCH contains three sets of PCI Express derived ports. These are:

- IOU0 Port Group - Ports 1-4
- IOU1 Port Group - Ports 5-8
- Port 9

The ESI port is the primary interface to the Intel 631xESB/632xESB I/O Controller Hub. This interface can be paired up with PCI Express port 9 to increase available bandwidth to the Intel 631xESB/632xESB I/O Controller Hub. The Intel® 5400 chipset MCH supports up to four high performance x8 ports at on GEN 1 speed or up to two high performance x16 graphics PCI Express ports at revision 2.0 speed. This port contains several architectural enhancements to increase graphics performance.

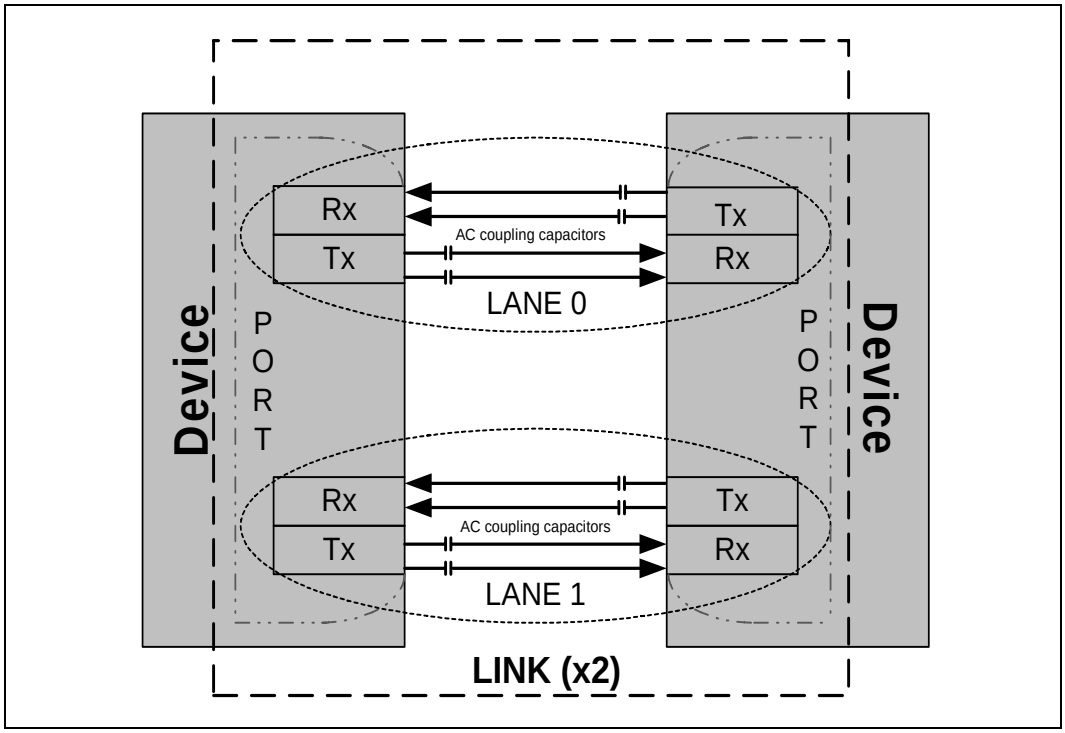
The following sections describe the characteristics of each of these port classes in detail.

5.7.2 Intel® 5400 Chipset MCH PCI Express Port Overview

The Intel® 5400 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, version 0.9 of Revision 2.0*

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 5-28.

Figure 5-28. x2 PCI Express Bit Lane



The raw bit-rate per PCI Express Gen 1 bit lane is 2.5 Gbit/s. This results in a real bandwidth per Gen 1bit lane pair 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. The raw bit-rate per PCI Express Gen 2 bit lane is 5 Gbit/s similarly with a maximum theoretical realized bandwidth of 2 GB/s in each direction. The Gen 2 speeds are only supported for the x16 High Performance PCI Express interfaces.

Each of the Intel® 5400 chipset MCH PCI Express port are organized as four bi-directional bit lanes, and are referred to as a x4 port.

Figure 5-29. PCI Express Port Configurations

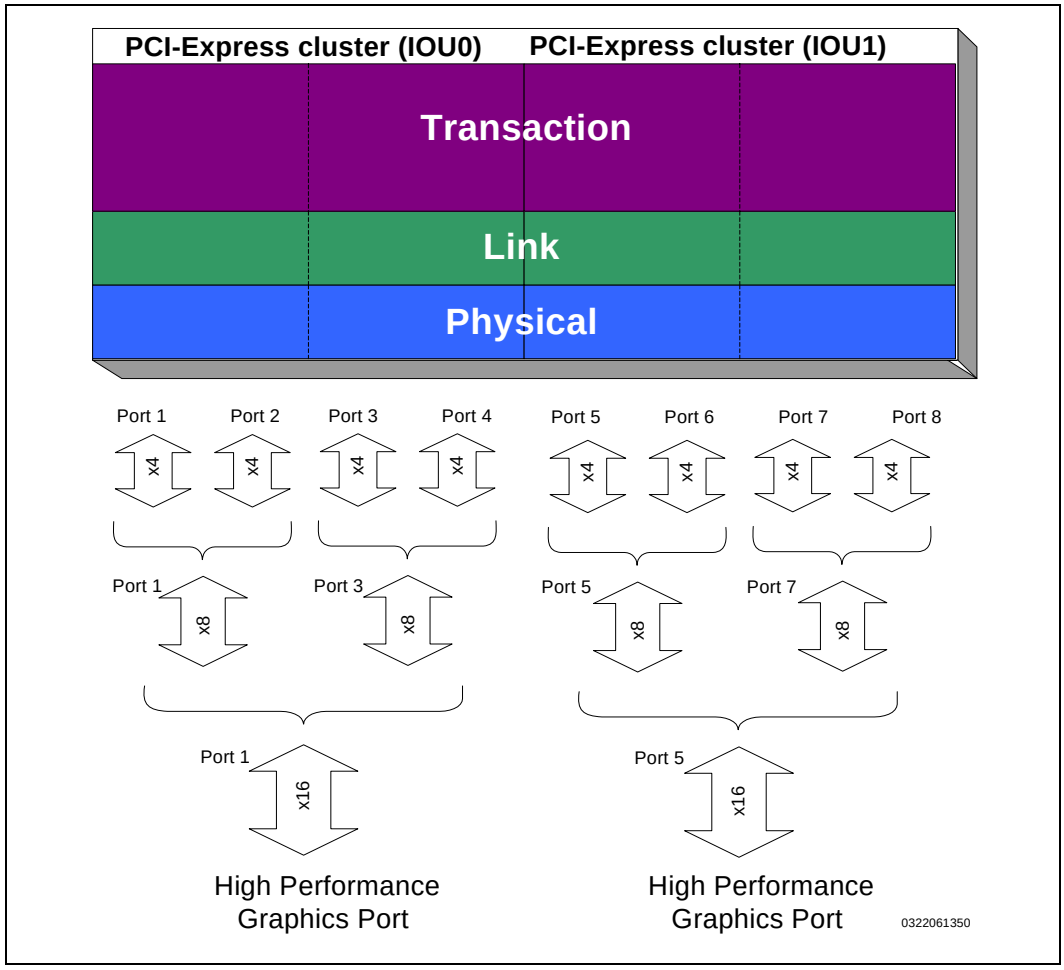
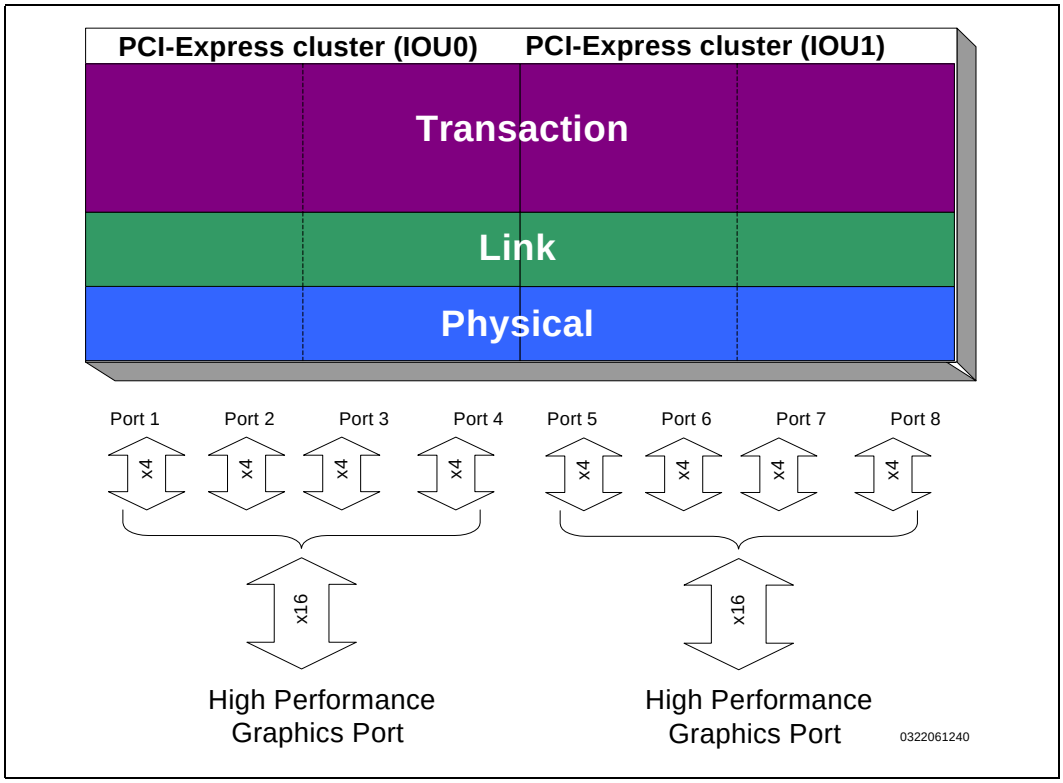


Figure 5-30. PCI Express High Performance x16 Port



5.7.3 Supported Length Width Port Partitioning

To establish a connection between PCI Express endpoints, they both participate in a sequence of steps known as training. This sequence will establish the operational width of the link 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. In the case of a x8 port, the x4 link pairs will first attempt to train independently, and will collapse to a single link at the x8 width upon detection of a single device returning link ID information upstream. Once the number of links has been established, they 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 two devices. 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 *PCI Express Base Specification Revision 2.0* requires that a port be capable of negotiating and operating at the native width and x1. The Intel® 5400 chipset MCH will support the following link widths for its PCI Express ports viz., x16, x8, x4, x2 and x1. During link training, the MCH will attempt link negotiation starting from its native link width from the highest and ramp down to the nearest supported link width that passes negotiation. For example, a port configured at 8x, will first attempt negotiation at 8x. If that attempt fails, an attempt is made at x4, then a x2 and finally a x1 link. Note that the x8, x4 and x2 link widths will only use the LSB positions from lane 0 while a x1 can be connect to any of the 4 positions (lane0,lane1, lane2, lane3) providing a higher tolerance to single point lane failures.

5.7.4 PCI Express Port Support Summary

The following table describes the options and limitations supported by the MCH PCI Express ports.

Table 5-16. Options and Limitations

Parameter	Support
Number of supported ports	The MCH will support nine x4 standard PCI Express ports and an additional x4 ESI port for Intel 631xESB/632xESB I/O Controller Hub. (Total: 9 + 1 = 10 ports)
Max payload	256B
Virtual Channels	MCH only supports VC0
Traffic Streams	The ports support only one stream and 1 priority level (including the ESI) on VC0.
Isochrony	MCH does not support isochrony
ECRC	MCH does not support ECRC
Ordering	MCH only supports relaxed PCI ordering (see Figure 5.6.14, "Ordering Rules")
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.
No Cable Support & no repeaters	Retry buffers are sized to meet the Intel® 5400 chipset platform requirements for an integrated DP chassis and which do not require cable or repeater support. Only an 8 inches of FR4 internal trace connector latency is assumed.
Poisoning	MCH will poison data that it cannot correct

5.7.5 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)
- PCI Express Generation 1 -- 2.5 GHz data rate (up to 2 GB/s/direction peak bandwidth for a x8 port)
- PCI Express Generation 2 -- 5 GHz data rate
- 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

5.7.6 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

Note: The PCI Express flow control credit types are described in Table 5-17. The *PCI Express Base Specification Revision 2.0* defines which TLPs are covered by each flow control type.

Table 5-17. PCI Express Credit Mapping for Inbound Transactions

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 (ESI) 28(4x) 56(x8) 112(x16)
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 [ESI] 108 [x4] 216 [x8] 432 [x16]
Inbound Non-Posted Request Header Credits (INPRH)	Tracks the number of non-posted requests the agent is capable of supporting. Each credit accounts for one non-posted request.	14 [ESI] 28 [x4] 56 [x8] 112 [x16]
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(ESI) 4(4X) 8(8X) 16(x16)
Completion Header Credits (CPH) (outbound request completions received at the MCH)	Tracks the number of completion headers the agent is capable of supporting.[1]	0 (Infinite) 4 (x4) 8 (x8) 16 (x16)
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) 32 (x16) 64 (x16)

Notes:

1. 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

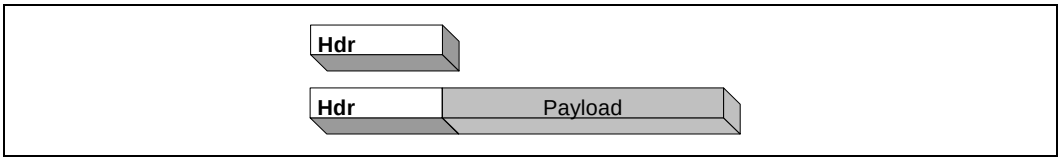
5.7.7 Transaction Layer

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 5-31 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 5-31. PCI Express Packet Visibility By Transaction Layer

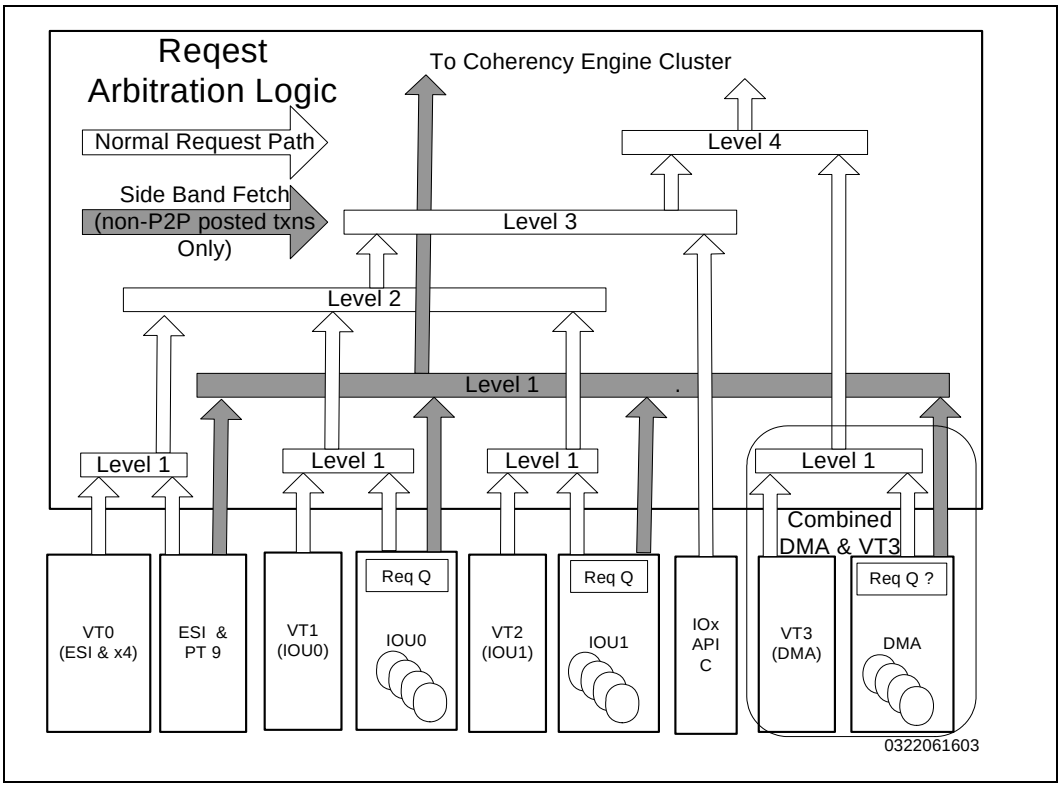


5.7.8 Upstream Target Request Arbitration

Arbitration in the PCI Express (IOU) cluster(s) occurs at several levels to dispatch upstream transactions to memory, FSB, or peer-to-peer targets. Virtualization, Intel® QuickData Technology Device, and IOxAPCI's MSI requests are serviced in addition to PCI Express traffic. The basic methodology is to guarantee fairness, prevent starvation and provide limited programmability for debug/field tuning. See Figure 5-32 for illustration of Arbitration.

5.7.9 Port Arbitration

Figure 5-32. Request Arbitration Logic for PCI Express, Intel VT-d, IOxAPIC, and Intel® QuickData Technology Device



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 and prevent starvation.

- Level 0:
  - Level 0a: Weighted round robin algorithm that alternates between the active ports with in an IOU or ESI/Pt9 based on width of the port and speed.
  - Level 0a VTd Arbitration: Weighted Round Robin based on port size and speed for (VT0, VT1, VT2).
  - Level 0 Intel® QuickData Technology Device Arbitration: Round robin arbitration between all active channels
  - Level 0b Flow Class Arbitration for PCIE Port Winner: Round robin based arbitration-based on transaction flow class (Posted, Non-posted, Completions)
- Level 1: This arbitration level is arbitrating between request source and its associated virtualization cache memory access when virtualization is enabled. For the Intel® QuickData Technology Device and its VTd traffic, this level is contained within the Intel® QuickData Technology Device. IOUs and its VTd device traffic: Programmable round robin algorithm for IOUs or ESI/PT9 and its associated VTd traffic.
- Level 2: This arbitration level is arbitrating between the different IOUs and ESI/PT9 request sources. The arbitration scheme is a full programmable weighted round robin scheme. Depending of the required bandwidth for each IOU or ESI/PT9, each request source can be weighted from 1 to 8 independently.
- Level 3: This arbitration level is arbitrating between the level 2 winner and IOxAPIC message system interrupt (MSI) transaction. It is a simple round robin arbitration.Level 3 - IOU0, IOU1, Intel® QuickData Technology Device Arbitration
- Level 4: This final level of arbitration is between the level 3 winner and Intel® QuickData Technology Device traffic. This level of arbitration has a limited programmable round robin algorithm. The default weight ratio is 4-to-1 or between demand level 3 to Intel® QuickData Technology Device traffic.

5.7.10 Supported PCI Express Transactions

Table 5-18 lists all the transactions supported by the Intel® 5400 chipset MCH which are expected to be received from the PCI Express interface. Similarly, Table 5-20 lists all the transactions to be expected by an attached PCI Express component. Refer to the *PCI Express Base Specification Revision 2.0* for the specific protocol requirements of this interface.

Table 5-18. Incoming PCI Express Requests (Sheet 1 of 2)

PCI Express Transaction	Address Space or Message	Intel® 5400 chipset MCH Response
Inbound Write Requests	Memory	Forward to Main Memory or PCI Express or ESI port depending on address.
	I/O	Forward to peer PCI Express or ESI port
	Configuration(Type0 or Type1)	Forward to peer PCI Express or Hub Interface port
Inbound Read Requests	Memory	Forward to Main Memory, or PCI Express or ESI
	I/O	Forward to peer PCI Express or ESI Interface port
	Configuration(Type0 or Type1)	Forward to peer PCI Express or Hub Interface port

Table 5-18. Incoming PCI Express Requests (Sheet 2 of 2)

PCI Express Transaction	Address Space or Message	Intel® 5400 chipset MCH Response
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.
	ATTENTION_BUTTON_PRESSED	Terminate the message and set the PEXSLTSTS.”Attention Button Pressed”. If PEXCTRL.”Attention Button Pressed Enable“and bits are set, send MSI, Assert_HPGPE, or assert_intx on the ESI.
	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
	Others	Set IO2 error (unsupported request), drop transaction (master abort) and return credit.

Table 5-19. Incoming PCI Express Completions

PCI Express Transaction	Address Space or Message	Intel® 5400 chipset MCH Response
Completions for Outbound Writes	I/O or Configuration[1]	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:

1. Outbound Memory writes are posted and have no completions

Table 5-20. 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, 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_Active_State_NAK	Intel® 5400 chipset MCH will generate the "PM_Active_State_NAK" message to the end device in response to receiving a "PM_Active_State_Request_L1" DLLP when Intel® 5400 chipset MCH cannot transition to the L1 state.
	PM_TURN_OFF	Intel® 5400 chipset MCH will generate the "PM_Active_State_NAK" message to the end device in response to receiving a "PM_Active_State_Request_L1" DLLP when Intel® 5400 chipset MCH cannot transition to the L1 state.
	PM_REQUEST_ACK (DLLP)	Received PM_ENTER_L1 and PM_ENTER_L23. This message is continuously issued until link is idle.
	PM_TURN_OFF	PEXGCTRL.PME_TURN_OFF bit was set. This message is broadcast to all enabled PCI Express ports.
	set_slot_power_limit	This Message is sent by Intel® 5400 chipset MCH to convey a slot power limitation value to a downstream component connected to this port. The Set_Slot_Power_Limit Message includes a 1 DW data payload1 consisting of the Slot Power limit Scale (SPLS) and Slot Power Limit Value (SPLV) fields copied from the PCI Express Slot Capabilities register. *PCI Express Base Specification Revision 2.0* for details on the set_slot_power_limit message (Sections 2.2.8.5) This Message is sent automatically when one of the following events occurs: • On a Configuration Write to the Slot Capabilities register (when the DL Layer reports a DL_Up status. • Anytime when a Link transitions from a non-DL_Up status to a DL_Up status

Table 5-21. 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.
Inbound Reads or Writes	I/O	Response for an inbound read to a peer I/O device.
	Configuration (Type0 or Type1)	Response for an inbound read to a peer I/O device.

5.7.10.1 Unsupported Messages

If the Intel® 5400 chipset MCH decodes any vendor message (which is not defined in Table 5-18), the MCH will take the following actions as specified in the Vendor defined message section of the *PCI Express Base Specification Revision 2.0*.

- Vendor Type 0 - Unsupported Request
- Vendor Type 1 - Drop request.

5.7.10.2 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 38 bits for MCH)

5.7.11 Transaction Descriptor

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

- Transaction ID
- Attributes
- Virtual Channel ID

5.7.11.1 Transaction ID

The Transaction ID uniquely identifies every transaction in the system. The Transaction ID comprises four sub-fields described in Table 5-22.

Table 5-22. 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.		

5.7.11.2 Attributes

PCI Express supports two attribute hints described in Table 5-23.

Table 5-23. 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.	

5.7.11.3 Traffic Class

Intel® 5400 chipset MCH does not support any PCI Express virtual channels except for the default channel VC0. There is no TC-VC mapping capability structure and there is no RCRB implemented. However, as per the *PCI Express Base Specification Revision 2.0* specification, a completer must accept requests with TC label other than TC0, and must preserve the TC label. Intel® 5400 chipset MCH accepts inbound transactions with zero/non-zero TC but it treats the non-zero TC as mapped to VC0 for inbound requests destined for memory. For completions heading upstream, Intel® 5400 chipset MCH will copy the TC value received into the completion packet. For peer-to-peer transactions (Memory), the TC value of the request is preserved and processed. However if Intel® 5400 chipset MCH receives a non-zero TC for any I/O or Configuration transactions, it will be treated as a malformed TLP and the error logged appropriately.

For outbound requests, the MCH sets this ID to zero.

5.7.12 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.

5.7.12.1 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 5-22. 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 5.7.13.

5.7.12.1.1 Inbound Memory Reads

Read Completion Policy

For inbound read requests, the MCH is allowed to split completions along a Read Completion Boundary (RCB) of 64B, PEXLNKCTRL[9:0] - PCI Express Link Control Register. For MCH, the maximum size of a read completion is specified with

Max_Payload_Size field as 256B, PEXDEVCTRL2[9:0]: PCI Express Device Control 2 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 cache line 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.

5.7.12.2 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: Multiple cache line as non-speculative prefetch or RFOs are issued for one or more enqueued requests. These prefetch or RFO commands are routed to the Coherency Engine (CE)/ central Data Manager (DM) for decoding and then sent to memory/FSB (for snoops) if required.
2. Fetch Phase: Fetch request or write request is issued to CE once PCIE IO ordering rules are satisfied. For reads, data from cache line buffers in the CE/DM are sent to destination port or device. For writes, data is written into the cache line buffers in the CE/DM to be forwarded to the ultimate destination. A Fetch completion is sent to terminate the cycle and remove buffer entries. In many cases1, a fetch request will be issued without a prefetch. This will always be the case for inbound read requests from the PCI Express ports.

In the prefetch phase, the streaming logic in the IOU breaks inbound transactions (in the order received) into one or more cachelines and pipelines them to the CE. It should send enough requests up to the total number of outstanding cachelines that the implementation2 can handle to maximize bandwidth on the PCI Express port. When acknowledgement for the individual cachelines in the prefetch phase is returned, the fetch phase begins immediately after PCIE IO ordering rules met. The internal commands are pipelined to the coherency engine/DM to obtain or push the data as fast as possible.

5.7.12.3 Zero-Length Reads/Writes

The *PCI Express Base Specification Revision 2.0* 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, the request is treated as a read request of one data and is actually read from memory. The read request is completed by Intel® 5400 chipset 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 the addressed location to the requesting PCI Express port.

Note: Any inbound memory request targeting control registers within Intel® 5400 chipset MCH will be master aborted.

regardless of length.The *PCI Express Base Specification Revision 2.0* 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.

5.7.12.4 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.

Note: *Because of the PCI Express ordering rules, transactions after an inbound write must wait until after the write is globally visible. The inbound queue must be deep enough such that continuos inbound traffic is not stalled waiting for the above write sequence even under heavily leaded conditions.*

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.

5.7.12.5 Inbound Write Combining

The MCH performs write combining opportunistically in the on-chip coherent data buffer.

5.7.12.6 Interrupt Handling

A PCI Express device represents interrupts with either MSI or inbound interrupt messages (ASSERT_INTx/DEASSERT_INTx).

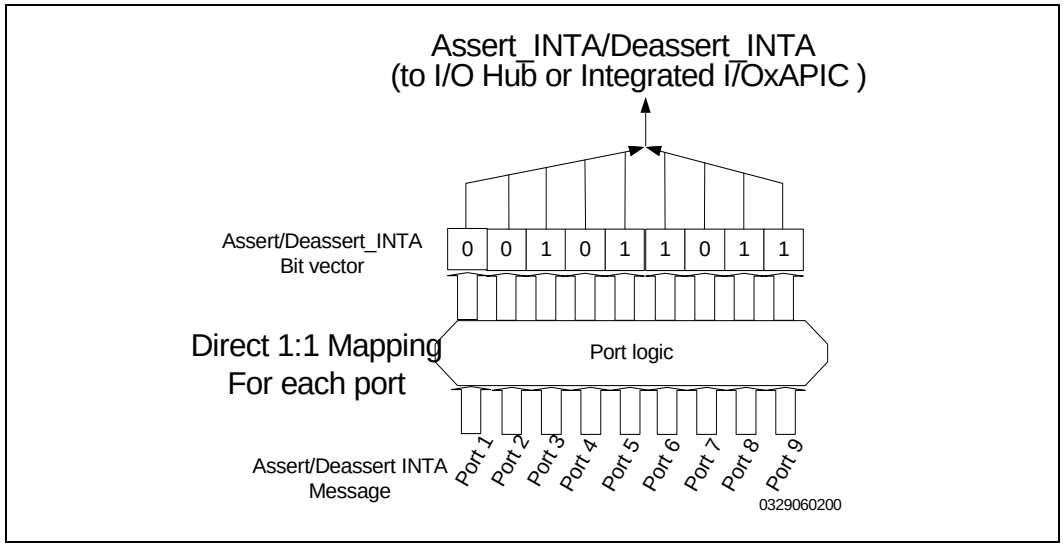
Each PCI Express port of the MCH is responsible for tracking assert/deassert messages for each of the four interrupts (INTA, INTB, INTC, INTD) and representing them with four output "virtual (ASSERT_INTA, ASSERT_INTB, ASSERT_INTC, and ASSERT_INTD) messages to the Intel 631xESB/632xESB I/O Controller Hub.

Figure 5-33 illustrates how the PCI Express interrupts are routed to the Intel 631xESB/632xESB I/O Controller Hub. The example shown represents Interrupt A and this logic is replicated for the four interrupts. The bits depicted are software visible.

When a PCI Express assert message is received for a specific interrupt, another assert message will not arrive until after a deassert message has arrived for that interrupt first.

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 0000h - FEDF FFFFh. 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 5-33. Legacy Interrupt Routing (INTA Example)



5.7.12.7 Error Messages

PCI Express reports many error conditions through explicit error messages: ERR_COR, ERR_UNC, ERR_FATAL.

5.7.12.8 Inbound Vendor-Specific Messages

ASSERT_GPE / DEASSERT_GPE

ASSERT_GPE and DEASSERT_GPE form a virtual wire which is sent by a PCI Express device.

5.7.12.9 Outbound Transactions

The MCH will generate the outbound transactions listed in Table 5-20. 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 Section 5.7.13.2, "Outbound Transaction Ordering Rules" .

5.7.12.10 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.

5.7.12.10.1 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.

5.7.12.10.2 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.

5.7.12.11 Outbound Vendor-Specific Messages

The MCH supports only vendor-specific EOI messages outbound.

5.7.12.11.1 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).

5.7.12.12 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 2.0*.

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 Intel® 5400 chipset 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.

5.7.13 Ordering Rules

This section describes the MCH ordering rules for transactions progressing through the PCI Express unit.

5.7.13.1 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.

5.7.13.1.1 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 5-18. 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.

5.7.13.2 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 2.0*.

5.7.13.2.1 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.

5.7.13.3 MCH Ordering Implementation

The following table summarizes the rules enforced on transactions from a given PCI Express port by the MCH.

Table 5-24. 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

5.7.13.3.1 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 5.7.13.1, "Inbound Transaction Ordering Rules" and Section 5.7.13.2, "Outbound Transaction Ordering Rules" .

5.7.13.4 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.

5.7.13.4.1 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.

5.7.14 Prefetching Policies

The MCH does not perform any speculative prefetching for PCI Express interface component reads. The PCI Express component south of the Intel® 5400 chipset 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.

5.7.15 No Isochronous Support

The Intel® 5400 chipset MCH does not support isochrony. Only the default virtual channel (channel 0) is supported on the PCI Express interfaces.

5.7.16 PCI Express RAS Features

The PCI Express interfaces will have traditional CRC protection. The data packets will utilize a 32-bit CRC protection scheme. The smaller and less error-prone link packets will utilize a 16-bit CRC scheme. Since packets utilize 8B/10B encoding, and not all the encodings are used; this provides further data protection, as illegal codes are also detected. If errors are detected on received data packets, these data packets are retransmitted. Hardware logic will support this link-level retry without software intervention. If a link lane fails, the link can reconfigure itself during retraining to a link of smaller width to continue operation at a lower performance level.

5.7.16.1 PCI Express Retry

The PCI Express interface will incorporate a link level retry mechanism. The hardware will detect when a transmission packet is corrupted and a retry of that particular packet and all following packets will be performed. Although this will cause a temporary interruption in the delivery of packets, it does so in order to maintain the link integrity.

5.7.16.2 PCI Express Recovery

When enough errors occur, the hardware may determine that the quality of the connection is in question, and the end points can enter what amounts to a quick training sequence known as recovery. The width of the connection will not be renegotiated, but the adjustment of skew between lanes of the link may occur. This occurs without any software intervention, but the software may be notified.

5.7.16.3 PCI Express Retrain

If the hardware is unable to perform a successful recovery, then the software is notified of the link condition, and software will invoke a hardware retrain. Once software has to be involved, then data will probably be lost, and processes will needed to be restarted, but this is still better than having to take the system down, or offline for an extended period of time.

5.7.17 I Unsupported Transactions and Unexpected Completions

If the MCH receives a packet that is not included in Table 5-18, 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
- 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.

5.7.18 ECRC

The *PCI Express Base Specification 2.0* supports an optional end-to-end CRC mechanism (ECRC). **The MCH does not support this feature**.

5.7.19 EDB

The *PCI Express Base Specification 2.0* 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". EDB can be generated in certain "stop and scream" situations when returning 128 or 256B completions. EDB will be used if poisoned data was discovered after transmission of TLP has been initiated

5.7.20 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 2.0* for the various mechanisms) then the PCI Express unit poisons the data before forwarding internally to the destination interface.

5.7.21 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) 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.

5.7.22 PCI Express Power Management Support

The *PCI Express Base Specification Revision 2.0* 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.

5.7.22.1 L0s

The MCH supports Active State PM transition (ASPM) into L0s state (i.e MCH powers down link into a "standby" mode and transmits electrical_idle ordered sets). The MCH also tolerates an end PCI Express device placing the link into an L0s state. The L0s support in the MCH follows the base *PCI Express Base Specification Revision 2.0*.

5.7.22.2 L1 Entry

Refer to the *PCI Express Base Specification Revision 2.0* for details. This section describes at a high level what is required of the MCH.

The *PCI Express Base Specification 2.0* lists two mechanisms for entering L1 state:

- Active State Power Management
- Software Initiated Power Management

Note: The MCH fully supports Software Initiated Power Management and Active State Power Management.

5.7.22.3 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.

5.7.22.4 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 21.8.9.34, "PEXGCTRL: PCI Express Global Control Register" on page 654). 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. When PEXGCTRL.PME_TO_Ack is set, the MCH will send an Assert_PMEGPE message to the Intel 631xESB/632xESB I/O Controller Hub. This causes an interrupt and the software takes appropriate action to inform the power controller. When software services this interrupt it clears PEXGCTRL.PME_TO_Ack, and this causes the MCH to send a deassert_PMEGPE message on the ESI port.

5.7.22.5 Wake Sequence

The *PCI Express Base Specification Revision 2.0* 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, 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.

5.7.23 BIOS Shutdown of Unused PCI Express Ports Support

The PCI Express cluster will allow BIOS to shutdown PCI Express ports for additional power savings. Port 9 (x4) would be ideal to turnoff if the port is not needed. Port 5/6 lanes or port 1/2 would be ideal if they are not needed to maximizes power savings.

The PCI Express root ports devices are controlled by DEVPRES: Device Present Control Register and the PEXLWTCTRL: PCI Express Link Width and Training Control Register. BIOS selects the "Device Not Present" for the PCI Express ports to turnoff/disable for power savings and locks down the register. The MCH hardware will use DEVPRES and PEXLWTCTRL register setting to determine the number of lanes that can be disabled for power savings. The PCI Express device states are permanent until the next hard reset. If port 9 is active when the DEVPRES register bit corresponding to port 9 is cleared, the harder will quiesce the port and bring the ports to a "Turnoff/Disabled" state gracefully. Refer to the PEXLWTCTRL: PCI Express Link Width and Training Control Register for more details on how to properly configure the PCI Express port.

5.8 Power Management

The MCH power management is compatible with the PCI Bus Power Management Interface Specification, Revision 1.2 (referred as PCI-PM). It is also compatible with the Advanced Configuration and Power Interface (ACPI) Specification, Revision 2.0b. The MCH is designed to operate seamless with operating systems employing these specifications.

The MCH power management support includes:

- ACPI supported
- System States: S0, S1 (desktop), S3, S4, S5, C0, C1, C2 (desktop)

5.8.1 Supported ACPI States

The MCH supports the following ACPI States:

- Processor
  - C0: Full On.
  - C1: Auto Halt.
- System
  - G0/S0: Full On.
  - G1/S1: Stop Grant, Desktop S1.
  - G1/S2: Not supported.
  - G1/S3: Suspend to RAM (STR). Power and context lost to chipset.

— 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).

5.8.2 Dynamic CKE Idle Control

The MCH is capable of switching off memory CKE during idle periods. This capability can be satisfied with the use of the DRAM power down control. See the DRAMPDCTRL register in the registers section

5.8.3 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.

5.8.3.1 Hardware-Based Thermal Management

The number of hex-words transferred over the DRAM interface are tracked. 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.

5.8.3.2 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.

5.8.4 FB-DIMM Thermal Diode Overview

The FB-DIMM Advanced Memory Buffer (AMB) contains an internal thermal sensor to measure AMB / DIMM temperature. Upon detecting a thermal over temperature condition the AMB initiates a thermal throttling event. For more information see the *Intel® 6400/6402 Advanced Memory Buffer External Design Specification (EDS), Revision 2.0*.

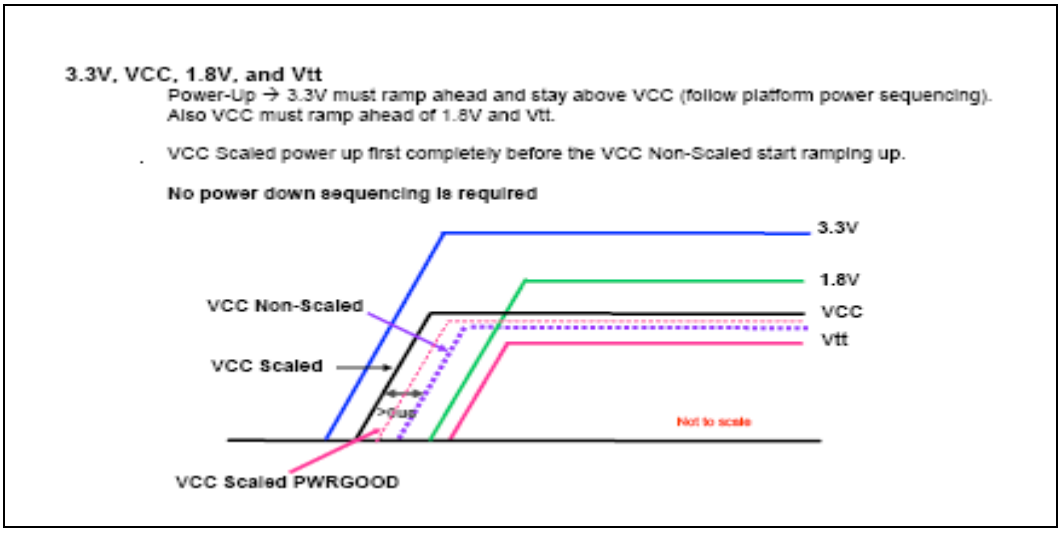
5.9 System Reset

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.

5.9.1 MCH Power Sequencing

General power sequencing requirements for the MCH are in general, the higher voltages must come up before lower voltages. Figure 5-34 depicts specific sequencing of the main voltages powering the MCH.

Figure 5-34. Power Sequencing



Note: Power-up -> 3.3 V must ramp ahead and stay above VCC, which must ramp ahead and stay above 1.8 V and Vtt. 3.3 V must always be at least 0.7 V greater than 1.8 V. Duration of the power ramp must be between 0.1 ms and 100 ms.

5.9.2 MCH Reset Types

The MCH differentiates among five types of reset as defined in table Table 5-25.

Table 5-25. MCH Reset Classes

Type	Mechanism	Effect / Description
Power-Good	PwrGd 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	**RSTIN**_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 FSBxRESET_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 FSBx**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.

5.9.2.1 Power-Good Mechanism

The initial boot of a MCH 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 2 mS) after the last voltage reference has stabilized. There are no requirements within the MCH regarding the precise sequencing of power-supply ramps, thus the platform should initialize properly regardless of the order in which supplies stabilize.

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_N output and maintain its assertion for 1 mS after power is good. The PCIRST_N output from Intel 631xESB/632xESB I/O Controller Hub is expected to drive the RSTIN_N input pin on the MCH, which will in turn hold the processor complex in reset via assertion of the FSBxRESET_N 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.

5.9.2.2 Hard Reset Mechanism

Once the MCH 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_N output pin, which is commonly connected directly to the MCH RSTIN_N input pin. The Intel 631xESB/632xESB I/O Controller Hub may be caused to assert PCIRST_N via both software and hardware mechanisms. The MCH will recognize a hard reset any time RSTIN_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_N 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 MCH platform in general will require a full configuration and initialization sequence to be brought back on-line.

5.9.2.3 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.

5.9.3 Targeted Reset Mechanism

The targeted reset is provided 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 Base Specification 2.0* for further description.

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 Data Link Layer Active bit 13 of PEXLNKSTS[9:0] register at 7Eh 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.

5.9.4 FSBxBINIT_N Mechanism

The FSBxBINIT_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 FSBxMCERR_N, 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 FSBxBINIT_N sequence on the FSB.

When FSBxBINIT_N is asserted, 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 FSBxBINIT_N and a general hard reset to recover from a hung system.

Although FSBxBINIT_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 FSBxBINIT_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 FSBxBINIT_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.

5.9.5 Reset Sequencing

Figure 5-35 contains a timing diagram illustrating the progression through the power-on reset sequence. This is intended as a quick reference for system designers to clarify the requirements of the MCH.

Note the breaks in the clock waveform at the top of Figure 5-35, which are intended to illustrate further elapsed time in the interest of displaying a lengthy sequence in a single picture. Each of the delays in the reset sequence is of fixed duration, enforced by either the MCH or the Intel 631xESB/632xESB I/O Controller Hub. In the case of a power-on sequence, the MCH internal "hard" and "core" resets deassert simultaneously. The two lines marked with names beginning "HLA" illustrate the ESI special cycle handshake between the MCH and the Intel 631xESB/632xESB I/O Controller Hub to coordinate across the deasserting edge of the FSBxRESET_N output from the MCH.

Table 5-26 summarizes the durations of the various reset stages illustrated above, and attributes the delays to the component that enforces them.

The fixed delays provide time for subordinate PLL circuitry to lock on interfaces where the clock is withheld or resynchronized during the reset sequence.

Figure 5-35. Power-On Reset Sequence

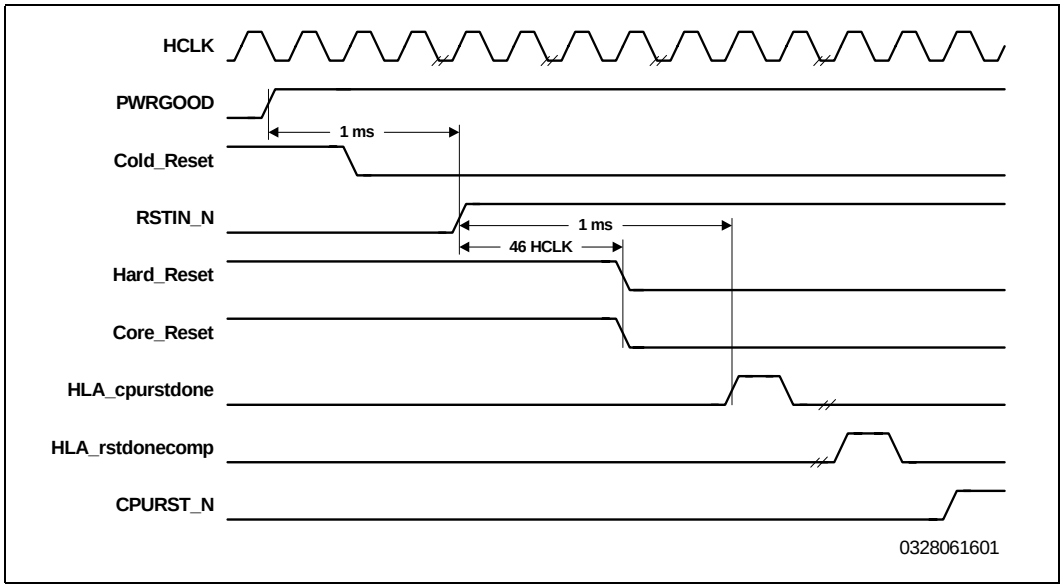


Table 5-26. Reset Sequences and Durations (Sheet 1 of 2)

From	To	Duration	Source	Comment
Power on	PwrGd	>2mS	Platform	Control logic on the platform must ensure that there are at least 2mS of stable power before PwrGd is asserted.
PwrGd	RSTIN_N deassertion	1mS	Intel 631xESB/632xESB I/O Controller Hub	Intel 631xESB/632xESB I/O Controller Hub enforces delay between detecting PwrGd asserted and releasing PCIRST_N (note that Intel 631xESB/632xESB I/O Controller Hub PCIRST_N is directly connected to MCH RSTIN_N).

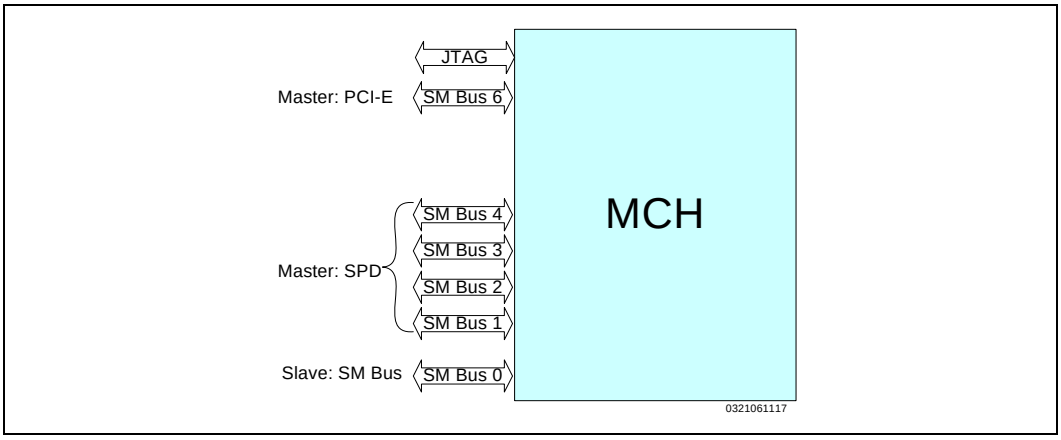
Table 5-26. Reset Sequences and Durations (Sheet 2 of 2)

From	To	Duration	Source	Comment
RSTIN_N deassertion	Hard/Core deassertion	4-6 HCLK	MCH	MCH waits for a common rising edge on all internal clocks, then releases core reset(s).
RSTIN_N deassertion	FSBxRESET_N deassertion	1mS	MCH	MCH enforces delay between RSTIN_N and FSBxRESET_N deassertion. ESI handshake is incremental to the timer.

5.10 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 and configuration of platform devices. Each of these interfaces have dedicated uses as shown in Figure 5-36.

Figure 5-36. MCH SMBus Interfaces



SM Bus 0 is used to support slave accesses. 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.

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.

5.10.1 Remote Error Handling

The MCH supports error escalation via both SMI and MCERR FSB signaling, thus error handling may be implemented in system management mode (SMM) software, machine check architecture (MCA) software, or a combination of the two. Such software could direct a I/O Hub or Intel 631xESB/632xESB I/O Controller Hub with an integrated NIC to “call home” when errors are reported by the MCH. The BMC could then interrogate internal MCH (as well as other SG platform components with SMBus) error logging registers under remote control across the network interface, providing full identification and isolation of reported errors as described elsewhere in this document. The further possibility exists for remotely managed re-configuration via the SMBus target port, as well as remotely managed system reboot via the BMC (if necessary).

5.10.2 Remote Platform Monitoring

The SMBus target also provides a sophisticated BMC the capability to monitor the health of a MCH based platform, such that statistics on correctable error location and frequency may be tracked remotely in an effort to anticipate and prevent more serious failures. The MCH includes significant RAS and Performance Monitoring functionality on both its memory subsystem and its PCI Express interfaces. Some types of errors are expected at a modest frequency within a platform of this complexity, and the MCH provides internal hardware to track the frequency of such errors. These include correctable ECC errors on the memory interface, as well as transient communication errors on the highs peed serial PCI Express interfaces. The BMC could be directed remotely to periodically poll the internal error logging registers of the MCH, permitting a remote management software package to maintain a running profile of error types and frequencies experienced by a Intel® 5400 chipset based platform. Changes in error frequency or type could be flagged by the remote monitoring software to prompt follow-up preventative maintenance on the server. A remote manager could also determine whether the DIMM sparing feature has been invoked, and accordingly generate a notification that a DIMM replacement must be scheduled.

5.10.3 SMBus Physical Layer

The component fabrication process may not support the pull-up voltage required by the SMBus protocol. Therefore, it will be required that voltage translators be placed on the platform to accommodate the differences in driving voltages. The MCH SMBus pads will operate at voltage of 3.3v. Also, the MCH complies with the SMBus SCL frequency of 100 kHz.

5.10.4 SMBus Supported Transactions

The MCH supports six SMBus commands total grouped in three categories:

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

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

Every configuration read or write first consists of an SMBus write *sequence* which initializes the Bus Number, Device, 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 performs a configuration register read) or a write sequence (which performs a configuration register write).

Each SMBus transaction has an 8-bit command that the master sends as part of the packet to instructs the MCH on to handle the data transfers. The format for this command is illustrated in Table 5-27.

Table 5-27. SMBus Command Encoding

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

- The *Begin* bit indicates the first transaction of the read or write sequence. The examples below illustrate when this bit should be set.
- The *End* bit indicates the last transaction of the read or write sequence. The examples below best describe when this bit should be set.
- The *PEC_en* bit enables the 8-bit packet error checking (PEC) generation and checking logic. For the examples below, if PEC was disabled, then there would be no PEC generated or checked by the slave.
- The *Internal Command* field specifies the internal command to be issued by the SMBus slave logic. The MCH supports dword reads and byte, word, and dword writes to configuration space.
- The *SMBus Command* field specifies the SMBus command to be issued on the bus. This field is used as an indication of the length of transfer so that the slave knows when to expect the PEC packet (if enabled).

5.10.5 SMBus Initiated Southbound Configuration Cycles

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

5.10.6 SMBus Error Handling

SMBus Error Handling feature list:

- Errors are reported in the status byte field.
- Errors in Table 5-28 are also coalesced in the FERR and NERR registers.

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

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

Each error bit must be routed to the FERR and NERR registers for error reporting. The status field encoding is defined in the Table 5-28. This field reports if an error occurred. If bits[2:0] are 000b then transaction was successful only to the extent that the MCH is aware. In other words a successful indication here does not necessarily mean that the transaction was completed correctly for all components in the system.

Note: As shown in the read/write diagrams below, the master may also receive a NACK for one of 2 cases. If a previous command is still being internally performed, and the master issues another command, the new command will receive a NACK. Also when a read is performed, the command byte after the address has been received will receive a NACK until the read transaction has internally completed. If COHDEF.SMBUS_SLV_FMEN[0] is set to one, the SMBUS target interface enables clock stretching, and this NACK is not performed.

Table 5-28. Status Field Encoding for SMBus Reads

Bit	Description
7	Internal Time-out. This bit is set if an SMBus request is not completed in 25ms. This field should be reset for each new transaction and flagged if the timeout happens. (This field will return zero if COHDEF.SMBUS_SLV_FMEN[0] = 0.) Note: The SMB timeout value is actually around 31 ms if the core clock frequency is 266 Mhz. At a core clock frequency of 333 Mhz, the SMB timeout value is 25 ms. Note that at 400Mhz, the timeout value decreases to about 21ms.
6	Reserved
5	Internal Master Abort
4	Internal Target Abort
3:1	Reserved
0	Successful

5.10.7 Configuration and Memory Read Protocol

Configuration and memory reads are accomplished through an SMBus write(s) and later followed by an SMBus read. The write sequence is used to initialize the Bus Number, Device, Function, and Register Number for the configuration access. The writing of this information can be accomplished through any combination of the supported SMBus write commands (Block, Word or Byte). The *Internal Command* field for each write should specify Read DWord. The slave interface will NACK the first command byte following the slave address if a previous command is still being processed, unless COHDEF.SMBUS_SLV_FMEN[0] = 1

Note: When COHDEF.SMBUS_SLV_FMEN[0] = 1, clock stretching is enabled, and care must be taken to ensure the PCIE completion timeouts are less than the SMB timeout value, which varies with clock speed.

After all the information is set up, the last write (*End* bit is set) initiates an internal configuration read. The master SMBus will perform the transaction sequence for reading the data. The first command byte of the read sequence will be NACKed until such time that the data is received and the status updated

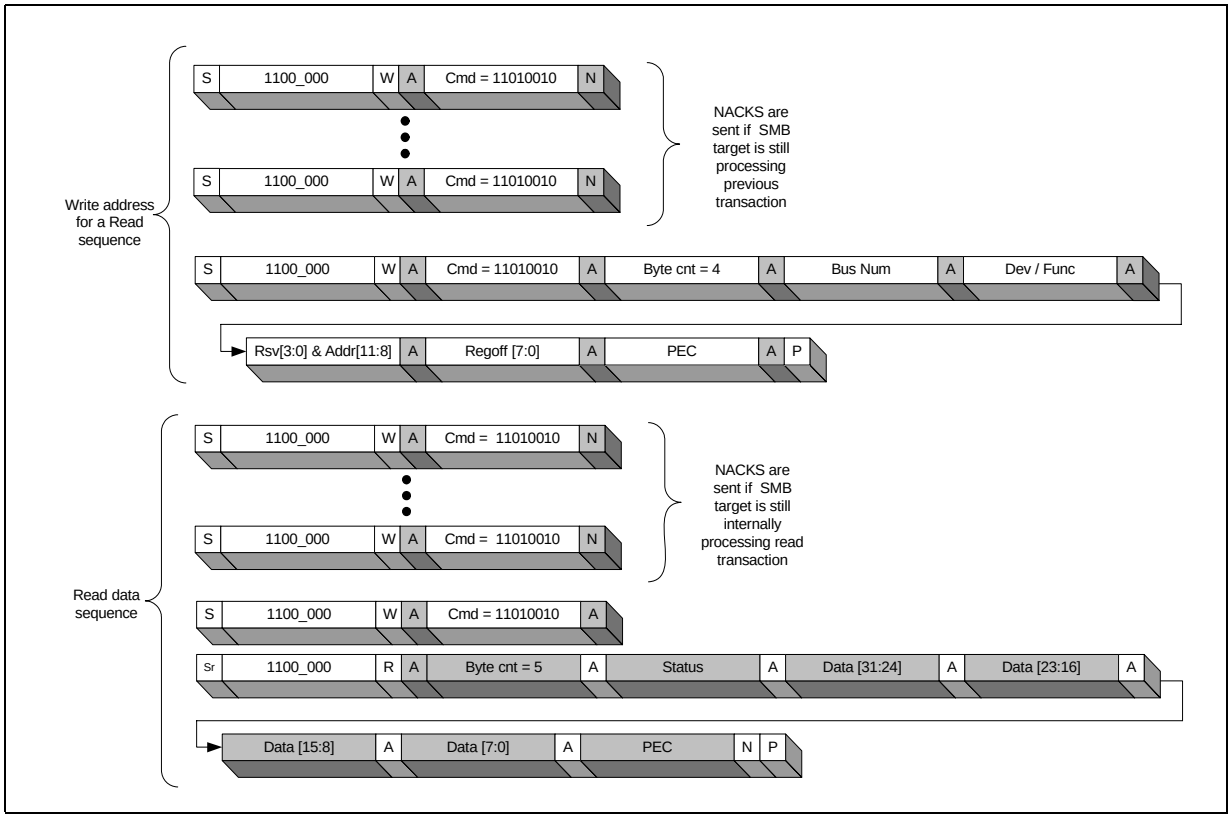
If an error occurs then the status byte will report the results. This field indicates abnormal termination and contains status information such as target abort, master abort, and time-outs.

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

5.10.8 SMB Configuration Read Pictographs

All pictographs assume no clock stretching and instead assert NACK when busy.

Figure 5-37. SMBus Configuration Read using Block Transfers



MCH

Figure 5-38. SMBus Configuration Read using Word sized Transfers

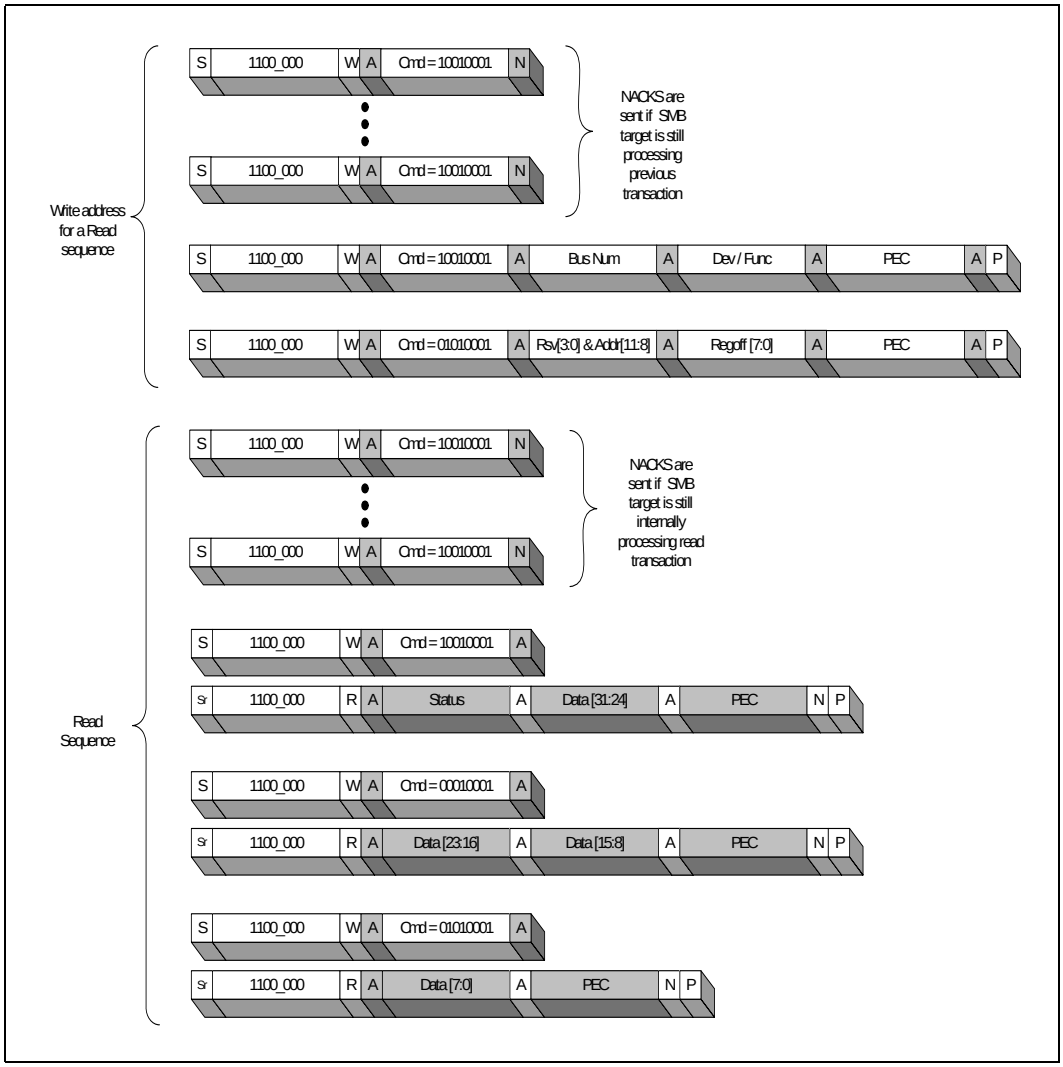
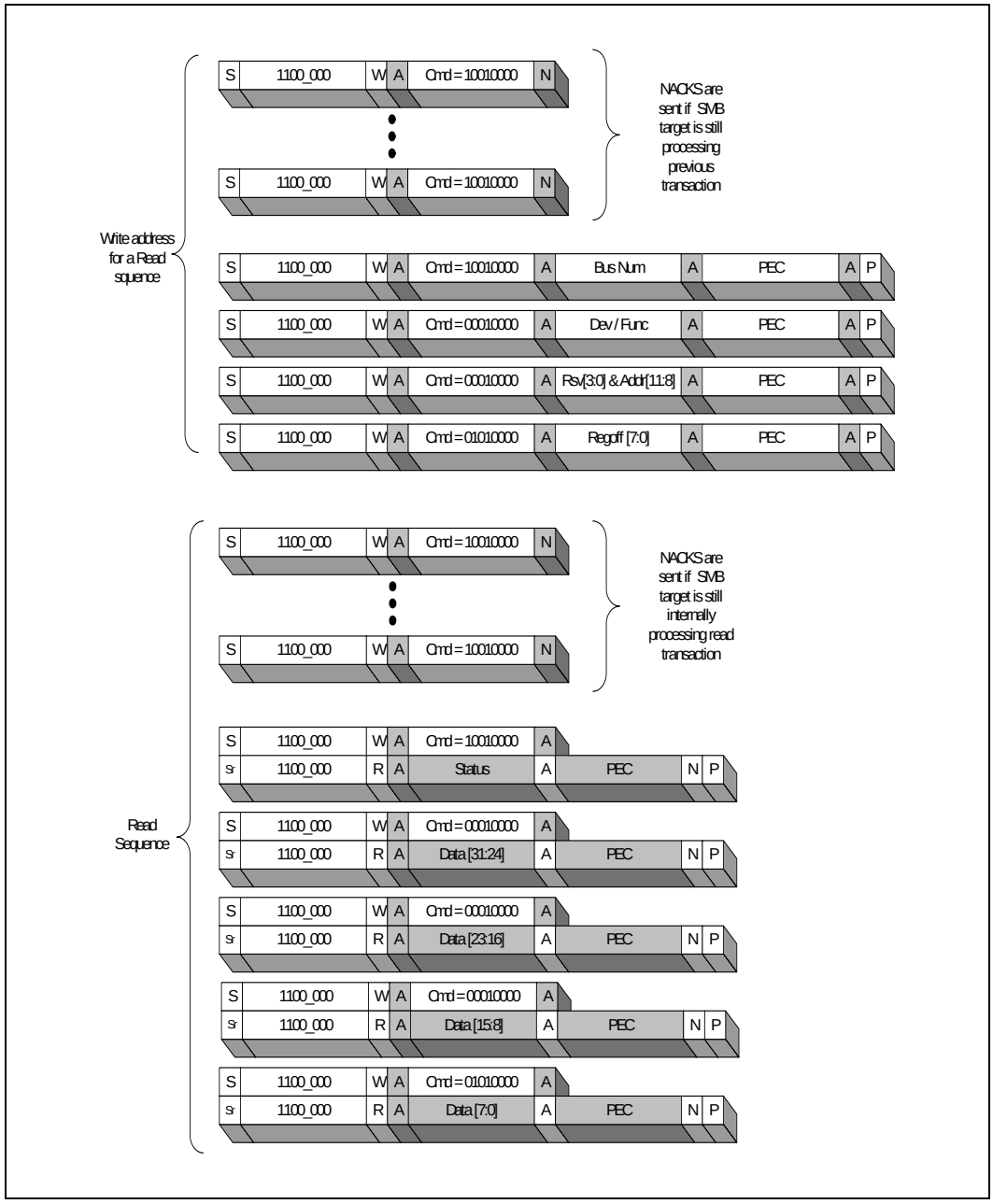


Figure 5-39. SMBus Configuration Read using Byte sized Transfers



5.10.9 Configuration and Memory Write Protocol

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

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

After all the information is set up, the SMBus master initiates one or more writes which sets up the data to be written. The final write (*End* bit is set) initiates an internal configuration write. The slave interface will NACK the first command byte following the slave address if a previous command is still being processed. If an error occurred, the SMBus interface NACKs the last write operation just before the stop bit.

The SMBus interface uses an internal register stack that is filled by the SMBus master before a request to the config master block is made. Shown in Table 5-29 is a list of the bytes in the stack and their descriptions.

Table 5-29. Internal SMBus Protocol Stack

SMBus stack	Description
Command	Command byte
Byte Count	The number of bytes for this transaction
Bus Number	Bus number
Device/Function	Device[4:0] and Function[2:0]
Address High	The following fields are further defined. Address High[7:4] = Reserved[3:0] Address High [3:0] = Address[11:8] : This is the high order PCIe address field.
Register Offset	Register offset lower order 8bit register address
Data3	Data byte 3
Data2	Data byte 2
Data1	Data byte 1
Data0	Data byte 0

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

5.10.10 SMBus Configuration Block Writes

All diagrams below assume no clock stretching, and instead assert NACK when busy

Figure 5-40. SMBus Configuration Write using Block Transfers

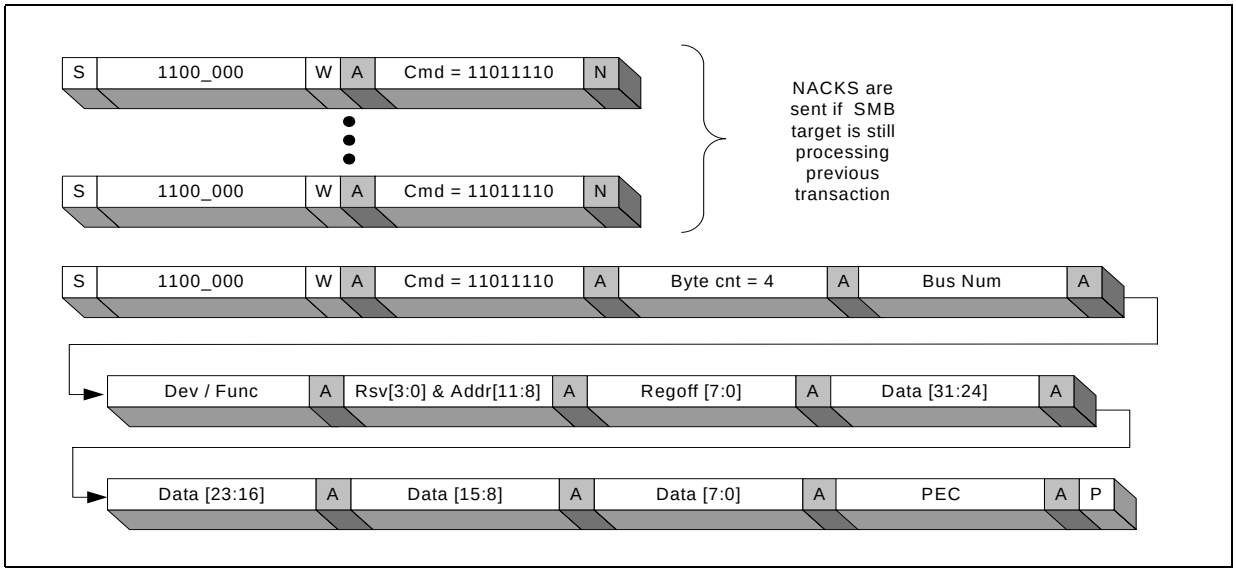


Figure 5-41. SMBus Configuration Write using Word Transfers

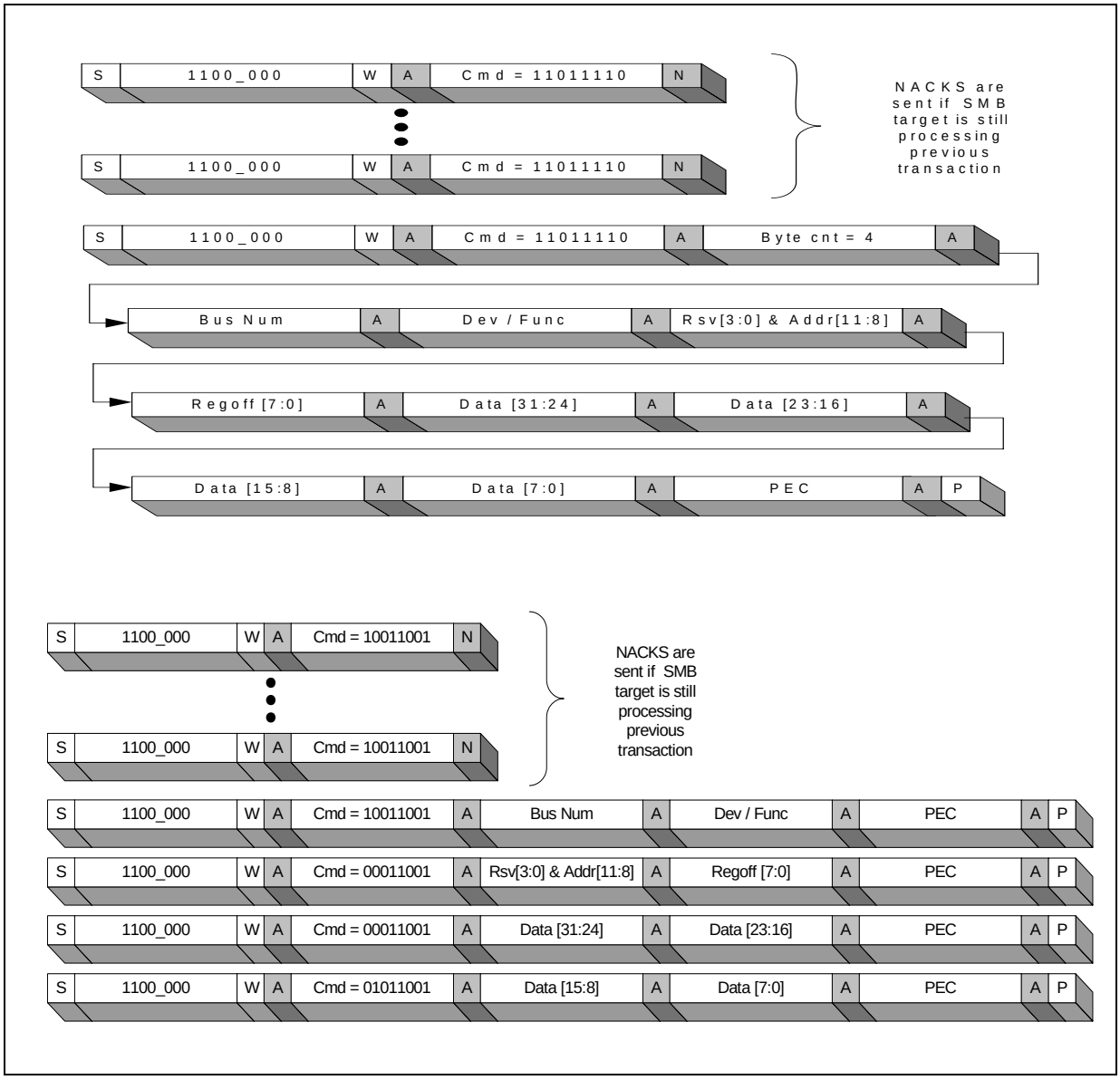
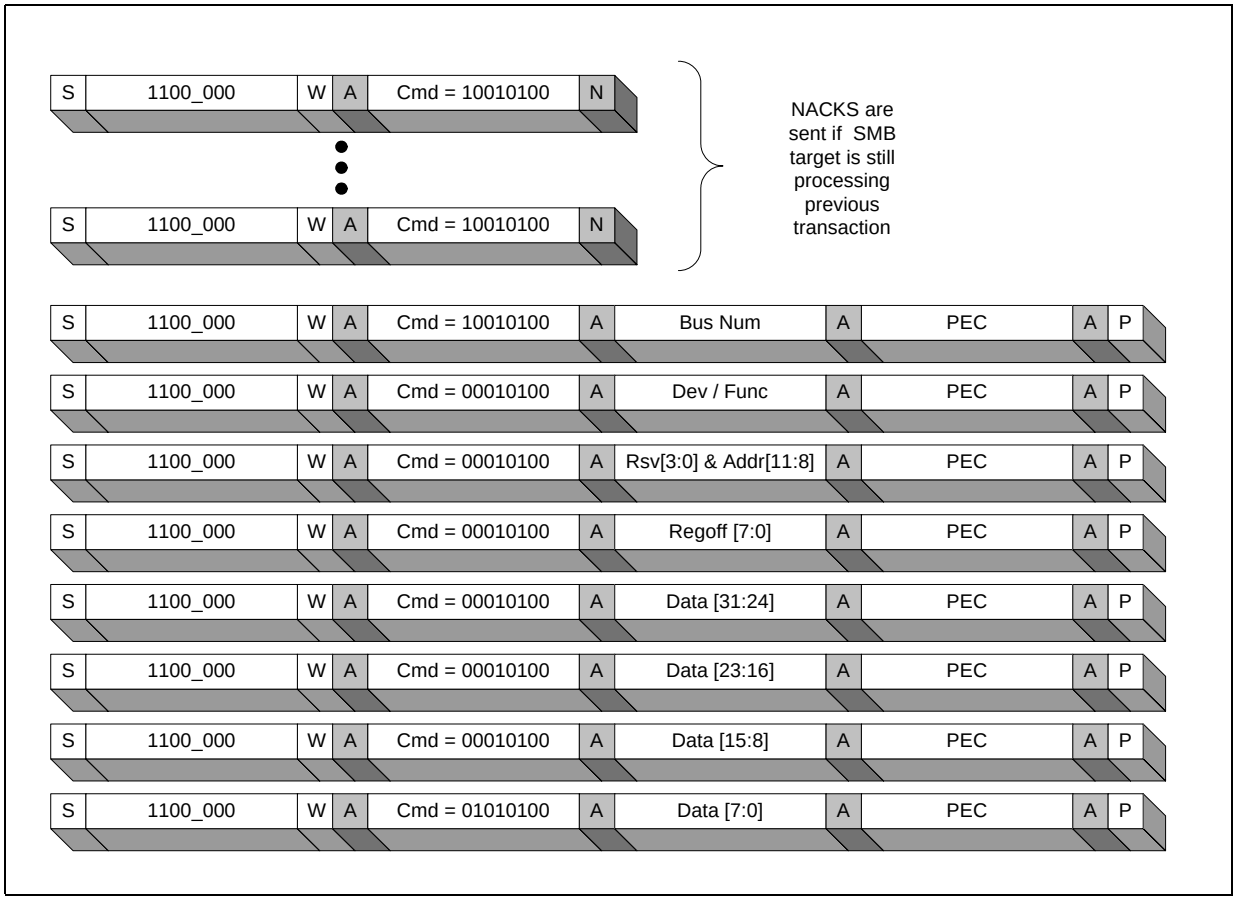


Figure 5-42. SMBus Configuration Write using Byte Transfers



5.10.11 SMBus Interface Reset

- The slave interface state machine can be reset by the master in two ways:
- The master holds SCL low for 25 ms cumulative. Cumulative in this case means that all the "low time" for SCL is counted between the Start and Stop bit. If this totals 25 ms before reaching the Stop bit, the interface is reset.
- The master holds SCL continuously high for 50 us.

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.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.

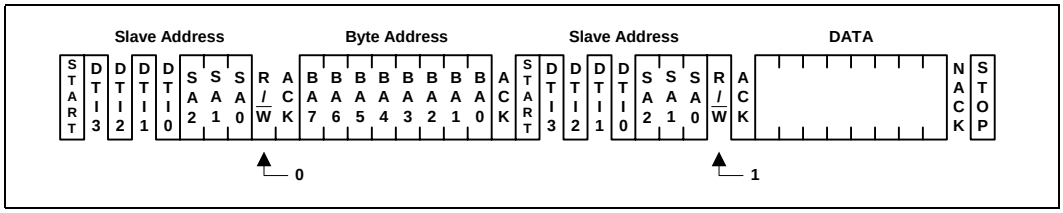
Valid SPD data is stored in the DATA field of the SPD configuration register upon successful completion of the SPDR command (indicated by 1 in the RDO field). Data to be written by an SPDW command is placed in the DATA field of the SPDCMD configuration register.

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.

5.10.11.1 Request Packet for SPD Random Read

Upon receiving the SPDR command, the MCH generates the Random Read Register command sequence as shown in Figure 5-43. 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.

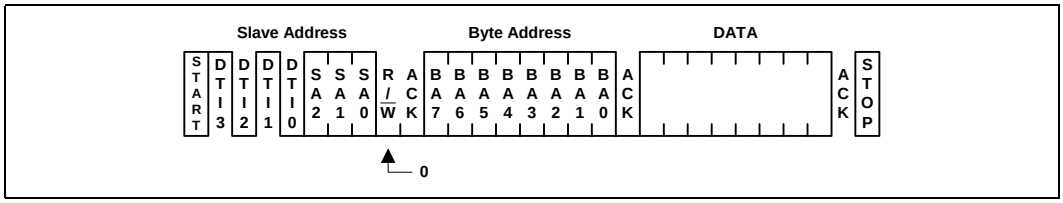
Figure 5-43. Random Byte Read Timing



5.10.11.2 Request Packet for SPD Byte Write

Upon receiving the SPDW command, the MCH generates the Byte Write Register command sequence as shown in Figure 5-44. The MCH indicates that the SIO command has completed by setting the WOD bit of the SPD configuration register to 1.

Figure 5-44. Byte Write Register Timing



5.10.11.3 SPD Protocols

The MCH supports the SPD protocols shown in Table 5-30.

Table 5-30. MCH Supported SPD Protocols

MCH Supported SPD Protocols
Random Byte Read
Byte Write

5.10.11.4 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.

5.11 Intel® Virtualization Technology

5.11.1 Introduction

Intel® Virtualization Technology is the technology that makes a single system appear as multiple independent systems to software. This allows for multiple independent operating systems to be running simultaneously on a single system. The first revision of this technology (Intel® Virtualization Technology (Intel® VT) for IA-32 Intel® Architecture (Intel® VT-x)) added hardware support in the processor to improve the virtualization performance and robustness. The second revision of this specification (Intel® Virtualization Technology (Intel® VT) for Directed I/O (Intel® VT-d)) adds MCH chipset hardware hooks to improve I/O performance and robustness. Intel® 5400 chipset implements Intel VT-d.

The reminder of this chapter talks about the implementation specific aspects of Intel VT-d. For details of Intel VT-d architecture in general refer to *Vanderpool Technology (VT) Chipset Architecture Specification Rev 0.95*.

5.11.2 Intel® VT-d Features Supported

- Support for root entry, context entry and default context
- 48 bit Guest and host address Widths
- Support for only 4 level page walks
- Support for 4 K page sizes
- Support for register based fault recording only and support for MSI interrupts for faults
  - — Support for only one fault recording register
- All Intel VT-d memory structures are treated as coherent
- Support for Intel VT-d on all PCI Express and ESI ports and Intel® QuickData Technology Device
- Support for both leaf and non-leaf caching
- Support for domain-specific device-context cache invalidation
- Support for domain-specific Page Selective Invalidation (PSI). Device-specific PSI is aliased to domain specific PSI.
- Support for boot protection of default page table
- Support for non-caching of invalid page table entries
- Support for hardware based flushing of translated but pending writes on Intel® QuickData Technology Device

5.11.3 Non-Supported Intel VT-d Features

The features that are not available in the MCH are:

- Advanced fault reporting
- Super pages
- 1, 2 or 3 level page walks
- Non-coherent Intel VT-d memory structures
- Address filtering or base-bounds translation

- Device-selective device context or page granular IOTLB invalidation
- No Snoop override functionality via page tables
- The VTD Capability Register incorrectly indicates support for high and low protected memory regions. MCH does not support protected memory regions.
- Hardware flushing of translated reads or writes that are pending global visibility in chipset on PCI Express root ports
- Hardware flushing of translated but pending reads from Intel® QuickData Technology Device
- No caching of prefetched IOTLB entries i.e. additional IOTLB entries fetched within a cacheline of Intel VT-d read will be discarded.

5.11.4 Software View of Intel® QuickData Technology Device Remapping Engine

Intel VT-d architecture allows for either a single software visible Intel® QuickData Technology Device remap engine per integrated device/root port or a common Intel® QuickData Technology Device for a collection of root ports and integrated devices. The MCH will support a separate software visible Intel VT-d engine per PCI Express root port, ESI port and Intel® QuickData Technology Device.

5.11.5 Interrupts, Power Management and RAS

Interrupt handling, power management and RAS architecture do not differ between non-Intel VT-d and Intel VT-d modes.

All Intel VT-d errors listed in the Intel VT-d spec, with one exception, result in a completer abort on the associated PCI Express transaction and the resulting PCI Express error logging/reporting action. The exception case is when the incoming transaction from PCI Express has a GPA address that is beyond the GPA limit (specified via VTCTRL register) and translation is enabled. That would result in a UR response on the associated PCI Express transaction with the resultant PCI Express error logging and reporting. Refer to *Vanderpool Technology (VT) Chipset Architecture Specification Rev 0.95* and *PCI Express Base Specification Revision 2.0* for further details.

Note that errors detected after the address has been translated to the host domain are reported the same way as in the non-Intel VT-d mode. Also, all Intel VT-d errors can be reported to system BMC as well as the OS.

5.11.6 Intel VT-d Fault Handling and Error Logging

Intel® 5400 chipset support address remapping for IO devices otherwise known as Intel Virtualization Technology for I/O Devices (Intel VT-d). All memory type transactions from IO devices received by the root ports and generated by the MCH Intel® QuickData Technology Device can be address remapped when Intel VT-d is enabled. Intel VT-d address translation faults can occur for various software or hardware errors. Table 5-31 indicates how Intel VT-d faults are handled including how the fault errors are logged. Intel VT-d fault errors are logged in the device unit error register for the PCI Express Root Port, ESI, and the Intel® QuickData Technology Device. There are two unit errors that indicate what type of fault occurred: General Intel VT-d Error and Internal HW Intel VT-d Fault Error.

Table 5-31. Intel VT-d Fault Handling Table

Fault Type Class	Intel VT-d Fault Description	Response to requester on Intel VT-d Fault[1]	Error Logging in Unit Error Register in PCIE/ESI/CB
1	Protected Memory Violation when Address Translation is off	UR(CA)	General Intel VT-d Error
2	GPA > VTCTRL.GPA_Limit	UR	No logging[2]
3	Address Translation Fault defined by Intel VT-d spec (includes UR/CA completion status returned for a Intel VT-d memory access)	UR(CA)	No logging[2]
4	Translation Fault due to poisoned data	UR(CA)	General Intel VT-d Error
5	Translation fault due to detected data parity error in Intel VT-d core	UR(CA)	Internal HW Intel VT-d Fault Error
6	HPA > VTCTRL.HPA_Limit register	UR(CA)	No logging[2]
7	HPA targets a Protected Memory Range when Address translation is on	UR(CA)	General Intel VT-d Error

Notes:
1. Some Intel VT-d Fault Type class can be configured to respond with a Completer Abort instead of Unsupported Requests.
2. Intel VT-d fault is logged in the General Intel VT-d Error if Intel VT-d MSI generation is inhibited when the back up MSI mechanism is enabled.

5.12 TPM Security Support

5.12.1 TPM Support 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

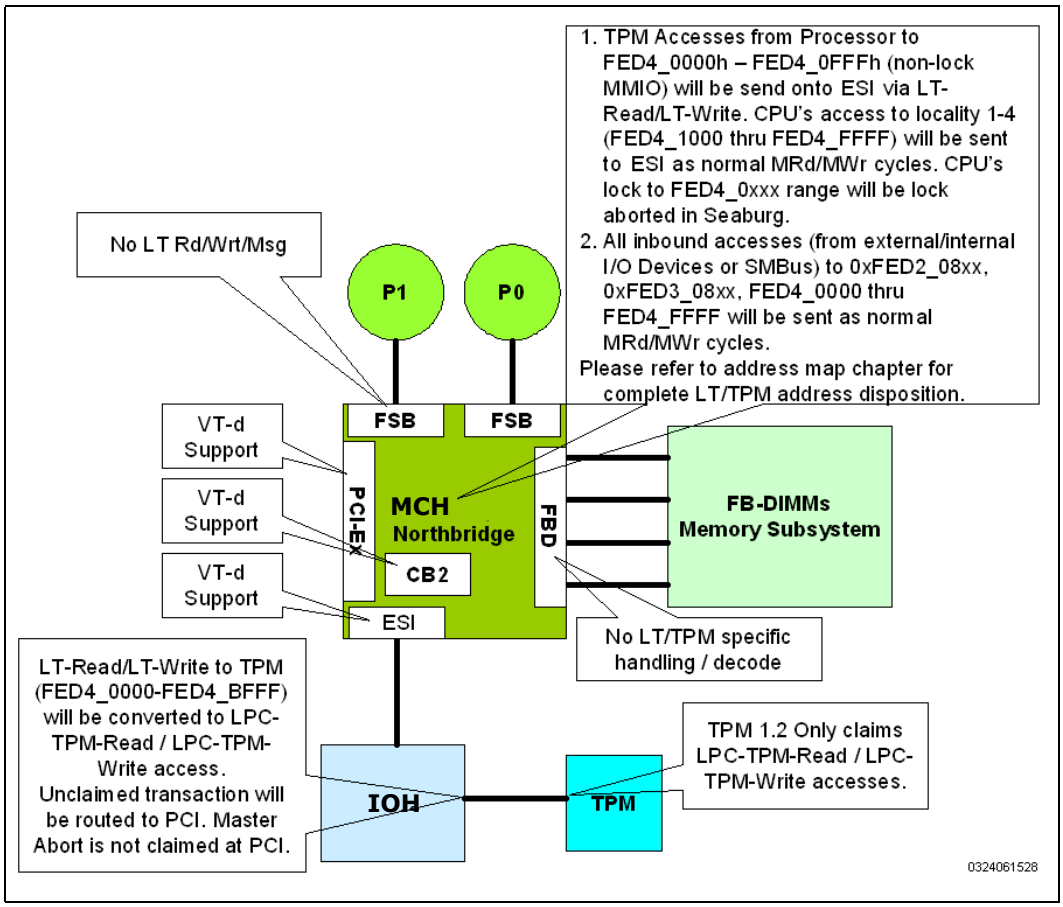
For more details on the usage model or TPM specific information, please visit the Trusted Computing web page at www.trustedcomputinggroup.com and refer to “TCG PC Client Specific TPM Interface Specification, Version 1.2”.

5.12.2 MCH Specific Support for TPM

The MCH does not support Intel® Trusted Execution Technology. Only the TPM 1.2 locality 0 (FED4_0xxx, i.e. Intel Trusted Execution Technology public space) is opened for non-Intel Trusted Execution Technology enabled chipset such as Intel® 5400 chipset. The MCH will convert CPU FED4_0xxxh memory cycle[1] to an ESI Intel Trusted

Execution Technology cycle. All accesses outside of the FED4_0xxxh range will **not** be converted to Intel Trusted Execution Technology cycle. These accesses will be sent as normal memory cycles.

Figure 5-45. MCH TPM Support Overview



1. Lock accesses will be master aborted in MCH lock flow. All other accesses (other than CPU), such as peer-to-peer, Intel® QuickData Technology Device or SMBus accesses will be routed as normal access without the Intel Trusted Execution Technology cycle.

5.12.3 Data Length Handling on TPM access

Intel® 5400 chipset will forward zero length MMIO read/write requests that originate from the FSB targeting the 0xFED4_0xxx range access to an appropriate Intel Trusted Execution Technology-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 length[1] or address alignment restrictions[2] for these MMIO reads/writes when they are mapped[3] to Intel Trusted Execution Technology transactions on ESI. They are passed through as they appear on FSB but as Intel Trusted Execution Technology transactions on the ESI.

5.13 Clocking

The following section describes the MCH Clocks.

5.13.1 Reference Clocks

The BUSCLK, and CORECLK (herein referred to "in aggregate" as "BUSCLK") reference clocks, operating at 133/167/266/333/400 MHz, are supplied to the MCH. These are the processor bus, core, and snoop filter PLL reference clocks. This frequency is common between all processor bus agents. Phase matching between agents is required. The two processor FSBs operate in phase with the core clock.

The FB-DIMM(0/1)CLK reference clocks, (herein referred to as FB-DIMMCLK) operating at half the DDR2 frequency (operating at the SDRAM command-clock frequency, which is the FB-DIMM packet frequency), are supplied to the MCH. This is the FB-DIMM PLL reference clock. This frequency is common between the MCH and DIMMs. Phase matching between agents is not required (plesiochronous). The MCH and DIMMs treat this frequency domain synchronously. The FB-DIMM unit-interval (UI) PLL outputs 12x the FB-DIMMCLK frequency. E.g. For DDR2 667 MHz DIMMs, the FB-DIMMCLK frequency is 333 MHz and the UI (link) frequency is 4.0 GHz.

The PECLK reference clock, operating at 100 MHz, 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 for GEN I operation and outputs 5.0 GHz for GEN I or GEN II operation. The phit clock frequency must be tightly matched (plesiochronous 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 BUSCLK and FB-DIMMCLK 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.

1. e.g. 1-4B reads/writes
2. e.g. Address should be double word (DW) aligned.
3. The MCH does not abort a zero length read to 0xFED4_0xxx range internally nor does it split an unaligned DW address into 2 or more transactions on ESI.

Table 5-32. MCH Frequencies for Processor and Core

Core	Domain	Frequency	Reference Clock
			BUSCLK
266 MHz	BUSCLK	266 MHz	
	FSB 1X		
	FSB 2X	533 MHz	
	FSB 4X	1,067 MHz	
333 MHz	BUSCLK	167 MHz	
	FSB 1X		
	FSB 2X	333 MHz	
	FSB 4X	667 MHz	
333 MHz	BUSCLK	333 MHz	
	FSB 1X		
	FSB 2X	667 MHz	
	FSB 4X	1,333 MHz	
400 MHz	BUSCLK	200 MHz	
	FSB 1X		
	FSB 2X	400 MHz	
	FSB 4X	800 MHz	
400 MHz	BUSCLK	400 MHz	
	FSB 1X		
	FSB 2X	800 MHz	
	FSB 4X	1,600 MHz	

Table 5-33. MCH Frequencies for Memory

DDR	Domain	Frequency	Reference Clock
533 MHz	FB-DIMM UI	3.2 GHz	FB-DIMMCLK
	FB-DIMM packet	266 MHz	
	FB-DIMMCLK	133 MHz	
667 MHz	FB-DIMM UI	4.0 GHz	
	FB-DIMM packet	333 MHz	
	FB-DIMMCLK	167 MHz	
640 MHz	FB-DIMM UI	3.84 GHz	
	FB-DIMM packet	320 MHz	
	FB-DIMMCLK	160 MHz	
800 MHz	FB-DIMM UI	4.8 GHz	
	FB-DIMM packet	400 MHz	
	FB-DIMMCLK	200 MHz	

Table 5-34. MCH Frequencies for PCI Express

Domain	Frequency	Reference Clock
PCI Express GEN II phit	2.5 GHz or 5.0 GHz	PECLK
PCI Express GEN I phit	2.5 GHz	PECLK
PCI Express flit	250 MHz	
PECLK	100 MHz	

5.13.2 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.

5.13.3 SMBus Clock

The SMBus clock is synchronized to the core clock. Data is driven into the SG 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.

5.13.4 GPIO Serial Bus Clock

The transmitted 100 KHz Virtual Pin Interface (VPI) clock (one of the SCL[4:0]'s) is derived from the core clock.

5.13.5 Clock Pins

Table 5-35. Clock Pins (Sheet 1 of 2)

Pin Name	Pin Description
CORECLKP	Processor bus clock
CORECLKN	Processor bus clock (Complement)
PE{0/1}CLKP	PCI Express clock
PE{0/1}CLKN	PCI Express clock (Complement)
FB-DIMM{01/23}CLKP	FB-DIMM clocks
FB-DIMM{01/23}CLKN	FB-DIMM clocks (Complement)
VCCAFB-DIMM{01/23}PLL, VCCAFB-DIMM{01/23}PLL18	Analog power supply for FB-DIMM PLLs
VCCAPE{0/1}PLL125, VCCAPE{0/1}PLL18	Analog power supply for PCI Express PLLs
VSSAPE{0/1}PLL	Analog ground for PCI Express PLLs
TCK	TAP clock
SPD{0/1/2/3}SMBCLK	FB-DIMM Channel 0/1/2/3 SMB clocks
GPIOSMBCLK	GPIO (Virtual Pin Port) clock

Table 5-35. Clock Pins (Sheet 2 of 2)

Pin Name	Pin Description
CFGSMBCLK	SMBus clock
XDPCLK1X	Debug bus clock reference
XDPSTBP_N	Debug bus data strobe
XDPSTBN_N	Debug bus data strobe (Complement)
FSB{0/1}STB_P[3:0]	Processor bus data strobes
FSB{0/1}STB_N[3:0]	Processor bus data strobes (Complements)
FSB{0/1}ADSTB_N[1:0]	Processor bus address strobes

5.13.6 High Frequency Clocking Support

5.13.6.1 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%.

5.13.6.2 Stop Clock

PLLs in the MCH cannot be stopped.

5.13.6.3 Jitter

PECLK cycle-to-cycle jitter must be less than 150ps (+/- 75ps).

The FB-DIMM UI clocks are produced by PLL's that multiply the FB-DIMMCLK frequency by 24. The PCI Express phit clocks are produced by PLL's 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.

Strong recommendation: FB-DIMMCLK cycle-to-cycle jitter delivered to the package ball should be less than or equal to 50ps (+/- 25ps). The long-term-input-to-output-jitter envelope of the on-die FB-DIMMCLK PLL is 60ps (+/- 30ps) over 6-sigma (165 million cycles).

BUSCLK cycle-to-cycle jitter delivered to the package ball must be less than 150ps (+/- 75ps). The long-term-input-to-output-jitter envelope for the output of the on-die core-clock PLL is 125ps (+/- 62.5ps) over 6-sigma (165 million cycles).

MCH must internally accept a long-term drift of +/-776ps between the memclk and FB-DIMMclk domains without timing violation. The platform exceeding this envelope may result in a fast reset. A static phase offset between MEMCLK and FB-DIMMCLK is acquired at the FB-DIMM.TS0 -> FB-DIMM.TS1 transition. This phase relationship is the center of the long-term phase drift envelope. When operating with a 1:1 gear ratio between the core and the FB-DIMM interface, these relationships also apply to the relationship between BUSCLK and FB-DIMMCLK.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.

5.13.6.4 PLL Lock Time

All PLLs should be locked by PWRGOOD signal assertion. 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. Many JTAG private registers are dependent on core PLL-generated clocks.

5.13.6.5 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.

5.13.6.6 Analog Power Supply Pins

The MCH incorporates six 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.

5.14 Error List

This section provides a summary of errors detected by the Intel® 5400 chipset MCH. 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 5-36. Intel® 5400 Chipset Error List (Sheet 1 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
	Memory Errors				
M1	Memory Write error on non-redundant retry or FB-DIMM configuration Write error on retry	SG detected a corrupted acknowledgement on a retry of a non-redundant memory write or SG detected a corrupted acknowledgement on a retry of an FB-DIMM configuration write	Fatal	Figure 5-17	See Figure 5-6 through Figure 5-15.
M2	Memory or FB-DIMM configuration CRC read error	SG detected a corrupted CRC on a non-redundant retry at a memory or FB-DIMM configuration read.	Fatal	Figure 5-17	See Figure 5-6 through Figure 5-15.
M4	Uncorrectable Data ECC Error on FB-DIMM Replay	SG detected an uncorrectable data ECC error during replay of the head of the FB-DIMM replay queue	Uncorr	Figure 5-17	See Figure 5-6 through Figure 5-15.

Table 5-36. Intel® 5400 Chipset Error List (Sheet 2 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
M5	Aliased Uncorrectable Non-Mirrored Demand Data ECC Error	SG determined that a normally "correctable" error could be an aliased (x8 only) full device failure plus an additional single bit error.	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M6	Aliased Uncorrectable Mirrored Demand Data ECC Error	In mirrored mode, SG determined that a normally "correctable" error could be an aliased (x8 only) full device failure plus an additional single bit error.	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M7	Aliased Uncorrectable or Spare-Copy Data ECC Error	During a Sparing copy read from the failing DIMM, SG determined that a normally "correctable" error could be an aliased (x8 only) full device failure plus an additional single bit error.	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M8	Aliased Uncorrectable Patrol Data ECC Error	During a Patrol Scrub, SG determined that a normally "correctable" error could be an aliased (x8 only) full device failure plus an additional single bit error.	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M9	Non-Aliased Uncorrectable Non-Mirrored Demand Data ECC Error	SG detected uncorrectable data with good CRC.	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M10	Non-Aliased Uncorrectable Mirrored Demand Data ECC Error	In mirrored mode, SG detected uncorrectable or poisoned data with good CRC.	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M11	Non-Aliased Uncorrectable Spare-Copy Data ECC Error	SG detected uncorrectable data with good CRC from the failing DIMM rank during a sparing copy.	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M12	Non-Aliased Uncorrectable Patrol Data ECC Error	During a patrol scrub, SG detected uncorrectable data with good CRC.	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M13	Memory Write error	SG detected a corrupted acknowledgement on a first attempt at a memory write	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M14	FB-DIMM Configuration Write error	SG detected a corrupted acknowledgement on a first attempt at an FB-DIMM configuration write	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M15	Memory or FB-DIMM configuration CRC read error	SG detected a corrupted CRC on: a non-redundant first attempt, a redundant first attempt, or a redundant retry at a memory or FB-DIMM configuration read.	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.

Table 5-36. Intel® 5400 Chipset Error List (Sheet 3 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
M16	Channel fail-over occurred	SG started operating in fail-over mode on a branch because of a bit-lane failure on a channel	Rec	NBTRL, NBTRH	Operate in reduced-RAS fail-over mode
M17	Correctable Non-Mirrored Demand Data ECC Error.	SG detected correctable data.	Corr	Figure 5-17	See Figure 5-6 through Figure 5-15.
M18	Correctable Mirrored Demand Data ECC Error	SG detected correctable data.	Corr	Figure 5-17	See Figure 5-6 through Figure 5-15.
M19	Correctable Spare-Copy Data ECC Error	SG detected correctable data from the failing DIMM rank during a sparing copy.	Corr	Figure 5-17	See Figure 5-6 through Figure 5-15.
M20	Correctable Patrol Data ECC Error	During a patrol scrub, SG detected correctable data.	Corr	Figure 5-17	See Figure 5-6 through Figure 5-15.
M21	FB-DIMM Northbound parity error on FB-DIMM Sync Status	SG detected a northbound parity error on a Sync Status	Corr	Figure 5-17	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 parity. **WARNING:** Possible double DIMM configuration command execution may incur undesirable side-effects.
M22	SPD protocol Error	SG detected an SPD interface error.	Corr	Figure 5-17	Successive correction attempts performed by software.
M23	Non-Redundant Fast Reset Timeout	SG hit a timeout on a non-redundant fast reset	Fatal	Figure 5-17	See Figure 5-6 through Figure 5-15. During redundancy loss, the fast reset failure on the degraded branch should be ignored as a fatal error and cleared. "Real" errors only appear on the non-degraded branch. This error may not occur in non-1:1 gearing modes when both channels fail a redundant write-recovery.
M24	Refresh error	SG detected a corrupted acknowledgement on a refresh	Rec	Figure 5-17	Refresh acknowledgements are not checked when the refreshes are co-incident with read data. The memory controller can oversubscribe gearing bandwidth. Refresh acknowledgements are not checked when read data plus acknowledgement bandwidth exceeds the gearing bandwidth. E.g., 4:5 gearing requires one out of every five northbound frames to be dropped. If read data consumes four of the five frames, then a refresh acknowledgement coinciding with the fifth frame will not be checked.

Table 5-36. Intel® 5400 Chipset Error List (Sheet 4 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
M25	Memory Write error on redundant retry	SG detected a corrupted acknowledgement on a redundant retry at a memory write	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M26	Redundant Fast Reset Timeout	SG hit a timeout on a redundant fast reset	Rec	Figure 5-17	See Figure 5-6 through Figure 5-15.
M27	Correctable Error Threshold Exceeded	Correctable Error Threshold Exceeded	Corr	Figure 5-17	Start DIMM-spare copy
M28	DIMM-Spare Copy complete	DIMM-Spare copy completed normally	Corr	Figure 5-17	No Action
M29	DIMM-isolation complete	DIMM-isolation complete	Corr	Figure 5-17	No Action
	FSB Errors				
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-hardfail response) NOTE: If an FSB address is masked during debug, the MCH may log this error inadvertently since the checking logic does not have visibility into the original address signals. Hence Software should mask the F1 error when any bits of FSBGBLMSK0/1 is not a zero.
F2	Unsupported Request or data size on FSB.	MCH detected an FSB Unsupported. MCH only detects this error caused by CPUs. Specifically, F2 is recorded for the following cases: - Unsupported txns - DEN not asserted when it should be '1' - Unsupported lengths	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. Data parity is signalled for each fsb clock based on the contents of the 4 data phases. 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

Table 5-36. Intel® 5400 Chipset Error List (Sheet 5 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
F9	FSB protocol Error	MCH detected FSB protocol error, for example, Hitm on BIL and HitM on EWB and HITM on BLW (lock write)	Fatal	FERR_FAT_FSB/NERR_FAT_FSB. NRECFSB, NRECFSB_ADDRH, NRECFSB_ADDRL for FERR only.	Complete transaction on FSB with response (IWB as in the example): Note: For Woodcrest based systems, BILs can be issued for non-shared state and hence it is not reported as an error.
	PCI Express Errors				
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 IO 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 Error'ed Packet
IO2	PCI Express - Received Unsupported Request	Received an unsupported request, similar to master abort.	Default= UnCorr (Check UNCERRSEV)		Log Header of Packet. The header log may be not valid for all cases of unsupported request. 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 southbridge.	Default= UnCorr (Check UNCERRSEV)		Log Header of Error'ed Packed 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 Error'ed Packet Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal). MCH PCI Express ports and potential the MCH will be left in an undefined/corrupted state.

Table 5-36. Intel® 5400 Chipset Error List (Sheet 6 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
IO6	PCI Express - Completion Timeout	Pending transaction did not complete within the time limit.	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEXPEX_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 IO errors based on UNCERSEV(	Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)
IO7	PCI Express - Completer Abort	Received return CA status for unknown error on the component. This is equivalent to a target abort on PCI.	Default= UnCorr (Check UNCERRSEV)		Log Header of Error'ed Packet 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 Error'ed Packet 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 Error'ed Packet Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal)
IO10	PCI Express - Receive Buffer Overflow Error	Receiver gets more data or transactions than credits allow.	Default= Fatal (Check UNCERRSEV)		Log Header of Error'ed Packet 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 CORRERSTS for their respective Error Types.
IO12	PCI Express - Receiver Error	Log Header of Error'ed Packet	Corr		Log CORRERSTS for their respective Error Types.
IO13	PCI Express - Bad TLP Error	Received bad CRC or a bad sequence number in a transport layer packet.	Corr		Log CORRERSTS for their respective Error Types.
IO14	PCI Express - BAD DLLP	Received bad CRC in a data link layer packet.	Corr		Log CORRERSTS for their respective Error Types.
IO15	PCI Express - Replay_Num Rollover	Replay maximum count for the Retry Buffer has been exceeded.	Corr		Log CORRERSTS for their respective Error Types.
IO16	PCI Express - Replay Timer Timeout	Replay timer timed out waiting for an Ack or Nak DLLP.	Corr		Log CORRERSTS for their respective Error Types.
IO17	PCI Express - Received Correctable Error Message	MCH received a correctable error message from the south bridge.	Corr		Log CORRERSTS for their respective Error Types.
IO18	ESI reset timeout	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	De-Assert processor RESET#. Necessary to prevent processor thermal runaway.

Table 5-36. Intel® 5400 Chipset Error List (Sheet 7 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
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.
IO20	Advisory Non-Fatal Error	PCI Express role based Error reporting	Corr	Log PEX_NF_FERR/NERR, CORRERSTS	Log CORRERSTS for their respective Error Types.
IO21	ACS Violation Error	PCI Express Port has detected a violation of the access control mechanism Unsupported feature in MCH	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (UNCERRSEV)	Log Header of Error'ed Packet Check corresponding bit in UNCERRSEV register for severity level (Fatal or Non Fatal) MCH does not expect to received this type of error
IO22	Internal Header/Address/ Control Parity Error	PCI Express Port Internal Header/Address/Control Parity Error	Fatal	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR	Log Errors in PEX_UNIT_STS register for respective Error Types. Initiate Stop & Scream mechanism to bring down PCI Express port. Do not log error IO27. A hard reset is required.
IO23	General VTd Error	General VTd Address Translation Table Fault Error (Includes Protected Memory Violations or Poisoned Data Error from DM cluster)	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (PEX_UNIT_SEV)	Log Errors in PEX_UNIT_STS register for respective Error Types.
IO24	Outbound Poisoned TLP Error	PCIE Port issues an Outbound poisoned TLP Error	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (PEX_UNIT_SEV)	Log Errors in PEX_UNIT_STS register for respective Error Types.
IO25	Received UR Completions Status Error	PCIE port received an UR completions status	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (PEX_UNIT_SEV)	Log Errors in PEX_UNIT_STS register for respective Error Types.
IO26	Received CA Completion Status Error	PCIE port received an CA completion status	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (PEX_UNIT_SEV)	Log Errors in PEX_UNIT_STS register for respective Error Types.

Table 5-36. Intel® 5400 Chipset Error List (Sheet 8 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
IO27	Stop and Scream Error	PCIE port stopped and screamed	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (PEX_UNIT_SEV)	Log Errors in PEX_UNIT_STS register for respective Error Types. If outbound poisoned TLP (IO24) and Stop&Scream is enabled, initiate Stop & Scream mechanism to bring down PCI Express port
IO28	BW Change Error	Link Bandwidth changed (software initiated or hardware initiated for reliable operation)	Default= Corr (Check UNCERRSEV)	Log PEX_NF_COR_FERR/NERR	Log Errors in PEX_UNIT_STS register for respective Error Types.
IO29	MSI Address Error	MSI Address violation that does not target 0xFEEx_xxxx	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (PEX_UNIT_SEV)	Log Errors in PEX_UNIT_STS register for respective Error Types.
IO30	VTd Internal HW Error	VTd Internal HW Parity Fault Error (Detected Hardware Parity Error)	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (PEX_UNIT_SEV)	Log Errors in PEX_UNIT_STS register for respective Error Types.
IO31	Received UR Posted Request Error (non-P2P)	Received UR Response from CE on inbound Posted Request Error	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (PEX_UNIT_SEV)	Log Errors in PEX_UNIT_STS register for respective Error Types.
IO32	Received CA Posted Request Error (non-P2P)	Received CA Response from CE on inbound Posted Request Error	Default= UnCorr (Check UNCERRSEV)	Log PEX_FAT_FERR/NERR or PEX_NF_COR_FERR/NERR based on their respective Error types and Severity (PEX_UNIT_SEV)	Log Errors in PEX_UNIT_STS register for respective Error Types.
IO33	Link Autonomous Change Error	Link Autonomous Bandwidth Change Error	Default= Corr	Log PEX_NF_COR_FERR/NERR	Log Errors in PEX_UNIT_STS register for respective Error Types.
	Internal Errors				
B1	MCH -Parity Error from CE	MCH detected internal CE parity error.	Fatal	FERR_FAT_INT/NERR_FAT_INT	No additional info logged.
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 Set & Tag on FERR. Current transaction is dropped by SF.
B3	MCH-Coherency Violation EWB Error	MCH detected a cache coherency protocol error for EWB. Any requestor not in "E/M" state in the SF.	Fatal	FERR_FAT_INT/NERR_FAT_INT, NRECINT and NRECSF	Log CE entry on FERR This applies to SF enable mode only

Table 5-36. Intel® 5400 Chipset Error List (Sheet 9 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
B4	Virtual Pin Interface Error	MCH detected an error on the virtual pin interface	Fatal	FERR_FAT_INT/NERR_FAT_INT and NRECINT	Log VPP Block on FERR. See PCIE unit error register: PEX_UNIT_STS[0].
B5	MCH-Address Map Error	MCH detected address mapping error due to software programming error. The errors are described in system address map chapter.	UnCorr	FERR_NF_INT/NERR_NF_INT and RECINT	Log CE Entry on FERR. 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 Way/Aff, Hit/Miss, Set, Tag, State and Presence vector on FERR. Entry is corrected.
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 Set & Tag on FERR. Lookup with error does not return a hit for that location.
B9	Illegal Access Error	A 64 bit access to this 32-bit only register has been attempted.	UnCorr	FERR_NF_INT/NERR_NF_INT RECINT	Log CE Entry on FERR. Illegal writes will not complete. Illegal reads will return all 1's.
B10	DM Parity Error	DM Parity Error	UnCorr	FERR_NF_INT/NERR_NF_INT RECINT	Log DM entry on FERR. Data returned will be poisoned
B11	Victim RAM parity error B0 Only: This error does not occur.	Victim RAM has encountered a parity error on a lookup	UnCorr	FERR_NF_INT/NERR_NF_INT RECSF	Log Aff, Set, and Tag on FERR. Transaction may evict wrong victim, possibly affecting performance. *NOTE:* If 96wayinv, RECSF[SET] = points to Vram entry. If 24wayinv, RECSF[AFF,SET] = points to Vram entry.
B12	Parity Protected Register	Parity error detected on registers accessible through configuration space.	Fatal	FERR_FAT_INT2/NERR_FAT_INT2	Component configuration integrity has failed. Configurable chip freeze if enabled will commence. No additional info logged.
B14	Scrub DBE	Scrub Lookup returned DBE (double bit error)	Fatal	FERR_FAT_INT2/NERR_FAT_INT2	Scrub detects uncorrectable error, notifies CE. No additional info logged.
B16	SMBus Error Status	SMB target reports error	UnCorr	FERR_NF_INT2/NERR_NF_INT2	System Management error occurred that may be related to other system transactions. No additional info logged.
B17	JTAG/TAP Error Status	JTAG target reports error	UnCorr	FERR_NF_INT2/NERR_NF_INT2	JTAG error occurred that may be related to other system transactions. No additional info logged.
B18	PerfMon Task Completion	OR of PerfMon Engine status bits	UnCorr	FERR_NF_INT2/NERR_NF_INT2	One of the PerfMon engines had a successful compare or an overflow has occurred.Software can use this bit instead of polling. No additional info logged.

Table 5-36. Intel® 5400 Chipset Error List (Sheet 10 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
B19	Scrub SBE	Scrub Lookup returned SBE (single bit error)	Corr	FERR_NF_INT2/NERR_NF_INT2	Tag entry is corrected, and software is notified. No additional info logged.
B20	Configuration Write Abort	Configuration write to internal register with data parity error did not alter configuration state.	UnCorr	FERR_NF_INT/NERR_NF_INT RECINT	Log DM entry on FERR. Configuration write with data errors was not allowed to alter configuration state (aborted). Non-fatal because register corruption did not occur.
B21	Illegal Way	Request to an unsupported way	Fatal	FERR_FAT_INT/NERR_FAT_INT	No additional info logged. SF ignores current transaction, and notifies CE.
B22	Victim ROM parity error	Victim ROM has encountered a parity error on a lookup	UnCorr	FERR_NF_INT/NERR_NF_INT RECSF	Log, Way/Aff, Set, and Tag on FERR. Transaction may evict wrong victim, possibly affecting performance. *NOTE*: If 96wayrand, RECSF[AFF, WAY] = ROM Aff/Way entry. If 24wayrand, RECSF[WAY] = ROM Way entry. If 24wayrand, RECSF[AFF] = FSB DID.
B23	Vt Unaffiliated Port Error (IOG)	Control parity error in Vt queues structure	Fatal	FERR_FAT_INT/NERR_FAT_INT NRECINT	Log Vt IOG Ports on FERR. Vt control parity errors non-specific to a PCIE port
B24	Reserved				
B25	HiSMM/TSEG	CPU Access to HiSMM/TSEG when not allowed	Fatal	FERR_NF_INT/NERR_NF_INT	No additional info logged.
B26	Reserved				
B27	Break from S1	Request received when in S1 (CE)	Corr	FERR_NF_INT2/NERR_NF_INT2	No additional info logged
	Thermal Errors				
TH5Err	Deadman Timeout (on Cooling Update)	The memory controller failed to observe two writes to MTCOOL within the THRTCTRLA.DMPER timeout.	REC	FERR_NF_THR/NERR_NF_THR	Violation of minimum cooling update frequency / Slam memory throttling limits to THRTLOW.THRTLOWLM0 and THRTMID.THRTMIDLM0.

Table 5-36. Intel® 5400 Chipset Error List (Sheet 11 of 11)

ERR # in MCH	Error Name	Definition	Error Type	Log Register	Cause / Actions
TH3Err	Throttling Event (THERMALERT#)	The temperature at the thermal sensor exceed the TSTHRHI threshold or memory throttling occurred	REC	FERR_NF_THR, NERR_NF_THR, CTSTS.THRMALRT, THRTSTS.THRMALRT	While CTCOOL is cleared, die temperature reached TSTHRHI but has not subsequently fallen to TSTHRLO / Throttle FSB's to TSTHRRFSB. Throttle PEX's to TSTHRRPEX. Assert THERMALERT_N Log CTSTS.THRMALRT on FERR. or Memory demand exceeded the bandwidth threshold defined by the THRTLOW or THRTMID vector in conjunction with MTCOOL, or THRTHI / Throttle memory to the appropriate threshold. Assert THERMALERT_N Log THRTSTS.THRMALRT on FERR.
TH2Err	GT Tmid	Intelligent throttling is disabled and the thermal sensor transitions from "below Tmid" to "above Tmid".	Fatal	FERR_FAT_THR/NERR_FAT_THR	No additional info logged.
TH1Err	Catastrophic On-Die Thermal Event (THERMTRIP#)	The temperature at the thermal sensor reached the TSTHRCATA threshold	Fatal	FERR_FAT_THR/NERR_FAT_THR	TSVAL reached TSTHRCATA /Assert THERMTRIP#

§

6 Electrical Characteristics

6.1 Absolute Maximum Ratings

Table 6-1 lists the maximum environmental stress ratings for the MCH. Functional operation at or exceeding the absolute maximum and minimum ratings is neither implied nor guaranteed. Functional operating parameters are listed in the AC tables.

Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operating beyond the "operating conditions" is not recommended and extended exposure beyond "operating conditions" may affect reliability.

Table 6-1. Absolute Maximum Ratings

Symbol	Parameter	Min	Max	Unit
$T_{storage}$	Storage Temperature	-40.0	85.0	°C
V_{TT}	FSB Termination Supply Voltage input with respect to Vss (1.1 V - 1.2 V)	-0.30	1.85	V
Vcc	Variable MCH Supply Voltage with respect to Vss (~1.25 V)	-0.30	1.85	V
VCC...18ICC	MCH 1.8 V Supply Voltages	-0.50	2.4	V
VCC...33	MCH 3.3 V Supply Voltages	-0.50	3.9	V

6.2 Thermal Characteristics

For information on thermal characteristics, consult the *Intel® 5400 Chipset Series Memory Controller Hub (MCH) Thermal/Mechanical Design Guidelines*.

6.3 Mechanical Specifications

MCH mechanical specifications are listed in the *Intel® 5400 Chipset Series Memory Controller Hub (MCH) Thermal/Mechanical Design Guidelines.*

6.4 Power Characteristics

Table 6-2 lists the operating conditions for the MCH power supply rails for Vcc and loading conditions (Icc) when running at 1333 MHz Front Side Bus (FSB). For 1600 MHz FSB, see Table 6-3 for the loading conditions.

Table 6-2. Operating Condition Power Supply Rails (Sheet 1 of 2)

Power Pins	Symbol	Parameter	Min	Nom	Max	Unit	Notes
VCC	VCC (DC)	MCH Core Supply Voltage DC spec (Variable Supply)	1.2125 (Vnom - 3%)	1.25 Vnom	1.2875 (Vnom + 3%)	V	1,5
	VCC (AC)	VCC AC spec, up to 20 MHz (Variable Supply)	-25	0	+25	mV	2
	VCC (AC+DC)	VCC AC+DC Combined spec, up to 20MHz (Variable Supply)	1.1875 (Vnom - 5%)	1.25 Vnom	1.3125 (Vnom + 5%)	V	3,4,5
	ICC	MCH Core Supply Current		7.62		A	
VCCSF	VCCSF (DC)	MCH Snoop Filter Supply Voltage DC Spec	1.2125 (Vnom - 3%)	1.25 Vnom	1.2875 (Vnom + 3%)	V	1,5
	VCCSF (AC)	VCCSF AC spec, up to 20 MHz	-25	0	+25	mV	2
	VCCSF (AC+DC)	VCCSF AC+DC Combined spec, up to 20 MHz	1.1875 (Vnom - 5%)	1.25 Vnom	1.3125 (Vnom + 5%)	V	3,4,5
	ICCSF	MCH Snoop Filter Supply Current		3.08		A	
VCCFBD	VCCFBD (DC)	MCH FBD Supply Voltage DC spec	1.2125	1.25	1.287	V	1
	VCCFBD (AC)	VCCFBD AC spec, up to 20MHz	-25	0	+25	mV	2
	VCCFBD (AC+DC)	VCCFBD AC+DC Combined spec, up to 20 MHz	1.1875	1.25	1.3125	V	3,4
	ICCFBD	MCH FBD Supply Current		4.70		A	
VCCAPE125	VCCAPE125 (DC)	MCH PXP2 Analog Ring Power Supply DC spec	1.2125	1.25	1.2875	V	1
	VCCAPE125 (AC)	MCH PXP2 Analog Ring Power Supply AC spec up to 20 MHz	-25	0	+25	mV	2
	VCCAPE125 (AC+DC)	MCH PXP2 Analog Ring Power Supply AC + DC spec (up to 20 MHz	1.1875	1.25 V	1.3125	V	3,4
	ICCAPE125	MCH PXP2 Analog Current		7.59		A	
VCCA125PLL	ICC125PLL	PXP2 and FBD PLL Supply Current		430		mA	
VCCA18PLL	ICC18PLL	PXP2 and FBD PLL Supply Current		100		mA	
VCC125MISC	ICC125MISC	1.25 V Fixed Misc Supply Current		30		mA	
VCCRPE125	VCCRPE125 (DC)	MCH PXP2 Ring Power Supply DC spec (Variable Supply)	1.2125 (Vnom - 3%)	1.25 Vnom	1.2875 (Vnom + 3%)	V	1,5
	VCCRPE125 (AC)	MCH PXP2 Ring Power Supply AC spec up to 20 MHz (Variable Supply)	-25	0	+25	mV	2
	VCCRPE125 (AC+DC)	MCH PXP2 Ring Power Supply AC + DC spec up to 20 MHz (Variable Supply)	1.1875 (Vnom - 5%)	1.25 Vnom	1.3125 (Vnom + 5%)	V	3,4,5
	ICCRPE125	MCH PXP2 Ring Current		2.24		A	

Table 6-2. Operating Condition Power Supply Rails (Sheet 2 of 2)

Power Pins	Symbol	Parameter	Min	Nom	Max	Unit	Notes
VTT	VTT (DC)	Host AGTL+ Termination Voltage - DC spec	1.067	1.1	1.130	V	1
	VTT (DC)	Host AGTL+ Termination Voltage - DC spec	1.164	1.2	1.236	V	1
	VTT (AC)	V_{TT} AC spec, up to 20MHz	-25	0	+25	mV	2
	VTT (AC + DC)	V_{TT} AC +DC combined variations, up to 20MHz	1.045	1.25	1.26	V	3,4
	I_{TT}	Host AGTL+ Termination Current		2.50		A	
VCC..33	VCC33 (DC)	MCH 3.3 V Power Supply DC Spec	3.201	3.3	3.399	V	
	VCC33 (AC)	MCH 3.3 V Power Supply AC Spec	-66		66	mV	
	VCC33 (AC + DC)	MCH 3.3 V Power Supply DC Spec up to 20 MHz	3.135	1.25	3.465	V	
	IVCC33	MCH 3.3 V Current		0.01		A	

Notes:
1. Under no circumstances may the supply voltage go past the AC min/max window. The supply voltage may go outside the DC min/max window for transient events.
2. The supply voltage must stay within the DC min/max window in a static system (no active switching). The DC window only assumes voltage regulator ripple and motherboard induced noise.
3. MCH with 4 active FBD channels. Total Core + I/O current drawn off the 1.25 V rail.
4. The analog voltage is intended to be a filtered copy of its associated supply voltage. Refer to the Platform Design Guide for the recommended implementation and frequency response requirements of each filter.
5. Vnom range is 1.0625 V to 1.25 V. Vnom is determined by taking the value driven by the MCH on the VID bus and subtracting the voltage regulator offset of -15mv.

Table 6-3. 1600 MHZ FSB Operating Condition Power Supply Rails

Power Pins	Symbol	Parameter	Min	Nom	Max	Unit	Notes
VCC	ICC	MCH Core Supply Current		8.71	27.32	A	
VCCSF	ICCSF	MCH Snoop Filter Supply Current		3.63	9.0	A	
VCCFDBD	ICCFBD	MCH FBD Supply Current		4.95	7.21	A	
VCCAPE125	ICCAPE125	MCH PXP2 Analog Current		7.59	7.59	A	
VCCA125PLL	ICC125PLL	PXP2 and FBD PLL Supply Current		0.44	0.44	A	
VCCA18PLL	ICC18PLL	PXP2 and FBD PLL Supply Current		0.10	0.10	A	
VCC125MISC	ICC125MISC	1.25 V Fixed Misc Supply Current		300	200	mA	
VCCRPE125	ICCRPE125	MCH PXP2 Ring Current		2.24	3.40	A	
VTT	I_{TT}	Host AGTL+ Termination Current		3.00	6.20	A	
VCC...33	IVCC33	MCH 3.3 V Current		10	70	mA	

6.5 DC Characteristics

This section documents the DC characteristics of the MCH. The specifications are split into eight sections:

- Clocks
- Front Side Bus (FSB) Interface
- FB-DIMM (Fully Buffered DIMM) Memory Interface
- PCI Express 1.0/ ESI Interface
- PCI Express 2.0 Interface
- SMBus Interface
- JTAG Interface
- Miscellaneous Interface

6.5.1 Clock DC Characteristics

Table 6-4. Clock DC Characteristics (Sheet 1 of 2)

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
266/333/400 MHz FSB Clock (CORECLKN / CORECLKP)							
V_{IL}	(h)	Single-ended Input Low Voltage	-0.150	0	0.150	V	1
V_{IH}	(h)	Single-ended Input High Voltage	0.660	0.710	0.850	V	
$V_{CROSS(abs)}$	(h)	Absolute Crossing Point	0.250	0.350	0.550	V	2, 7
$V_{CROSS(rel)}$	(h)	Relative Crossing Point	0.250 + 0.5 x (V_{Havg} – 0.700)		0.550 + 0.5 x (V_{Havg} - 0.700)	V	7, 8
ΔV_{CROSS}	(h)	Vcross Variation			0.140	V	
V_{OS}	(h)	Single-ended maximum Overshoot voltage			1.15	V	3
V_{US}	(h)	Single-ended minimum Undershoot voltage	-0.300			V	4
V_{RBM}	(h)	Ringback Margin	0.200			V	5
V_{TR}	(h)	Threshold Region	V_{CROSS} – 0.100		V_{CROSS} + 0.100	V	6
100 MHz PCI Express Clock (PECLKN / PECLKP)							
V_{IL}	(q)	Input Low Voltage	-0.150	0		V	
V_{IH}	(q)	Input High Voltage	0.660	0.710	0.850	V	
$V_{CROSS(abs)}$	(q)	Absolute Crossing Point	0.250		0.550	V	2, 7
$V_{CROSS(rel)}$	(q)	Relative Crossing Point	0.250 + 0.5 x (V_{Havg} – 0.700)		0.550 + 0.5 x (V_{Havg} – 0.700)	V	7, 8
ΔV_{CROSS}	(q)	Range of Crossing Points			0.140	V	1, 2
V_{OS}	(q)	Overshoot			V_{IH} + 0.300	V	3
V_{US}	(q)	Undershoot	-0.300			V	4
V_{RBM}	(q)	Ringback Margin	0.200			V	5
V_{TR}	(q)	Threshold Region	V_{CROSS} – 0.100		V_{CROSS} + 0.100	V	6

Table 6-4. Clock DC Characteristics (Sheet 2 of 2)

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
133/167/200 MHz FB-DIMM Clock (FBDxxCLKN/ FBDxxCLKP)							
V_{IL}	(k)	Input Low Voltage	-0.150	0		V	
V_{IH}	(k)	Input High Voltage	0.660	0.710	0.850	V	
$V_{CROSS(abs)}$	(k)	Absolute Crossing Point	0.250		0.550	V	2, 7
$V_{CROSS(rel)}$	(k)	Relative Crossing Point	0.250 + 0.5 x (V_{Havg} – 0.700)		0.550 + 0.5 x (V_{Havg} – 0.700)	V	7, 8

Notes:
1. Crossing voltage is defined as the instantaneous voltage when the rising edge of CORECLKP is equal to the falling edge of CORECLKN.
2. Overshoot is defined as the absolute value of the maximum voltage.
3. Undershoot is defined as the absolute value of the minimum voltage.
4. Ringback Margin is defined as the absolute voltage difference between the maximum Rising Edge Ringback and the maximum Falling Edge Ringback. Both maximum Rising and Falling Ringbacks should not cross the threshold region.
5. Threshold Region is defined as a region centered around the crossing point voltage in which the differential receiver switches. It includes input threshold hysteresis.
6. The crossing point must meet the absolute and relative crossing point specifications simultaneously.
7. VHavg (the average of V_{IH}) can be measured directly using “Vtop” on Agilent scopes and “High” on Tektronix scopes.

6.5.2 FSB Interface DC Characteristics

Table 6-5. FSB Interface DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
V_{IL}	• (a) (b)	Host AGTL+ Input Low Voltage	0		GTLREF – (0.1 × V_{TT})	V	1, 2
V_{IH}	• (a) (b)	Host AGTL+ Input High Voltage	GTLREF + (0.1 × V_{TT})		V_{TT}	V	1, 3
V_{OL}	• (a) (c)	Host AGTL+ Output Low Voltage			0.4	V	
V_{OH}	• (a) (c)	Host AGTL+ Output High Voltage	0.90 x V_{TT}		V_{TT}	V	4
I_{OL}	• (a) (c)	Host AGTL+ Output Low Current			V_{TT} / (0.50 x R_{tt_min} + R_{on_min})	mA	8
I_{LI}	• (a) (b)	Host AGTL+ Input Leakage Current	n/a		± 200	uA	5, 6
I_{LO}	• (a) (b)	Host AGTL+ Output Leakage Current	n/a		± 200	uA	5, 6
R_{on}	•	Buffer on Resistance	7		11	Ω	
GTLREF	• (e)	Host Bus Reference Voltage	(0.98 x 0.67) x V_{TT}	0.67 x V_{TT}	(1.02 x 0.67) x V_{TT}	V	1
R_{TT}	•	Host Termination Resistance Common Clock, Async on Stripline	45	50	55	Ω	7

Notes:
1. GTLREF is equivalent to FSBxVREF. GTLREF is generated from V_{TT} on the baseboard by a voltage divider or 1% resistors.
2. V_{IL} is defined as the voltage range at a receiving agent that will be interpreted as an electrical low value.
3. V_{IH} is defined as the voltage range at a receiving agent that will be interpreted as an electrical high value.
4. V_{IH} and V_{OH} may experience excursions above V_{CC}. However, input signal drivers must comply with the signal quality specifications chapter in the document.
5. Leakage to VSS with land held at V_{TT}.
6. Leakage to V_{TT} with land held at 300 mV.
7. Use 50 ohm ± 15% for all Microstrip.

8. I_{OL} is defined as current when Output Low. The formula computes the total current drawn by the driver from VR (Voltage Regulator). Half of the total current goes through RTT on the chipset, and another half goes through the RTT on the CPU (the End-Bus-Agency).

6.5.3 FB-DIMM DC Characteristics

Table 6-6. FB-DIMM Transmitter (Tx) Output DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
V_{TX-CM_S}	(i) (j)	DC Common Mode Output Voltage for Small Voltage Swing	135		280	mV	1
V_{TX-CM_L}	(i) (j)	DC Common Mode Output Voltage for Large Voltage Swing			375	mV	1
V_{TX-SE}	(i) (j)	Single-ended Voltage	-75		750	mV	
RLTX-Diff	(i) (j)	Differential Return Loss	-8			dB	
RLTX-CM	(i) (j)	Common Mode Return Loss	-6			dB	
$Z_{TX-MATCH-DC}$	(i) (j)	D+/D- TX Impedance Difference			4%	Ω	2
$Z_{TX-COM-ESI-IMP-DC}$	(i) (j)	D+/D- TX Common Mode High Impedance State	5		20	kΩ	

Notes:
1. Defined as: $V_{TX-CM} = DC_{(avg)}$ of $| V_{TX-D+} + V_{TX-D-} | / 2$
2. TX DC impedance matching between D+ and D- on a given lane.

Table 6-7. FB-DIMM Receiver (Rx) Output DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
V_{RX-CM}	(i) (j)	DC Common Mode Input Voltage	120		400	mV	1
$Z_{RX-MATCH-DC}$	(i) (j)	D+/D- RX Impedance Difference			4%	Ω	2
RLRX-Diff	(i) (j)	Differential Return Loss	-10			dB	
RLRX-CM	(i) (j)	Common Mode Return Loss	-6			dB	

Notes:
1. DC (avg) of $| V_{RX-D+} + V_{RX-D-} | / 2$.
2. RX DC impedance matching between D+ and D- on a given lane.

6.5.4 PCI Express 1.0 * / ESI Interface DC Characteristics

Table 6-8. PCI Express / ESI Differential Transmitter (Tx) Output DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
VTX-CM-DC-ACTIVE-IDLE-DELTA	(o) (p)	Absolute Delta of DC Common Mode Voltage During L0 and Electrical Idle	0		100	mV	2
VTX-CM-DC-LINE-DELTA	(o) (p)	Absolute Delta of DC Common Mode Voltage between D+ and D-	0		25	mV	2
VTX-IDLE-DIFFp	(o) (p)	Electrical Idle Differential Peak Output Voltage			20	mV	2
VTX-RCV-DETECT	(o) (p)	The amount of voltage change allowed during Receiver Detection			600	mV	
VTX-DC-CM	(o) (p)	The TX DC Common Mode Voltage	0		3.6	V	2
ITX-SHORT	(o) (p)	The Short Circuit Current Limit			90	mA	
ZTX-DIFF-DC	(o) (p)	DC Differential TX Impedance	80	100	120	Ω	
ZTX-DC	(o) (p)	Transmitter DC Impedance	40			Ω	

Notes:
1. No test load is necessarily associated with this value.
2. Specified at the measurement point into a timing and voltage compliance test load and measured over any 250 consecutive TX UIs.

Table 6-9. PCI Express / ESI Differential Receiver (Rx) Input DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
ZRX-DIFF-DC	(o) (p)	DC Differential Input Impedance	80	100	120	Ω	5
ZRX-DC	(o) (p)	DC Input Impedance	40	50	60	Ω	2, 3
ZRX-High-Imp-DC	(o) (p)	Power Down DC Input Common Mode Impedance	200			kΩ	6
VRX-IDLE-DET-DIFFp	(o) (p)	Electrical Idle Detect Threshold	65		175	mV	

Notes:
1. No test load is necessarily associated with this value.
2. Specified at the measurement point and measured over any 250 consecutive UIs. If the clock to the RX and TX are not derived from the same reference clock, the TX UI recovered from 3500 consecutive UI must be used as a reference for the eye diagram.
3. A TRX-EYE=0.40UI provides for a total sum of 0.60 UI deterministic and random jitter budget for the Transmitter and interconnect collected any 250 consecutive UIs. The TRX-EYE-MEDIAN-to-MAX-JITTER specification ensures a jitter distribution in which the median and the maximum deviation from the median is less than half of the total .6 UI jitter budget collected over any 250 consecutive TX UIs. It should be noted that the median is not the same as the mean. The jitter median describes the point in time where the number of jitter points on either side is approximately equal as opposed to the averaged time value. If the clocks to the RX and TX are not derived from the same reference clock, the TX UI recovered from 3500 consecutive UI must be used as the reference for the eye diagram.
4. The Receiver input impedance shall result in a differential return loss greater than or equal to 15 dB with the D+ line biased to 300mV and the D- line biased to -300 mV and a common mode return loss greater than or equal to 6 dB (no bias required) over a frequency range of 50 MHz to 1.25 GHz. This input impedance requirement applies to all valid input levels. The reference impedance for return loss measurements for is 50 ohms to ground for both the D+ and D- line (i.e., as measured by a Vector Network Analyzer with 50 ohm probes). Note: that the series capacitors CTX is optional for the return loss measurement.
5. Impedance during all LTSSM states. When transitioning from a Fundamental Reset to Detect (the initial state of the LTSSM) there is a 5ms transition time before Receiver termination values must be met on all un-configured Lanes of a Port.
6. The RX DC Common Mode Impedance that exists when no power is present or Fundamental Reset is asserted. This helps ensure that the Receiver Detect circuit will not falsely assume a Receiver is powered on when it is not. This term must be measured at 300 mV above the RX ground.

6.5.5 PCI Express 2.0 Interface

Table 6-10. PCI Express 2.0 (Tx) Output DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
VTX-CM-DC-ACTIVE-IDLE-DELTA	(o) (p)	Absolute Delta of DC Common Mode Voltage During L0 and Electrical Idle	0		100	mV	2
VTX-CM-DC-LINE-DELTA	(o) (p)	Absolute Delta of DC Common Mode Voltage between D+ and D-	0		25	mV	2
VTX-IDLE-DIFF-DC	(o) (p)	DC Electrical Idle Differential Output Voltage			5	mV	2
VTX-RCV-DETECT	(o) (p)	The amount of voltage change allowed during Receiver Detection			600	mV	
VTX-DC-CM	(o) (p)	The TX DC Common Mode Voltage	0		3.6	V	2
ITX-SHORT	(o) (p)	The Short Circuit Current Limit			90	mA	
ZTX-DIFF-DC	(o) (p)	DC Differential TX Impedance	80		120	Ω	
ZTX-DC	(o) (p)	Transmitter DC Impedance	40			Ω	

Notes:
1. SSC permits a +0, - 5000 ppm modulation of the clock frequency at a modulation rate not to exceed 33 kHz.
2. Measurements at 5.0 GT/s require an oscilloscope with a bandwidth of .12.5 GHz, or equivalent, while measurements made at 2.5 GT/s require a scope with at least 6.2 GHz bandwidth. Measurement at 5.0 GT/s must deconvolve effects of compliance test board to yield an effective measurement at Tx pins. 2.5 GT/s may be measured within 200 mils of Tx device pins, although deconvolution is recommended.
3. Transmitter jitter is measured by driving the transmitter under test with a low jitter ìidealî clock and connecting the DUT to a reference load.
4. Transmitter raw jitter data must be convolved with a filtering function that represents the worst case CDR tracking BW. 2.5 GT/s and 5.0 GT/s use different filter functions. After the convolution process has been applied, the center of the resulting eye must be determined and used as a reference point for obtaining eye voltage and margins.
5. Common mode peak-to-peak voltage is defined by the expression: max{|(Vd+ + Vd-)|/2} ñ min{|(Vd+ + Vd-)|/2}. Measurement is made over at least 106 UI.
6. The Tx PLL Bandwidth must lie between the min and max ranges given in the above table. PLL peaking must lie below the value listed above. Note: the PLL B/W extends from zero up to the value(s) specified in the above table.

Table 6-11. PCI Express 2.0 (Rx) Output DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
ZRX-DIFF-DC	(o) (p)	DC Differential Input Impedance	80		120	Ω	5
ZRX-DC	(o) (p)	DC Input Impedance	40		60	Ω	2, 3
ZRX-High-Imp-DC_NEG	(o) (p)	Power Down DC Input Common Mode Impedance	1.0			kΩ	6
ZRX-High-Imp-DC_POS	(o) (p)	Power Down DC Input Common Mode Impedance	50			kΩ	6
VRX-IDLE-DET-DIFFp-p	(o) (p)	Electrical Idle Detect Threshold	65		175	mV	

Notes:
1. Receiver eye margins are defined into a 2 x 50 ohms reference load.
2. The four inherent timing error parameters are defined for the convenience of Rx designers, and they are measured during receiver tolerancing.
3. Two combinations of PLL BW and peaking are specified at 5.0 GT/s to permit designers to make tradeoffs between the two parameters. If the PLL's min B/W is greater or equal to 8 MHz, then up to 3.0 dB of peaking is permitted. If the PLL's min BW is relaxed to greater than or equal to 5.0 MHz, then a tighter peaking value of 1.0 dB must be met. Note: a PLL BW extends from zero up to the value(s) defined as the min or max in the above table. For 2.5 GT/s a single PLL bandwidth and peaking value of 1.5-22 MHz and 3.0 dB are defined.
4. Measurements must be made for both common mode and differential return loss. In both cases the DUT must be powered up and DC isolated, and its D+/D- inputs must be in the low-Z state.
5. The Rx DC Common Mode Impedance must be present when the receiver terminations are first enabled to ensure that the Receiver Detect occurs properly. Compensation of this impedance can start immediately and the Rx Common Mode Impedance (constrained by RLRX-CM to 50 ohms ±20%) must be within the specified range by the time Detect is entered.
6. Common mode peak voltage is defined by the expression: max{|(Vd+ - Vd-) - V-CMDC|}.

7. Separate ZRX-HIGH-IMP-DC parameters are defined for V<0 and V>0 to account for the input characteristics of some receivers. Transmitter designers need to comprehend the large difference between >0 and <0 Rx impedances when designing receiver detect circuits.

6.5.6 Miscellaneous DC Characteristics

Table 6-12. SMBus DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
V_{IH}	(w)	Input High Voltage	2.1			V	
V_{IL}	(w)	Input Low Voltage			0.8	V	
V_{OL}	(w)	Output Low Voltage			0.4	V	1
I_{OL}	(w)	Output Low Current			4	mA	
I_{Leak}	(w)	Leakage Current			10	µA	
C_{Pad}	(w)	Pad Capacitance			10	pF	

Notes:
1. At Vol max, Iol = max.

Table 6-13. JTAG DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
V_{IH}	(y)	Input High Voltage	0.9			V	
V_{IL}	(y)	Input Low Voltage			0.5	V	
V_{OL}	(z)	Output Low Voltage			0.4	V	
I_{Leak}	(y) (z)	Leakage Current			2.9	µA	

Table 6-14. 1.25 V CMOS DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
V_{IH}	(d) (cc)	Input High Voltage	0.9		1.25	V	
V_{IL}	(d) (cc)	Input Low Voltage	-0.2		0.5	V	
V_{OH}	(cc)	Output High Voltage	1.0			V	
V_{OL}	(cc)	Output Low Voltage			0.4	V	
I_{Leak}	(cc)	Leakage Current			70	µA	
V_{ABS}	(d) (cc)	Input Damage Thresholds	-0.2		1.35	V	

Table 6-15. 3.3 V CMOS DC Characteristics

Symbol	Signal Group	Parameter	Min	Nom	Max	Unit	Notes
V_{IH}	(dd)	Input High Voltage	2.1			V	
V_{IL}	(dd)	Input Low Voltage			0.8	V	
V_{OH}	(ee)	Output High Voltage				V	
V_{OL}	(ee)	Output Low Voltage			0.4	V	
I_{Leak}	(ee)	Leakage Current			10	µA	
V_{ABS}	(dd)	Input Damage Thresholds	-0.3	3	3.5	V	

§

7 Testability

7.1 JTAG Port

Each component in the Intel® 5400 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.

7.1.1 TAP Signals

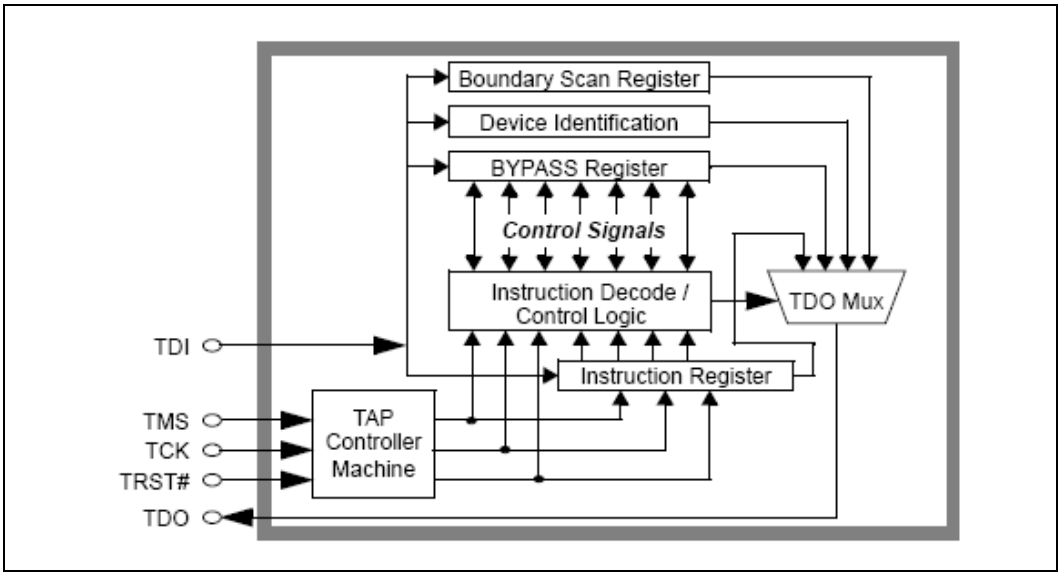
The TAP logic is accessed serially through 5 dedicated pins on each component as shown in Table 7-1.

Table 7-1. 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® 5400 chipset components is shown in Figure 7-1.

Figure 7-1. Simplified TAP Controller Block Diagram

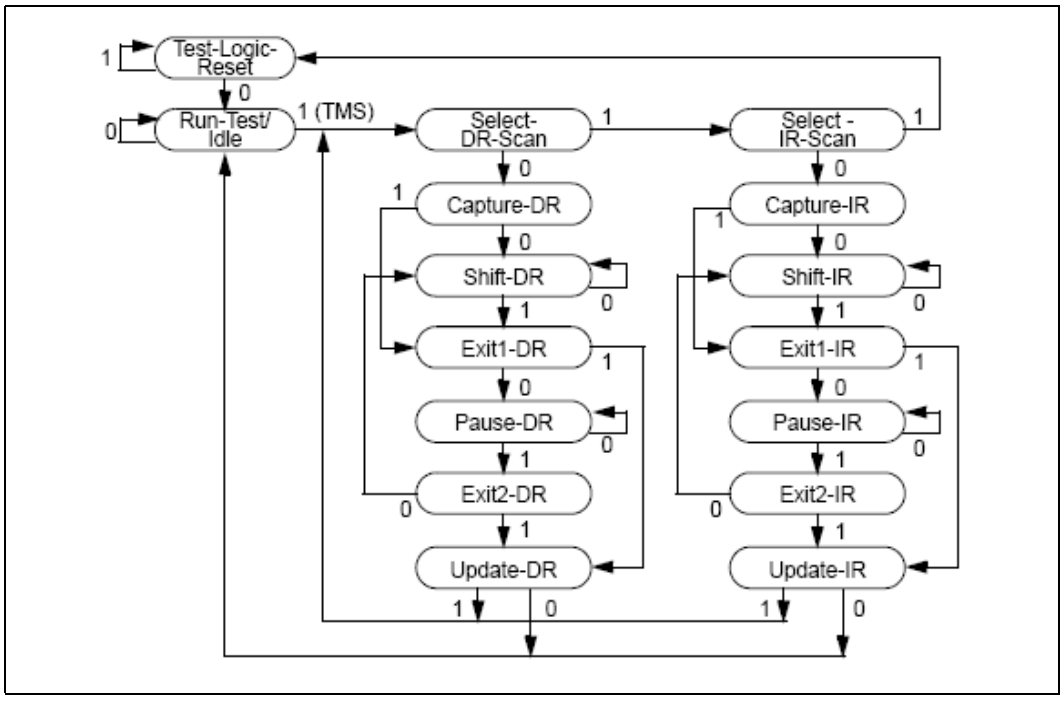


The TAP logic consists of a finite state machine controller, a serially-accessible instruction register, instruction decode logic and data registers. The set of data registers includes those described in the 1149.1 standard (the bypass register, device ID register, etc.), plus Intel® 5400 chipset specific additions.

7.1.2 Accessing the TAP Logic

The TAP is accessed through an 1149.1-compliant TAP controller finite state machine, which is illustrated in Figure 7-1. 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 7-2. 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.

7.1.3 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 7-2 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 7-2. 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.

7.1.4 Clocking the TAP

There is no minimum frequency at which the Intel® 5400 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.

7.1.5 Accessing the Instruction Register

Figure 7-3 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 7-3. TAP Instruction Register

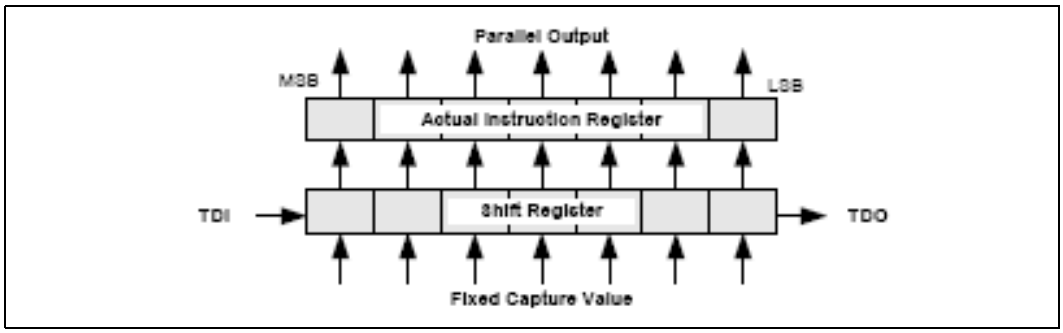


Figure 7-4 shows the operation of the instruction register during the Capture-IR, Shift-IR and Update-IR states. Shaded areas indicate the bits that are updated. In Capture-IR, the shift register portion of the instruction register is loaded in parallel with the fixed value "0000001". In Shift-IR, the 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 7-4. TAP Instruction Register Operation

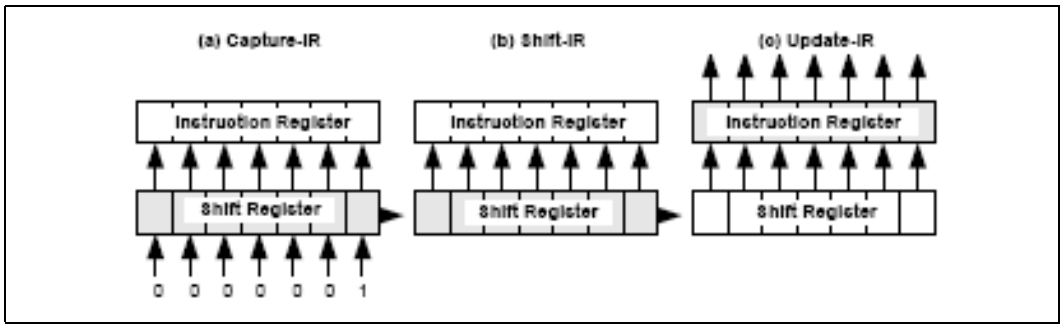
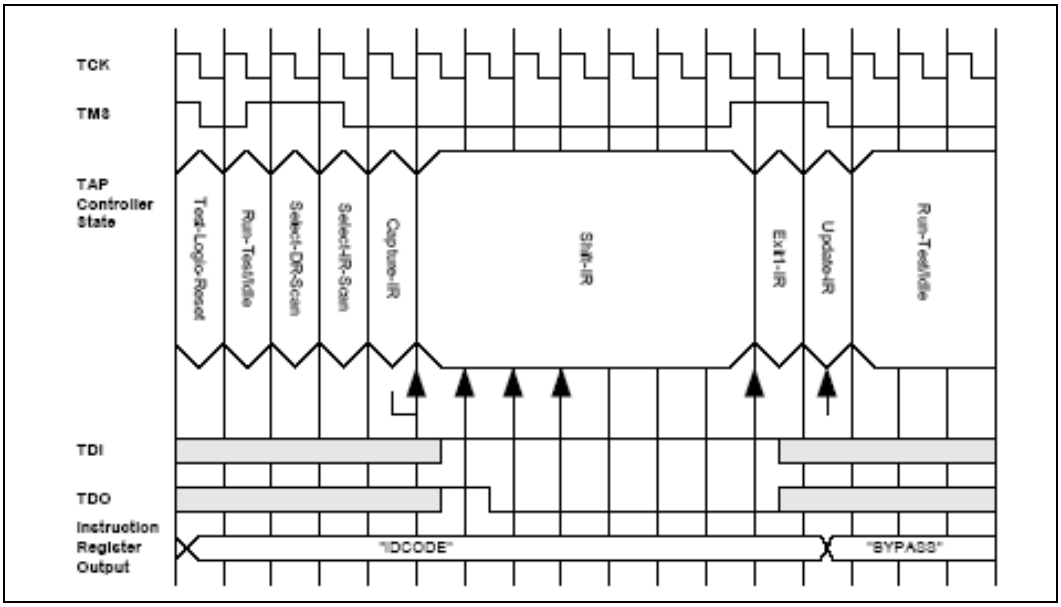


Figure 7-5 illustrates the timing when loading the BYPASS instruction (opcode 1111111b) into the TAP instruction register. Vertical arrows on the figure show the specific clock edges on which the Capture-IR, Shift-IR and Update-IR actions actually

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 7-5. TAP Instruction Register Access



7.1.6 Accessing the Data Registers

The test data registers in the Intel® 5400 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 7-2. 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.

7.1.7 Public TAP Instructions

Table 7-3 contains descriptions of the encoding and operation of the public TAP instructions. There are four 1149.1-defined instructions implemented in the Intel® 5400 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.

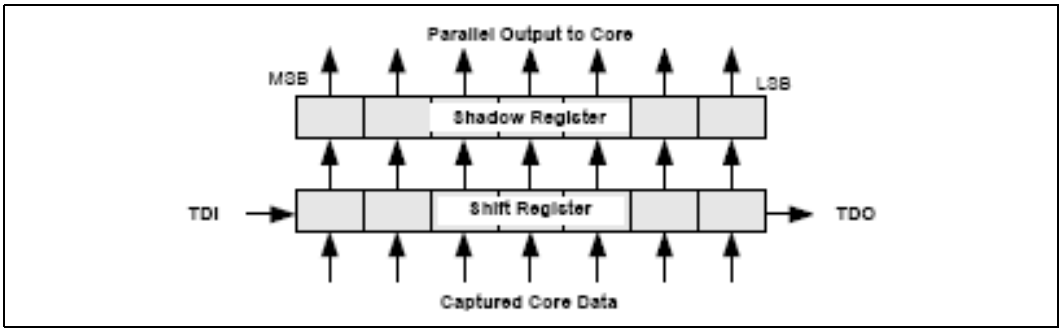
Table 7-3. Public TAP Instructions

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.

7.1.8 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 7-6 which is the same as the instruction register.

Figure 7-6. TAP Data Register



7.1.9 Public Data Register Control

Table 7-4 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 7-4. 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

7.1.10 Bypass Register

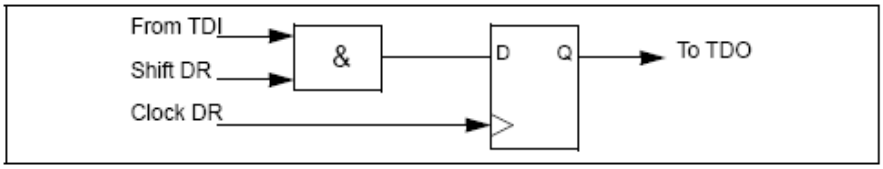
This register provides the minimal length path between TDI and TDO. It is loaded with a logical 0 during the Capture-DR state. The Bypass Register is a single bit register and is used to provide a minimum-length serial path through the device. This allows more rapid movement of test data to and from 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 7-7 for an implementation example.

7.1.10.1 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 7-7. Bypass Register Implementation



7.1.11 Device ID Register

This register contains the device identification code in the format shown in Table 7-5. 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 7-5. Intel 5400 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
Intel® 5400 chipset MCH – B0	0100	000100	01000	01010	00000001001	1	0x4110A013

7.1.12 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.

7.2 Extended Debug Port (XDP)

The Extended Debug Port is covered in the chipset XDP design guide.

§

8 Ballout and Package Information

8.1 Intel 5400 Chipset MCH Ballout

The following section presents preliminary ballout information for the MCH. This ballout is subject to change and is to be used for informational purposes only.

Figure 8-1. Intel 5400 Chipset MCH Quadrant Map update Figure

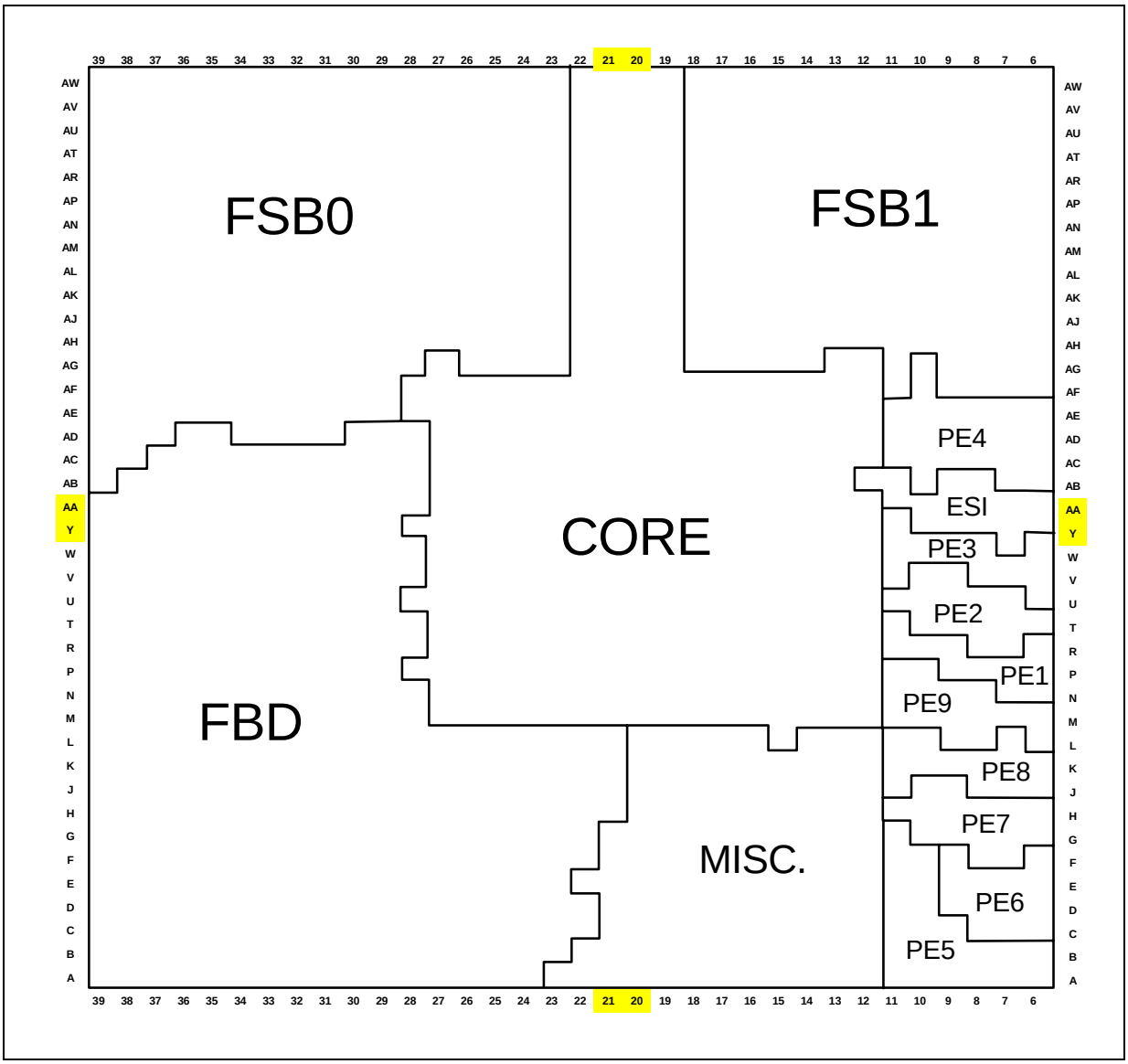
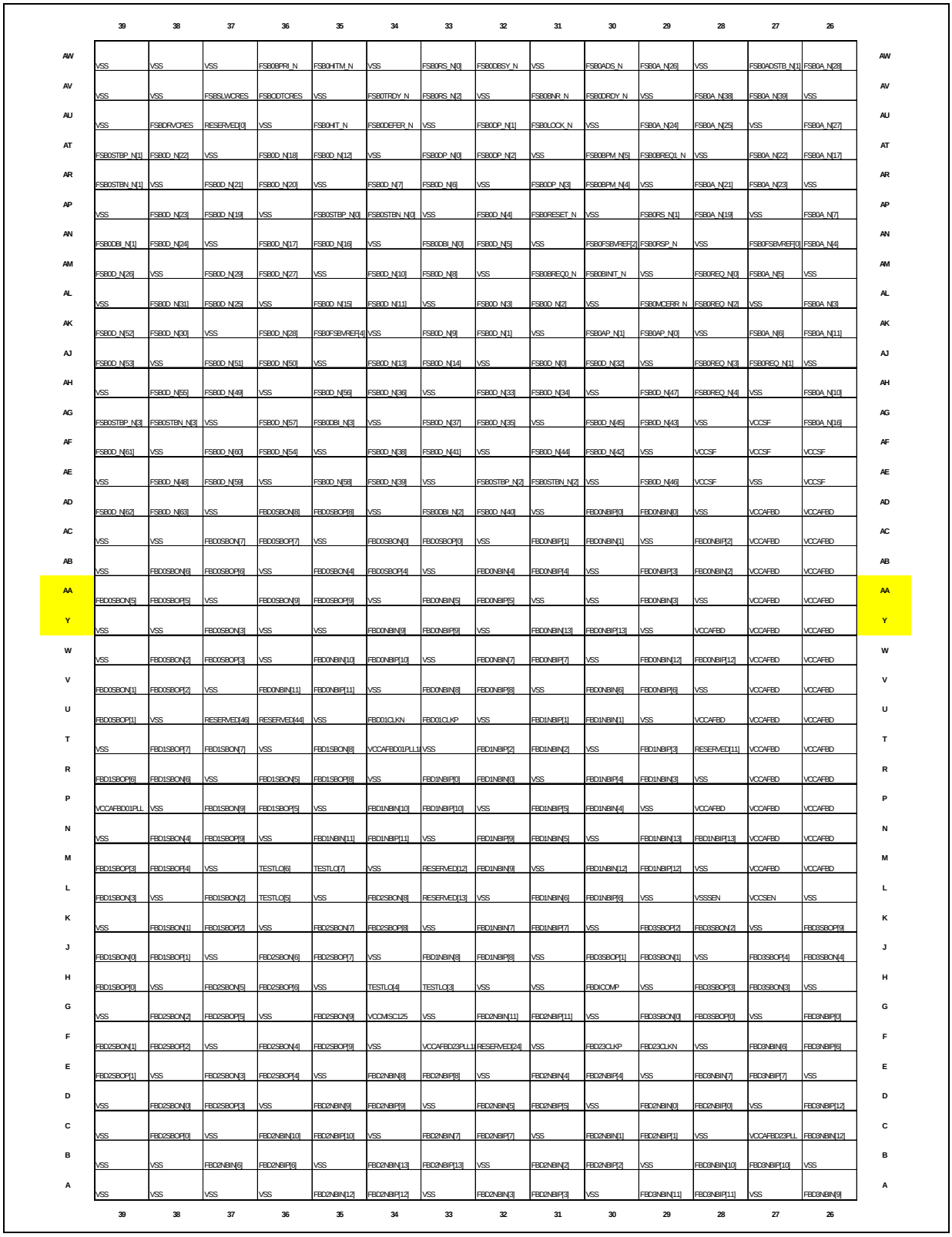


Figure 8-2. Intel 5400 Chipset MCH Ballout Left Side (Top View)

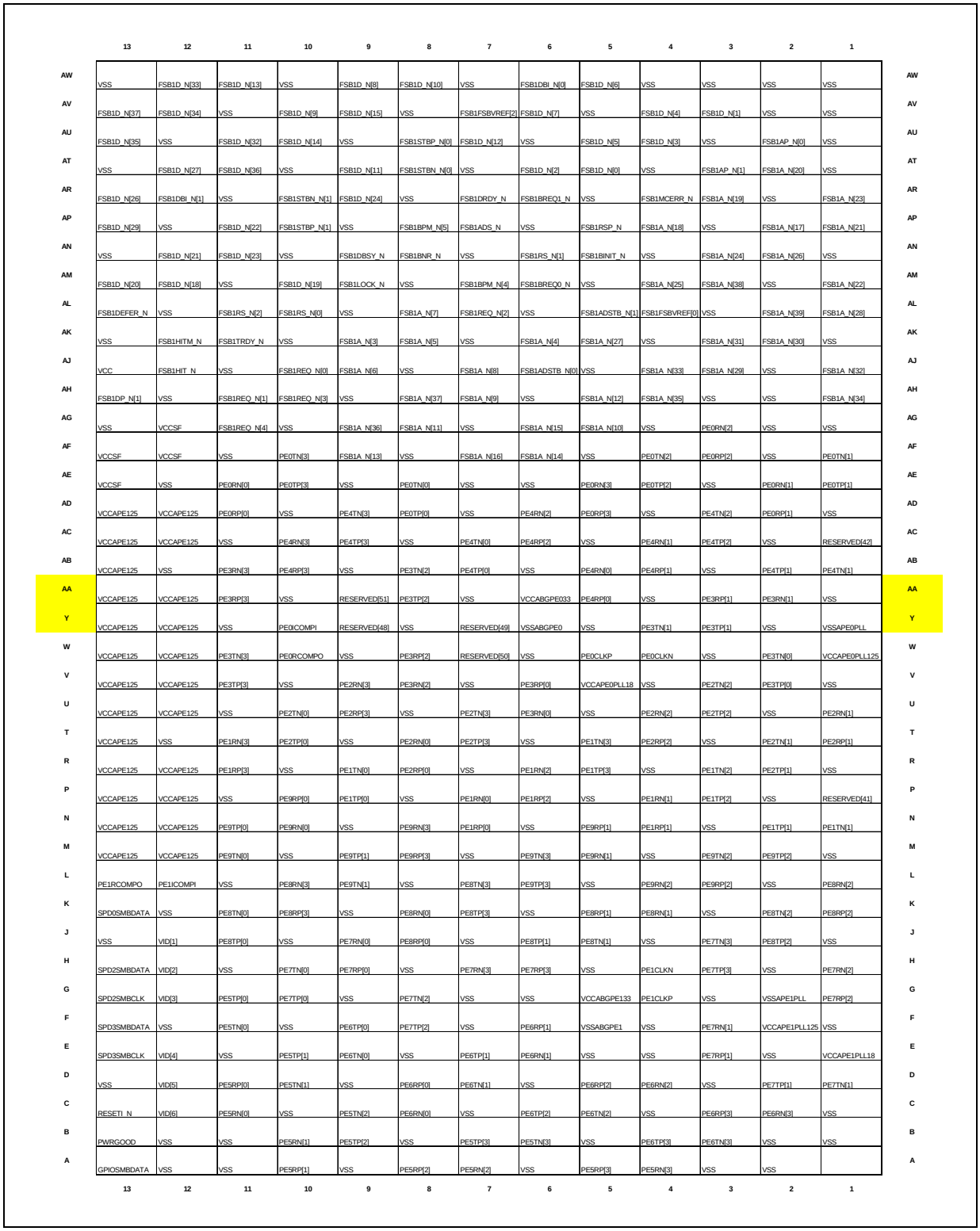


	39	38	37	36	35	34	33	32	31	30	29	28	27	26	
AW	VSS	VSS	VSS	FSB0BPRI_N	FSB0HITM_N	VSS	FSB0RS_N[0]	FSB0DBSY_N	VSS	FSB0ADS_N	FSB0A_N[26]	VSS	FSB0ADSTB_N[1]	FSB0A_N[28]	AW
AV	VSS	VSS	FSBSLWCRES	FSBODTCRES	VSS	FSB0TRDY_N	FSB0RS_N[2]	VSS	FSB0BNR_N	FSB0DRDY_N	VSS	FSB0A_N[38]	FSB0A_N[39]	VSS	AV
AU	VSS	FSB0RVCRES	RESERVED[0]	VSS	FSB0HIT_N	FSB0DEFER_N	VSS	FSB0DP_N[1]	FSB0LOCK_N	VSS	FSB0A_N[24]	FSB0A_N[25]	VSS	FSB0A_N[27]	AU
AT	FSB0STBP_N[1]	FSB0D_N[22]	VSS	FSB0D_N[18]	FSB0D_N[12]	VSS	FSB0DP_N[0]	FSB0DP_N[2]	VSS	FSB0BPM_N[5]	FSB0BREQ1_N	VSS	FSB0A_N[22]	FSB0A_N[17]	AT
AR	FSB0STBN_N[1]	VSS	FSB0D_N[21]	FSB0D_N[20]	VSS	FSB0D_N[7]	FSB0D_N[6]	VSS	FSB0DP_N[3]	FSB0BPM_N[4]	VSS	FSB0A_N[21]	FSB0A_N[23]	VSS	AR
AP	VSS	FSB0D_N[23]	FSB0D_N[19]	VSS	FSB0STBP_N[0]	FSB0STBN_N[0]	VSS	FSB0D_N[4]	FSB0RESET_N	VSS	FSB0RS_N[1]	FSB0A_N[19]	VSS	FSB0A_N[7]	AP
AN	FSB0DBI_N[1]	FSB0D_N[24]	VSS	FSB0D_N[17]	FSB0D_N[16]	VSS	FSB0DBI_N[0]	FSB0D_N[5]	VSS	FSB0FSBVREF[2]	FSB0RSP_N	VSS	FSB0FSBVREF[0]	FSB0A_N[4]	AN
AM	FSB0D_N[26]	VSS	FSB0D_N[29]	FSB0D_N[27]	VSS	FSB0D_N[10]	FSB0D_N[8]	VSS	FSB0BREQ0_N	FSB0BINIT_N	VSS	FSB0REQ_N[0]	FSB0A_N[5]	VSS	AM
AL	VSS	FSB0D_N[31]	FSB0D_N[25]	VSS	FSB0D_N[15]	FSB0D_N[11]	VSS	FSB0D_N[3]	FSB0D_N[2]	VSS	FSB0MCERR_N	FSB0REQ_N[2]	VSS	FSB0A_N[3]	AL
AK	FSB0D_N[52]	FSB0D_N[30]	VSS	FSB0D_N[28]	FSB0FSBVREF[4]	VSS	FSB0D_N[9]	FSB0D_N[1]	VSS	FSB0AP_N[1]	FSB0AP_N[0]	VSS	FSB0A_N[6]	FSB0A_N[11]	AK
AJ	FSB0D_N[53]	VSS	FSB0D_N[51]	FSB0D_N[50]	VSS	FSB0D_N[13]	FSB0D_N[14]	VSS	FSB0D_N[0]	FSB0D_N[32]	VSS	FSB0REQ_N[3]	FSB0REQ_N[1]	VSS	AJ
AH	VSS	FSB0D_N[55]	FSB0D_N[49]	VSS	FSB0D_N[56]	FSB0D_N[36]	VSS	FSB0D_N[33]	FSB0D_N[34]	VSS	FSB0D_N[47]	FSB0REQ_N[4]	VSS	FSB0A_N[10]	AH
AG	FSB0STBP_N[3]	FSB0STBN_N[3]	VSS	FSB0D_N[57]	FSB0DBI_N[3]	VSS	FSB0D_N[37]	FSB0D_N[35]	VSS	FSB0D_N[45]	FSB0D_N[43]	VSS	VCCSF	FSB0A_N[16]	AG
AF	FSB0D_N[61]	VSS	FSB0D_N[60]	FSB0D_N[54]	VSS	FSB0D_N[38]	FSB0D_N[41]	VSS	FSB0D_N[44]	FSB0D_N[42]	VSS	VCCSF	VCCSF	VCCSF	AF
AE	VSS	FSB0D_N[48]	FSB0D_N[59]	VSS	FSB0D_N[58]	FSB0D_N[39]	VSS	FSB0STBP_N[2]	FSB0STBN_N[2]	VSS	FSB0D_N[46]	VCCSF	VSS	VCCSF	AE
AD	FSB0D_N[62]	FSB0D_N[63]	VSS	FBD0SBON[8]	FBD0SBOP[8]	VSS	FSB0DBI_N[2]	FSB0D_N[40]	VSS	FBD0NBIP[0]	FBD0NBIN[0]	VSS	VCCAFBD	VCCAFBD	AD
AC	VSS	VSS	FBD0SBON[7]	FBD0SBOP[7]	VSS	FBD0SBON[0]	FBD0SBOP[0]	VSS	FBD0NBIP[1]	FBD0NBIN[1]	VSS	FBD0NBIP[2]	VCCAFBD	VCCAFBD	AC
AB	VSS	FBD0SBON[6]	FBD0SBOP[6]	VSS	FBD0SBON[4]	FBD0SBOP[4]	VSS	FBD0NBIN[4]	FBD0NBIP[4]	VSS	FBD0NBIP[3]	FBD0NBIN[2]	VCCAFBD	VCCAFBD	AB
AA	FBD0SBON[5]	FBD0SBOP[5]	VSS	FBD0SBON[9]	FBD0SBOP[9]	VSS	FBD0NBIN[5]	FBD0NBIP[5]	VSS	VSS	FBD0NBIN[3]	VSS	VCCAFBD	VCCAFBD	AA
Y	VSS	VSS	FBD0SBON[3]	VSS	VSS	FBD0NBIN[9]	FBD0NBIP[9]	VSS	FBD0NBIN[13]	FBD0NBIP[13]	VSS	VCCAFBD	VCCAFBD	VCCAFBD	Y
W	VSS	FBD0SBON[2]	FBD0SBOP[3]	VSS	FBD0NBIN[10]	FBD0NBIP[10]	VSS	FBD0NBIN[7]	FBD0NBIP[7]	VSS	FBD0NBIN[12]	FBD0NBIP[12]	VCCAFBD	VCCAFBD	W
V	FBD0SBON[1]	FBD0SBOP[2]	VSS	FBD0NBIN[11]	FBD0NBIP[11]	VSS	FBD0NBIN[8]	FBD0NBIP[8]	VSS	FBD0NBIN[6]	FBD0NBIP[6]	VSS	VCCAFBD	VCCAFBD	V
U	FBD0SBOP[1]	VSS	RESERVED[46]	RESERVED[44]	VSS	FBD01CLKN	FBD01CLKP	VSS	FBD1NBIP[1]	FBD1NBIN[1]	VSS	VCCAFBD	VCCAFBD	VCCAFBD	U
T	VSS	FBD1SBOP[7]	FBD1SBON[7]	VSS	FBD1SBON[8]	VCCAFBD01PLL1	VSS	FBD1NBIP[2]	FBD1NBIN[2]	VSS	FBD1NBIP[3]	RESERVED[11]	VCCAFBD	VCCAFBD	T
R	FBD1SBOP[6]	FBD1SBON[6]	VSS	FBD1SBON[5]	FBD1SBOP[8]	VSS	FBD1NBIP[0]	FBD1NBIN[0]	VSS	FBD1NBIP[4]	FBD1NBIN[3]	VSS	VCCAFBD	VCCAFBD	R
P	VCCAFBD01PLL	VSS	FBD1SBON[9]	FBD1SBOP[5]	VSS	FBD1NBIN[10]	FBD1NBIP[10]	VSS	FBD1NBIP[5]	FBD1NBIN[4]	VSS	VCCAFBD	VCCAFBD	VCCAFBD	P
N	VSS	FBD1SBON[4]	FBD1SBOP[9]	VSS	FBD1NBIN[11]	FBD1NBIP[11]	VSS	FBD1NBIP[9]	FBD1NBIN[5]	VSS	FBD1NBIN[13]	FBD1NBIP[13]	VCCAFBD	VCCAFBD	N
M	FBD1SBOP[3]	FBD1SBOP[4]	VSS	TESTLO[6]	TESTLO[7]	VSS	RESERVED[12]	FBD1NBIN[9]	VSS	FBD1NBIN[12]	FBD1NBIP[12]	VSS	VCCAFBD	VCCAFBD	M
L	FBD1SBON[3]	VSS	FBD1SBON[2]	TESTLO[5]	VSS	FBD2SBON[8]	RESERVED[13]	VSS	FBD1NBIN[6]	FBD1NBIP[6]	VSS	VSSSEN	VCCSEN	VSS	L
K	VSS	FBD1SBON[1]	FBD1SBOP[2]	VSS	FBD2SBON[7]	FBD2SBOP[8]	VSS	FBD1NBIN[7]	FBD1NBIP[7]	VSS	FBD3SBOP[2]	FBD3SBON[2]	VSS	FBD3SBOP[9]	K
J	FBD1SBON[0]	FBD1SBOP[1]	VSS	FBD2SBON[6]	FBD2SBOP[7]	VSS	FBD1NBIN[8]	FBD1NBIP[8]	VSS	FBD3SBOP[1]	FBD3SBON[1]	VSS	FBD3SBOP[4]	FBD3SBON[4]	J
H	FBD1SBOP[0]	VSS	FBD2SBON[5]	FBD2SBOP[6]	VSS	TESTLO[4]	TESTLO[3]	VSS	VSS	RBDICOMP	VSS	FBD3SBOP[3]	FBD3SBON[3]	VSS	H
G	VSS	FBD2SBON[7]	FBD2SBOP[5]	VSS	FBD2SBON[9]	VCCMISC125	VSS	FBD2NBIN[11]	FBD2NBIP[11]	VSS	FBD3SBON[0]	FBD3SBOP[0]	VSS	FBD3NBIP[0]	G
F	FBD2SBON[1]	FBD2SBOP[2]	VSS	FBD2SBON[4]	FBD2SBOP[9]	VSS	VCCAFBD23PLL1	RESERVED[24]	VSS	FBD23CLKP	FBD23CLKN	VSS	FBD3NBIN[6]	FBD3NBIP[6]	F
E	FBD2SBOP[1]	VSS	FBD2SBON[3]	FBD2SBOP[4]	VSS	FBD2NBIN[8]	FBD2NBIP[8]	VSS	FBD2NBIN[4]	FBD2NBIP[4]	VSS	FBD3NBIN[7]	FBD3NBIP[7]	VSS	E
D	VSS	FBD2SBON[0]	FBD2SBOP[3]	VSS	FBD2NBIN[9]	FBD2NBIP[9]	VSS	FBD2NBIN[5]	FBD2NBIP[5]	VSS	FBD2NBIN[0]	FBD2NBIP[0]	VSS	FBD3NBIP[12]	D
C	VSS	FBD2SBOP[0]	VSS	FBD2NBIN[10]	FBD2NBIP[10]	VSS	FBD2NBIN[7]	FBD2NBIP[7]	VSS	FBD2NBIN[1]	FBD2NBIP[1]	VSS	VCCAFBD23PLL	FBD3NBIN[12]	C
B	VSS	VSS	FBD2NBIN[6]	FBD2NBIP[6]	VSS	FBD2NBIN[13]	FBD2NBIP[13]	VSS	FBD2NBIN[2]	FBD2NBIP[2]	VSS	FBD3NBIN[10]	FBD3NBIP[10]	VSS	B
A	VSS	VSS	VSS	VSS	FBD2NBIN[12]	FBD2NBIP[12]	VSS	FBD2NBIN[3]	FBD2NBIP[3]	VSS	FBD3NBIN[11]	FBD3NBIP[11]	VSS	FBD3NBIN[9]	A
	39	38	37	36	35	34	33	32	31	30	29	28	27	26	

Figure 8-3. Intel 5400 Chipset MCH Ballout Center (Top View)

	26	25	24	23	22	21	20	19	18	17	16	15	14	
AW	FSB0A_N[28]	VSS	FSB0A_N[29]	FSB0A_N[35]	VTT	VTT	VTT	VTT	FSB1D_N[45]	FSB1D_N[44]	VSS	FSB1STBN_N[2]	FSB1D_N[39]	AW
AV	VSS	FSB0A_N[30]	FSB0A_N[33]	VSS	VTT	VTT	VTT	VTT	FSB1D_N[46]	VSS	FSB1FSBVREF[4]	FSB1STBP_N[2]	VSS	AV
AU	FSB0A_N[27]	FSB0A_N[31]	VSS	VCCACPLL	VTT	VTT	VTT	VTT	VSS	FSB1D_N[42]	FSB1D_N[41]	VSS	FSB1DBI_N[2]	AU
AT	FSB0A_N[17]	VSS	FSB0A_N[32]	VCCDCPLL	VTT	VTT	VTT	VTT	FSB1D_N[55]	FSB1D_N[53]	VSS	FSB1D_N[40]	FSB1D_N[38]	AT
AR	VSS	FSB0A_N[20]	FSB0A_N[34]	VSS	VTT	VTT	VTT	VTT	FSB1D_N[49]	VSS	FSB1D_N[47]	FSB1D_N[43]	VSS	AR
AP	FSB0A_N[7]	FSB0A_N[18]	VSS	CORECLKN	VTT	VTT	VTT	VTT	VSS	FSB1D_N[51]	FSB1D_N[50]	VSS	FSB1D_N[31]	AP
AN	FSB0A_N[4]	VSS	FSB0A_N[8]	CORECLKP	VTT	VTT	VTT	VTT	FSB1STBN_N[3]	FSB1D_N[52]	VSS	FSB1D_N[30]	FSB1D_N[25]	AN
AM	VSS	FSB0A_N[36]	FSB0ADSTB_N[0]	VSS	VTT	VTT	VTT	VTT	FSB1STBP_N[3]	VSS	FSB1D_N[56]	FSB1D_N[28]	VSS	AM
AL	FSB0A_N[3]	FSB0A_N[37]	VSS	RESERVED[1]	VTT	VTT	VTT	VTT	VSS	FSB1DBI_N[3]	FSB1D_N[57]	VSS	FSB1D_N[17]	AL
AK	FSB0A_N[11]	VSS	FSB0A_N[9]	VSS	VTT	VTT	VTT	VTT	FSB1D_N[60]	FSB1D_N[61]	VSS	FSB1BPRI_N	FSB1D_N[16]	AK
AJ	VSS	FSB0A_N[12]	FSB0A_N[13]	VSS	VTT	VTT	VTT	VTT	FSB1D_N[48]	VSS	FSB1D_N[54]	FSB1DP_N[0]	VSS	AJ
AH	FSB0A_N[10]	VCC	VSS	VCC	VTT	VTT	VTT	VTT	VSS	FSB1D_N[63]	FSB1D_N[59]	VSS	FSB1DP_N[2]	AH
AG	FSB0A_N[16]	VSS	FSB0A_N[14]	FSB0A_N[15]	VTT	VTT	VTT	VTT	FSB1D_N[58]	FSB1D_N[62]	VSS	FSB1RESET_N	FSB1DP_N[3]	AG
AF	VCCSF	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VCCSF	AF
AE	VCCSF	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VTT	VCCSF	AE
AD	VCCAFBD	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VCCRPE125	VCCRPE125	AD
AC	VCCAFBD	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCCRPE125	VCCRPE125	AC
AB	VCCAFBD	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCCAPE125	AB
AA	VCCAFBD	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCCRPE125	VCCAPE125	AA
Y	VCCAFBD	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VCCRPE125	VCCAPE125	Y
W	VCCAFBD	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCCRPE125	VCCAPE125	W
V	VCCAFBD	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCCAPE125	V
U	VCCAFBD	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCCRPE125	VCCAPE125	U
T	VCCAFBD	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VCCRPE125	VCCAPE125	T
R	VCCAFBD	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCC	VSS	VCCRPE125	VCCAPE125	R
P	VCCAFBD	VCCAFBD	VCCAFBD	VCCAFBD	VCCAFBD	VSS	VCC	VSS	VCC	VSS	VCCRPE125	VSS	VCCAPE125	P
N	VCCAFBD	VCCAFBD	VCCAFBD	VCCAFBD	VCCAFBD	VCCAFBD	VCCAFBD	VCC	VSS	VCC	VCCRPE125	VCCRPE125	VCCAPE125	N
M	VCCAFBD	VCCAFBD	VCCAFBD	VCCAFBD	VCCAFBD	VCCAFBD	VCCAFBD	VSS	VCC	VCC	VSS	VCCRPE125	VSS	M
L	VSS	FBD3SBOP[5]	FBD3SBON[5]	VSS	FBD3SBOP[8]	FBD3SBON[8]	VSS	XDPDATA_N[12]	XDPRDYREQ_N	VSS	VSS	VCCRPE125	THERMTRIP_N	L
K	FBD3SBOP[9]	FBD3SBON[9]	VSS	FBD3SBOP[7]	FBD3SBON[7]	VSS	PSEL[0]	VSS	XDPCLK1X	XDPDATA_N[1]	VSS	VSS	SPD0SMBCLK	K
J	FBD3SBON[4]	VSS	FBD3SBOP[6]	FBD3SBON[6]	VSS	VSS	VSS	VSCALEN	VCCMISC125	VSS	VSS	VCCMISC33	PLLPWRDET	J
H	VSS	RESERVED[45]	RESERVED[47]	VSS	FBD3NBIP[1]	VSS	VSS	VSS	XDPDATA_N[14]	XDPDATA_N[2]	VCCMISC33	SPD1SMBDATA	VCCMISC33	H
G	FBD3NBIP[0]	FBD3NBIN[0]	VSS	FBD3NBIP[2]	FBD3NBIN[1]	TESTLO[10]	PSEL[1]	VCCMISC125	VSS	XDPVREF	VSS	SPD1SMBCLK	VSS	G
F	FBD3NBIP[6]	VSS	FBD3NBIP[3]	FBD3NBIN[2]	VSS	VSS	PSEL[2]	TESTLO[9]	XDPDATA_N[5]	XDPDATA_N[0]	XDPDATA_N[4]	ERR2_N	VCCMISC33	F
E	VSS	FBD3NBIP[13]	FBD3NBIN[3]	VSS	TESTLO[11]	RESERVED[27]	VCCMISC125	VSS	XDPDATA_N[8]	VSS	XDPDATA_N[3]	VSS	ERR1_N	E
D	FBD3NBIP[12]	FBD3NBIN[13]	VSS	FBD3NBIN[4]	FBD3NBIP[4]	FSBSLWCTRL	TCK	TESTLO[8]	VSS	XDPDATA_N[7]	XDPDATA_N[6]	THERMALERT_N	ERR0_N	D
C	FBD3NBIN[12]	VSS	FBD3NBIN[5]	FBD3NBIP[5]	VSS	XOROUT	VSS	TDI	XDPDATA_N[11]	XDPDSTBN_N	VSS	SPARE_IO	CFGSMBDATA	C
B	VSS	FBD3NBIN[8]	FBD3NBIP[8]	VSS	VSS	VSS	TDO	XDPRDYACK_N	VSS	XDPDSTBP_N	XDPDATA_N[9]	XDPDATA_N[13]	VSS	B
A	FBD3NBIN[9]	FBD3NBIP[9]	VSS	TESTLO[1]	TESTLO[2]	VSS	TRSTNN	TMS	XDPDATA_N[15]	VSS	XDPDATA_N[10]	CFGSMBCLK	GPIOSMBCLK	A
	26	25	24	23	22	21	20	19	18	17	16	15	14	

Figure 8-4. Intel 5400 Chipset MCH Ballout Right Side (Top View)



	13	12	11	10	9	8	7	6	5	4	3	2	1	
AW	VSS	FSB1D_N[33]	FSB1D_N[13]	VSS	FSB1D_N[8]	FSB1D_N[10]	VSS	FSB1DBI_N[0]	FSB1D_N[6]	VSS	VSS	VSS	VSS	AW
AV	FSB1D_N[37]	FSB1D_N[34]	VSS	FSB1D_N[9]	FSB1D_N[15]	VSS	FSB1FSBVREF[2]	FSB1D_N[7]	VSS	FSB1D_N[4]	FSB1D_N[1]	VSS	VSS	AV
AU	FSB1D_N[35]	VSS	FSB1D_N[32]	FSB1D_N[14]	VSS	FSB1STBP_N[0]	FSB1D_N[12]	VSS	FSB1D_N[5]	FSB1D_N[3]	VSS	FSB1AP_N[0]	VSS	AU
AT	VSS	FSB1D_N[27]	FSB1D_N[36]	VSS	FSB1D_N[11]	FSB1STBN_N[0]	VSS	FSB1D_N[2]	FSB1D_N[0]	VSS	FSB1AP_N[1]	FSB1A_N[20]	VSS	AT
AR	FSB1D_N[26]	FSB1DBI_N[1]	VSS	FSB1STBN_N[1]	FSB1D_N[24]	VSS	FSB1DRDY_N	FSB1BREQ1_N	VSS	FSB1MCERR_N	FSB1A_N[19]	VSS	FSB1A_N[23]	AR
AP	FSB1D_N[29]	VSS	FSB1D_N[22]	FSB1STBP_N[1]	VSS	FSB1BPM_N[5]	FSB1ADS_N	VSS	FSB1RSP_N	FSB1A_N[18]	VSS	FSB1A_N[17]	FSB1A_N[21]	AP
AN	VSS	FSB1D_N[21]	FSB1D_N[23]	VSS	FSB1DBSY_N	FSB1BNR_N	VSS	FSB1RS_N[1]	FSB1BINIT_N	VSS	FSB1A_N[24]	FSB1A_N[26]	VSS	AN
AM	FSB1D_N[20]	FSB1D_N[18]	VSS	FSB1D_N[19]	FSB1LOCK_N	VSS	FSB1BPM_N[4]	FSB1BREQ0_N	VSS	FSB1A_N[25]	FSB1A_N[38]	VSS	FSB1A_N[22]	AM
AL	FSB1DEFER_N	VSS	FSB1RS_N[2]	FSB1RS_N[0]	VSS	FSB1A_N[7]	FSB1REQ_N[2]	VSS	FSB1ADSTB_N[1]	FSB1FSBVREF[0]	VSS	FSB1A_N[39]	FSB1A_N[28]	AL
AK	VSS	FSB1HITM_N	FSB1TRDY_N	VSS	FSB1A_N[3]	FSB1A_N[5]	VSS	FSB1A_N[4]	FSB1A_N[27]	VSS	FSB1A_N[31]	FSB1A_N[30]	VSS	AK
AJ	VCC	FSB1HIT_N	VSS	FSB1REQ_N[0]	FSB1A_N[6]	VSS	FSB1A_N[8]	FSB1ADSTB_N[0]	VSS	FSB1A_N[33]	FSB1A_N[29]	VSS	FSB1A_N[32]	AJ
AH	FSB1DP_N[1]	VSS	FSB1REQ_N[1]	FSB1REQ_N[3]	VSS	FSB1A_N[37]	FSB1A_N[9]	VSS	FSB1A_N[12]	FSB1A_N[35]	VSS	VSS	FSB1A_N[34]	AH
AG	VSS	VCCSF	FSB1REQ_N[4]	VSS	FSB1A_N[36]	FSB1A_N[11]	VSS	FSB1A_N[15]	FSB1A_N[10]	VSS	PE0RN[2]	VSS	VSS	AG
AF	VCCSF	VCCSF	VSS	PE0TN[3]	FSB1A_N[13]	VSS	FSB1A_N[16]	FSB1A_N[14]	VSS	PE0TN[2]	PE0RP[2]	VSS	PE0TN[1]	AF
AE	VCCSF	VSS	PE0RN[0]	PE0TP[3]	VSS	PE0TN[0]	VSS	VSS	PE0RN[3]	PE0TP[2]	VSS	PE0RN[1]	PE0TP[1]	AE
AD	VCCAPE125	VCCAPE125	PE0RP[0]	VSS	PE4TN[3]	PE0TP[0]	VSS	PE4RN[2]	PE0RP[3]	VSS	PE4TN[2]	PE0RP[1]	VSS	AD
AC	VCCAPE125	VCCAPE125	VSS	PE4RN[3]	PE4TP[3]	VSS	PE4TN[0]	PE4RP[2]	VSS	PE4RN[1]	PE4TP[2]	VSS	RESERVED[42]	AC
AB	VCCAPE125	VSS	PE3RN[3]	PE4RP[3]	VSS	PE3TN[2]	PE4TP[0]	VSS	PE4RN[0]	PE4RP[1]	VSS	PE4TP[1]	PE4TN[1]	AB
AA	VCCAPE125	VCCAPE125	PE3RP[3]	VSS	RESERVED[51]	PE3TP[2]	VSS	VCCABGPE033	PE4RP[0]	VSS	PE3RP[1]	PE3RN[1]	VSS	AA
Y	VCCAPE125	VCCAPE125	VSS	PE0ICOMPI	RESERVED[48]	VSS	RESERVED[49]	VSSABGPE0	VSS	PE3TN[1]	PE3TP[1]	VSS	VSSAPE0PLL	Y
W	VCCAPE125	VCCAPE125	PE3TN[3]	PE0RCOMPO	VSS	PE3RP[2]	RESERVED[50]	VSS	PE0CLKP	PE0CLKN	VSS	PE3TN[0]	VCCAPE0PLL125	W
V	VCCAPE125	VCCAPE125	PE3TP[3]	VSS	PE2RN[3]	PE3RN[2]	VSS	PE3RP[0]	VCCAPE0PLL18	VSS	PE2TN[2]	PE3TP[0]	VSS	V
U	VCCAPE125	VCCAPE125	VSS	PE2TN[0]	PE2RP[3]	VSS	PE2TN[3]	PE3RN[0]	VSS	PE2RN[2]	PE2TP[2]	VSS	PE2RN[1]	U
T	VCCAPE125	VSS	PE1RN[3]	PE2TP[0]	VSS	PE2RN[0]	PE2TP[3]	VSS	PE1TN[3]	PE2RP[2]	VSS	PE2TN[1]	PE2RP[1]	T
R	VCCAPE125	VCCAPE125	PE1RP[3]	VSS	PE1TN[0]	PE2RP[0]	VSS	PE1RN[2]	PE1TP[3]	VSS	PE1TN[2]	PE2TP[1]	VSS	R
P	VCCAPE125	VCCAPE125	VSS	PE9RP[0]	PE1TP[0]	VSS	PE1RN[0]	PE1RP[2]	VSS	PE1RN[1]	PE1TP[2]	VSS	RESERVED[41]	P
N	VCCAPE125	VCCAPE125	PE9TP[0]	PE9RN[0]	VSS	PE9RN[3]	PE1RP[0]	VSS	PE9RP[1]	PE1RP[1]	VSS	PE1TP[1]	PE1TN[1]	N
M	VCCAPE125	VCCAPE125	PE9TN[0]	VSS	PE9TP[1]	PE9RP[3]	VSS	PE9TN[3]	PE9RN[1]	VSS	PE9TN[2]	PE9TP[2]	VSS	M
L	PE1RCOMPO	PE1ICOMPI	VSS	PE8RN[3]	PE9TN[1]	VSS	PE8TN[3]	PE9TP[3]	VSS	PE9RN[2]	PE9RP[2]	VSS	PE8RN[2]	L
K	SPD0SMBDATA	VSS	PE8TN[0]	PE8RP[3]	VSS	PE8RN[0]	PE8TP[3]	VSS	PE8RP[1]	PE8RN[1]	VSS	PE8TN[2]	PE8RP[2]	K
J	VSS	VID[1]	PE8TP[0]	VSS	PE7RN[0]	PE8RP[0]	VSS	PE8TP[1]	PE8TN[1]	VSS	PE7TN[3]	PE8TP[2]	VSS	J
H	SPD2SMBDATA	VID[2]	VSS	PE7TN[0]	PE7RP[0]	VSS	PE7RN[3]	PE7RP[3]	VSS	PE1CLKN	PE7TP[3]	VSS	PE7RN[2]	H
G	SPD2SMBCLK	VID[3]	PE5TP[0]	PE7TP[0]	VSS	PE7TN[2]	VSS	VSS	VCCABGPE133	PE1CLKP	VSS	VSSAPE1PLL	PE7RP[2]	G
F	SPD3SMBDATA	VSS	PE5TN[0]	VSS	PE6TP[0]	PE7TP[2]	VSS	PE6RP[1]	VSSABGPE1	VSS	PE7RN[1]	VCCAPE1PLL125	VSS	F
E	SPD3SMBCLK	VID[4]	VSS	PE5TP[1]	PE6TN[0]	VSS	PE6TP[1]	PE6RN[1]	VSS	VSS	PE7RP[1]	VSS	VCCAPE1PLL18	E
D	VSS	VID[5]	PE5RP[0]	PE5TN[1]	VSS	PE6RP[0]	PE6TN[1]	VSS	PE6RP[2]	PE6RN[2]	VSS	PE7TP[1]	PE7TN[1]	D
C	RESETI_N	VID[6]	PE5RN[0]	VSS	PE5TN[2]	PE6RN[0]	VSS	PE6TP[2]	PE6TN[2]	VSS	PE6RP[3]	PE6RN[3]	VSS	C
B	PWRGOOD	VSS	VSS	PE5RN[1]	PE5TP[2]	VSS	PE5TP[3]	PE5TN[3]	VSS	PE6TP[3]	PE6TN[3]	VSS	VSS	B
A	GPIOSMBDATA	VSS	VSS	PE5RP[1]	VSS	PE5RP[2]	PE5RN[2]	VSS	PE5RP[3]	PE5RN[3]	VSS	VSS		A
	13	12	11	10	9	8	7	6	5	4	3	2	1	

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 1 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
A2	VSS			B1	VSS		
A3	VSS			B2	VSS		
A4	PE5RN[3]			B3	PE6TN[3]		
A5	PE5RP[3]			B4	PE6TP[3]		
A6	VSS			B5	VSS		
A7	PE5RN[2]			B6	PE5TN[3]		
A8	PE5RP[2]			B7	PE5TP[3]		
A9	VSS			B8	VSS		
A10	PE5RP[1]			B9	PE5TP[2]		
A11	VSS			B10	PE5RN[1]		
A12	VSS			B11	VSS		
A13	GPIOSMBDATA			B12	VSS		
A14	GPIOSMBCLK			B13	PWRGOOD		
A15	CFGSMBCLK			B14	VSS		
A16	XDPDATA_N[10]			B15	XDPDATA_N[13]		
A17	VSS			B16	XDPDATA_N[9]		
A18	XDPDATA_N[15]			B17	XDPDSTBP_N		
A19	TMS			B18	VSS		
A20	TRSTNN			B19	XDPRDYACK_N		
A21	VSS			B20	TDO		
A22	TESTLO[2]			B21	VSS		
A23	TESTLO[1]			B22	VSS		
A24	VSS			B23	VSS		
A25	FBD3NBIP[9]			B24	FBD3NBIP[8]		
A26	FBD3NBIN[9]			B25	FBD3NBIN[8]		
A27	VSS			B26	VSS		
A28	FBD3NBIP[11]			B27	FBD3NBIP[10]		
A29	FBD3NBIN[11]			B28	FBD3NBIN[10]		
A30	VSS			B29	VSS		
A31	FBD2NBIP[3]			B30	FBD2NBIP[2]		
A32	FBD2NBIN[3]			B31	FBD2NBIN[2]		
A33	VSS			B32	VSS		
A34	FBD2NBIP[12]			B33	FBD2NBIP[13]		
A35	FBD2NBIN[12]			B34	FBD2NBIN[13]		
A36	VSS			B35	VSS		
A37	VSS			B36	FBD2NBIP[6]		
A38	VSS			B37	FBD2NBIN[6]		
A39	VSS			B38	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 2 of 20)

Ball No.	Signal Name	Buffer Type	Direction
B39	VSS		
C1	VSS		
C2	PE6RN[3]		
C3	PE6RP[3]		
C4	VSS		
C5	PE6TN[2]		
C6	PE6TP[2]		
C7	VSS		
C8	PE6RN[0]		
C9	PE5TN[2]		
C10	VSS		
C11	PE5RN[0]		
C12	VID[6]		
C13	RESETI_N		
C14	CFGSMBDATA		
C15	SPARE_IO		
C16	VSS		
C17	XDPDSTBN_N		
C18	XDPDATA_N[11]		
C19	TDI		
C20	VSS		
C21	XOROUT		
C22	VSS		
C23	FBD3NBIP[5]		
C24	FBD3NBIN[5]		
C25	VSS		
C26	FBD3NBIN[12]		
C27	VCCAFBD23PLL		
C28	VSS		
C29	FBD2NBIP[1]		
C30	FBD2NBIN[1]		
C31	VSS		
C32	FBD2NBIP[7]		
C33	FBD2NBIN[7]		
C34	VSS		
C35	FBD2NBIP[10]		
C36	FBD2NBIN[10]		
C37	VSS		
C38	FBD2SBOP[0]		
C39	VSS		
D1	PE7TN[1]		
D2	PE7TP[1]		
D3	VSS		
D4	PE6RN[2]		
D5	PE6RP[2]		
D6	VSS		
D7	PE6TN[1]		
D8	PE6RP[0]		
D9	VSS		
D10	PE5TN[1]		
D11	PE5RP[0]		
D12	VID[5]		
D13	VSS		
D14	ERR0_N		
D15	THERMALERT_N		
D16	XDPDATA_N[6]		
D17	XDPDATA_N[7]		
D18	VSS		
D19	TESTLO[8]		
D20	TCK		
D21	FSBSLWCTRL		
D22	FBD3NBIP[4]		
D23	FBD3NBIN[4]		
D24	VSS		
D25	FBD3NBIN[13]		
D26	FBD3NBIP[12]		
D27	VSS		
D28	FBD2NBIP[0]		
D29	FBD2NBIN[0]		
D30	VSS		
D31	FBD2NBIP[5]		
D32	FBD2NBIN[5]		
D33	VSS		
D34	FBD2NBIP[9]		
D35	FBD2NBIN[9]		
D36	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 3 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
D37	FBD2SBOP[3]			E36	FBD2SBOP[4]		
D38	FBD2SBON[0]			E37	FBD2SBON[3]		
D39	VSS			E38	VSS		
E1	VCCAPE1PLL18			E39	FBD2SBOP[1]		
E2	VSS			F1	VSS		
E3	PE7RP[1]			F2	VCCAPE1PLL125		
E4	VSS			F3	PE7RN[1]		
E5	VSS			F4	VSS		
E6	PE6RN[1]			F5	VSSABGPE1		
E7	PE6TP[1]			F6	PE6RP[1]		
E8	VSS			F7	VSS		
E9	PE6TN[0]			F8	PE7TP[2]		
E10	PE5TP[1]			F9	PE6TP[0]		
E11	VSS			F10	VSS		
E12	VID[4]			F11	PE5TN[0]		
E13	SPD3SMBCLK			F12	VSS		
E14	ERR1_N			F13	SPD3SMBDATA		
E15	VSS			F14	VCCMISC33		
E16	XDPDATA_N[3]			F15	ERR2_N		
E17	VSS			F16	XDPDATA_N[4]		
E18	XDPDATA_N[8]			F17	XDPDATA_N[0]		
E19	VSS			F18	XDPDATA_N[5]		
E20	VCCMISC125			F19	TESTLO[9]		
E21	RESERVED[27]			F20	PSEL[2]		
E22	TESTLO[11]			F21	VSS		
E23	VSS			F22	VSS		
E24	FBD3NBIN[3]			F23	FBD3NBIN[2]		
E25	FBD3NBIP[13]			F24	FBD3NBIP[3]		
E26	VSS			F25	VSS		
E27	FBD3NBIP[7]			F26	FBD3NBIP[6]		
E28	FBD3NBIN[7]			F27	FBD3NBIN[6]		
E29	VSS			F28	VSS		
E30	FBD2NBIP[4]			F29	FBD23CLKN		
E31	FBD2NBIN[4]			F30	FBD23CLKP		
E32	VSS			F31	VSS		
E33	FBD2NBIP[8]			F32	RESERVED[24]		
E34	FBD2NBIN[8]			F33	VCCAFBD23PLL18		
E35	VSS			F34	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 4 of 20)

Ball No.	Signal Name	Buffer Type	Direction
F35	FBD2SBOP[9]		
F36	FBD2SBON[4]		
F37	VSS		
F38	FBD2SBOP[2]		
F39	FBD2SBON[1]		
G1	PE7RP[2]		
G2	VSSAPE1PLL		
G3	VSS		
G4	PE1CLKP		
G5	VCCABGPE133		
G6	VSS		
G7	VSS		
G8	PE7TN[2]		
G9	VSS		
G10	PE7TP[0]		
G11	PE5TP[0]		
G12	VID[3]		
G13	SPD2SMBCLK		
G14	VSS		
G15	SPD1SMBCLK		
G16	VSS		
G17	XDPVREF		
G18	VSS		
G19	VCCMISC125		
G20	PSEL[1]		
G21	TESTLO[10]		
G22	FBD3NBIN[1]		
G23	FBD3NBIP[2]		
G24	VSS		
G25	FBD3NBIN[0]		
G26	FBD3NBIP[0]		
G27	VSS		
G28	FBD3SBOP[0]		
G29	FBD3SBON[0]		
G30	VSS		
G31	FBD2NBIP[11]		
G32	FBD2NBIN[11]		
G33	VSS		
G34	VCCMISC125		
G35	FBD2SBON[9]		
G36	VSS		
G37	FBD2SBOP[5]		
G38	FBD2SBON[2]		
G39	VSS		
H1	PE7RN[2]		
H2	VSS		
H3	PE7TP[3]		
H4	PE1CLKN		
H5	VSS		
H6	PE7RP[3]		
H7	PE7RN[3]		
H8	VSS		
H9	PE7RP[0]		
H10	PE7TN[0]		
H11	VSS		
H12	VID[2]		
H13	SPD2SMBDATA		
H14	VCCMISC33		
H15	SPD1SMBDATA		
H16	VCCMISC33		
H17	XDPDATA_N[2]		
H18	XDPDATA_N[14]		
H19	VSS		
H20	VSS		
H21	VSS		
H22	FBD3NBIP[1]		
H23	VSS		
H24	RESERVED[47]		
H25	RESERVED[45]		
H26	VSS		
H27	FBD3SBON[3]		
H28	FBD3SBOP[3]		
H29	VSS		
H30	FBDICOMP		
H31	VSS		
H32	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 5 of 20)

Ball No.	Signal Name	Buffer Type	Direction
H33	TESTLO[3]		
H34	TESTLO[4]		
H35	VSS		
H36	FBD2SBOP[6]		
H37	FBD2SBON[5]		
H38	VSS		
H39	FBD1SBOP[0]		
J1	VSS		
J2	PE8TP[2]		
J3	PE7TN[3]		
J4	VSS		
J5	PE8TN[1]		
J6	PE8TP[1]		
J7	VSS		
J8	PE8RP[0]		
J9	PE7RN[0]		
J10	VSS		
J11	PE8TP[0]		
J12	VID[1]		
J13	VSS		
J14	PLLPWRDET		
J15	VCCMISC33		
J16	VSS		
J17	VSS		
J18	VCCMISC125		
J19	VSCALEN		
J20	VSS		
J21	VSS		
J22	VSS		
J23	FBD3SBON[6]		
J24	FBD3SBOP[6]		
J25	VSS		
J26	FBD3SBON[4]		
J27	FBD3SBOP[4]		
J28	VSS		
J29	FBD3SBON[1]		
J30	FBD3SBOP[1]		
J31	VSS		
J32	FBD1NBIP[8]		
J33	FBD1NBIN[8]		
J34	VSS		
J35	FBD2SBOP[7]		
J36	FBD2SBON[6]		
J37	VSS		
J38	FBD1SBOP[1]		
J39	FBD1SBON[0]		
K1	PE8RP[2]		
K2	PE8TN[2]		
K3	VSS		
K4	PE8RN[1]		
K5	PE8RP[1]		
K6	VSS		
K7	PE8TP[3]		
K8	PE8RN[0]		
K9	VSS		
K10	PE8RP[3]		
K11	PE8TN[0]		
K12	VSS		
K13	SPD0SMBDATA		
K14	SPD0SMBCLK		
K15	VSS		
K16	VSS		
K17	XDPDATA_N[1]		
K18	XDPCLK1X		
K19	VSS		
K20	PSEL[0]		
K21	VSS		
K22	FBD3SBON[7]		
K23	FBD3SBOP[7]		
K24	VSS		
K25	FBD3SBON[9]		
K26	FBD3SBOP[9]		
K27	VSS		
K28	FBD3SBON[2]		
K29	FBD3SBOP[2]		
K30	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 6 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
K31	FBD1NBIP[7]			L30	FBD1NBIP[6]		
K32	FBD1NBIN[7]			L31	FBD1NBIN[6]		
K33	VSS			L32	VSS		
K34	FBD2SBOP[8]			L33	RESERVED[13]		
K35	FBD2SBON[7]			L34	FBD2SBON[8]		
K36	VSS			L35	VSS		
K37	FBD1SBOP[2]			L36	TESTLO[5]		
K38	FBD1SBON[1]			L37	FBD1SBON[2]		
K39	VSS			L38	VSS		
L1	PE8RN[2]			L39	FBD1SBON[3]		
L2	VSS			M1	VSS		
L3	PE9RP[2]			M2	PE9TP[2]		
L4	PE9RN[2]			M3	PE9TN[2]		
L5	VSS			M4	VSS		
L6	PE9TP[3]			M5	PE9RN[1]		
L7	PE8TN[3]			M6	PE9TN[3]		
L8	VSS			M7	VSS		
L9	PE9TN[1]			M8	PE9RP[3]		
L10	PE8RN[3]			M9	PE9TP[1]		
L11	VSS			M10	VSS		
L12	PE1ICOMPI			M11	PE9TN[0]		
L13	PE1RCOMPO			M12	VCCAPE125		
L14	THERMTRIP_N			M13	VCCAPE125		
L15	VCCRPE125			M14	VSS		
L16	VSS			M15	VCCRPE125		
L17	VSS			M16	VSS		
L18	XDPRDYREQ_N			M17	VCC		
L19	XDPDATA_N[12]			M18	VCC		
L20	VSS			M19	VSS		
L21	FBD3SBON[8]			M20	VCCAFBD		
L22	FBD3SBOP[8]			M21	VCCAFBD		
L23	VSS			M22	VCCAFBD		
L24	FBD3SBON[5]			M23	VCCAFBD		
L25	FBD3SBOP[5]			M24	VCCAFBD		
L26	VSS			M25	VCCAFBD		
L27	VCCSEN			M26	VCCAFBD		
L28	VSSSEN			M27	VCCAFBD		
L29	VSS			M28	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 7 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
M29	FBD1NBIP[12]			N28	FBD1NBIP[13]		
M30	FBD1NBIN[12]			N29	FBD1NBIN[13]		
M31	VSS			N30	VSS		
M32	FBD1NBIN[9]			N31	FBD1NBIN[5]		
M33	RESERVED[12]			N32	FBD1NBIP[9]		
M34	VSS			N33	VSS		
M35	TESTLO[7]			N34	FBD1NBIP[11]		
M36	TESTLO[6]			N35	FBD1NBIN[11]		
M37	VSS			N36	VSS		
M38	FBD1SBOP[4]			N37	FBD1SBOP[9]		
M39	FBD1SBOP[3]			N38	FBD1SBON[4]		
N1	PE1TN[1]			N39	VSS		
N2	PE1TP[1]			P1	RESERVED[41]		
N3	VSS			P2	VSS		
N4	PE1RP[1]			P3	PE1TP[2]		
N5	PE9RP[1]			P4	PE1RN[1]		
N6	VSS			P5	VSS		
N7	PE1RP[0]			P6	PE1RP[2]		
N8	PE9RN[3]			P7	PE1RN[0]		
N9	VSS			P8	VSS		
N10	PE9RN[0]			P9	PE1TP[0]		
N11	PE9TP[0]			P10	PE9RP[0]		
N12	VCCAPE125			P11	VSS		
N13	VCCAPE125			P12	VCCAPE125		
N14	VCCAPE125			P13	VCCAPE125		
N15	VCCRPE125			P14	VCCAPE125		
N16	VCCRPE125			P15	VSS		
N17	VCC			P16	VCCRPE125		
N18	VSS			P17	VSS		
N19	VCC			P18	VCC		
N20	VCCAFBD			P19	VSS		
N21	VCCAFBD			P20	VCC		
N22	VCCAFBD			P21	VSS		
N23	VCCAFBD			P22	VCCAFBD		
N24	VCCAFBD			P23	VCCAFBD		
N25	VCCAFBD			P24	VCCAFBD		
N26	VCCAFBD			P25	VCCAFBD		
N27	VCCAFBD			P26	VCCAFBD		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 8 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
P27	VCCAFBD			R26	VCCAFBD		
P28	VCCAFBD			R27	VCCAFBD		
P29	VSS			R28	VSS		
P30	FBD1NBIN[4]			R29	FBD1NBIN[3]		
P31	FBD1NBIP[5]			R30	FBD1NBIP[4]		
P32	VSS			R31	VSS		
P33	FBD1NBIP[10]			R32	FBD1NBIN[0]		
P34	FBD1NBIN[10]			R33	FBD1NBIP[0]		
P35	VSS			R34	VSS		
P36	FBD1SBOP[5]			R35	FBD1SBOP[8]		
P37	FBD1SBON[9]			R36	FBD1SBON[5]		
P38	VSS			R37	VSS		
P39	VCCAFBD01PLL			R38	FBD1SBON[6]		
R1	VSS			R39	FBD1SBOP[6]		
R2	PE2TP[1]			T1	PE2RP[1]		
R3	PE1TN[2]			T2	PE2TN[1]		
R4	VSS			T3	VSS		
R5	PE1TP[3]			T4	PE2RP[2]		
R6	PE1RN[2]			T5	PE1TN[3]		
R7	VSS			T6	VSS		
R8	PE2RP[0]			T7	PE2TP[3]		
R9	PE1TN[0]			T8	PE2RN[0]		
R10	VSS			T9	VSS		
R11	PE1RP[3]			T10	PE2TP[0]		
R12	VCCAPE125			T11	PE1RN[3]		
R13	VCCAPE125			T12	VSS		
R14	VCCAPE125			T13	VCCAPE125		
R15	VCCRPE125			T14	VCCAPE125		
R16	VSS			T15	VCCRPE125		
R17	VCC			T16	VCC		
R18	VSS			T17	VSS		
R19	VCC			T18	VCC		
R20	VSS			T19	VSS		
R21	VCC			T20	VCC		
R22	VSS			T21	VSS		
R23	VCC			T22	VCC		
R24	VSS			T23	VSS		
R25	VCC			T24	VCC		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 9 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
T25	VSS			U24	VSS		
T26	VCCAFBD			U25	VCC		
T27	VCCAFBD			U26	VCCAFBD		
T28	RESERVED[11]			U27	VCCAFBD		
T29	FBD1NBIP[3]			U28	VCCAFBD		
T30	VSS			U29	VSS		
T31	FBD1NBIN[2]			U30	FBD1NBIN[1]		
T32	FBD1NBIP[2]			U31	FBD1NBIP[1]		
T33	VSS			U32	VSS		
T34	VCCAFBD01PLL18			U33	FBD01CLKP		
T35	FBD1SBON[8]			U34	FBD01CLKN		
T36	VSS			U35	VSS		
T37	FBD1SBON[7]			U36	RESERVED[44]		
T38	FBD1SBOP[7]			U37	RESERVED[46]		
T39	VSS			U38	VSS		
U1	PE2RN[1]			U39	FBD0SBOP[1]		
U2	VSS			V1	VSS		
U3	PE2TP[2]			V2	PE3TP[0]		
U4	PE2RN[2]			V3	PE2TN[2]		
U5	VSS			V4	VSS		
U6	PE3RN[0]			V5	VCCAPE0PLL18		
U7	PE2TN[3]			V6	PE3RP[0]		
U8	VSS			V7	VSS		
U9	PE2RP[3]			V8	PE3RN[2]		
U10	PE2TN[0]			V9	PE2RN[3]		
U11	VSS			V10	VSS		
U12	VCCAPE125			V11	PE3TP[3]		
U13	VCCAPE125			V12	VCCAPE125		
U14	VCCAPE125			V13	VCCAPE125		
U15	VCCRPE125			V14	VCCAPE125		
U16	VSS			V15	VSS		
U17	VCC			V16	VCC		
U18	VSS			V17	VSS		
U19	VCC			V18	VCC		
U20	VSS			V19	VSS		
U21	VCC			V20	VCC		
U22	VSS			V21	VSS		
U23	VCC			V22	VCC		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 10 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
V23	VSS			W22	VSS		
V24	VCC			W23	VCC		
V25	VSS			W24	VSS		
V26	VCCAFBD			W25	VCC		
V27	VCCAFBD			W26	VCCAFBD		
V28	VSS			W27	VCCAFBD		
V29	FBD0NBIP[6]			W28	FBD0NBIP[12]		
V30	FBD0NBIN[6]			W29	FBD0NBIN[12]		
V31	VSS			W30	VSS		
V32	FBD0NBIP[8]			W31	FBD0NBIP[7]		
V33	FBD0NBIN[8]			W32	FBD0NBIN[7]		
V34	VSS			W33	VSS		
V35	FBD0NBIP[11]			W34	FBD0NBIP[10]		
V36	FBD0NBIN[11]			W35	FBD0NBIN[10]		
V37	VSS			W36	VSS		
V38	FBD0SBOP[2]			W37	FBD0SBOP[3]		
V39	FBD0SBON[1]			W38	FBD0SBON[2]		
W1	VCCAPE0PLL125			W39	VSS		
W2	PE3TN[0]			Y1	VSSAPE0PLL		
W3	VSS			Y2	VSS		
W4	PE0CLKN			Y3	PE3TP[1]		
W5	PE0CLKP			Y4	PE3TN[1]		
W6	VSS			Y5	VSS		
W7	RESERVED[50]			Y6	VSSABGPE0		
W8	PE3RP[2]			Y7	RESERVED[49]		
W9	VSS			Y8	VSS		
W10	PE0RCOMPO			Y9	RESERVED[48]		
W11	PE3TN[3]			Y10	PE0ICOMPI		
W12	VCCAPE125			Y11	VSS		
W13	VCCAPE125			Y12	VCCAPE125		
W14	VCCAPE125			Y13	VCCAPE125		
W15	VCCRPE125			Y14	VCCAPE125		
W16	VSS			Y15	VCCRPE125		
W17	VCC			Y16	VCC		
W18	VSS			Y17	VSS		
W19	VCC			Y18	VCC		
W20	VSS			Y19	VSS		
W21	VCC			Y20	VCC		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 11 of 20)

Ball No.	Signal Name	Buffer Type	Direction
Y21	VSS		
Y22	VCC		
Y23	VSS		
Y24	VCC		
Y25	VSS		
Y26	VCCAFBD		
Y27	VCCAFBD		
Y28	VCCAFBD		
Y29	VSS		
Y30	FBD0NBIP[13]		
Y31	FBD0NBIN[13]		
Y32	VSS		
Y33	FBD0NBIP[9]		
Y34	FBD0NBIN[9]		
Y35	VSS		
Y36	VSS		
Y37	FBD0SBON[3]		
Y38	VSS		
Y39	VSS		
AA1	VSS		
AA2	PE3RN[1]		
AA3	PE3RP[1]		
AA4	VSS		
AA5	PE4RP[0]		
AA6	VCCABGPE033		
AA7	VSS		
AA8	PE3TP[2]		
AA9	RESERVED[51]		
AA10	VSS		
AA11	PE3RP[3]		
AA12	VCCAPE125		
AA13	VCCAPE125		
AA14	VCCAPE125		
AA15	VCCRPE125		
AA16	VSS		
AA17	VCC		
AA18	VSS		
AA19	VCC		
AA20	VSS		
AA21	VCC		
AA22	VSS		
AA23	VCC		
AA24	VSS		
AA25	VCC		
AA26	VCCAFBD		
AA27	VCCAFBD		
AA28	VSS		
AA29	FBD0NBIN[3]		
AA30	VSS		
AA31	VSS		
AA32	FBD0NBIP[5]		
AA33	FBD0NBIN[5]		
AA34	VSS		
AA35	FBD0SBOP[9]		
AA36	FBD0SBON[9]		
AA37	VSS		
AA38	FBD0SBOP[5]		
AA39	FBD0SBON[5]		
AB1	PE4TN[1]		
AB2	PE4TP[1]		
AB3	VSS		
AB4	PE4RP[1]		
AB5	PE4RN[0]		
AB6	VSS		
AB7	PE4TP[0]		
AB8	PE3TN[2]		
AB9	VSS		
AB10	PE4RP[3]		
AB11	PE3RN[3]		
AB12	VSS		
AB13	VCCAPE125		
AB14	VCCAPE125		
AB15	VSS		
AB16	VCC		
AB17	VSS		
AB18	VCC		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 12 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AB19	VSS		
AB20	VCC		
AB21	VSS		
AB22	VCC		
AB23	VSS		
AB24	VCC		
AB25	VSS		
AB26	VCCAFBD		
AB27	VCCAFBD		
AB28	FBD0NBIN[2]		
AB29	FBD0NBIP[3]		
AB30	VSS		
AB31	FBD0NBIP[4]		
AB32	FBD0NBIN[4]		
AB33	VSS		
AB34	FBD0SBOP[4]		
AB35	FBD0SBON[4]		
AB36	VSS		
AB37	FBD0SBOP[6]		
AB38	FBD0SBON[6]		
AB39	VSS		
AC1	RESERVED[42]		
AC2	VSS		
AC3	PE4TP[2]		
AC4	PE4RN[1]		
AC5	VSS		
AC6	PE4RP[2]		
AC7	PE4TN[0]		
AC8	VSS		
AC9	PE4TP[3]		
AC10	PE4RN[3]		
AC11	VSS		
AC12	VCCAPE125		
AC13	VCCAPE125		
AC14	VCCRPE125		
AC15	VCCRPE125		
AC16	VSS		
AC17	VCC		

Ball No.	Signal Name	Buffer Type	Direction
AC18	VSS		
AC19	VCC		
AC20	VSS		
AC21	VCC		
AC22	VSS		
AC23	VCC		
AC24	VSS		
AC25	VCC		
AC26	VCCAFBD		
AC27	VCCAFBD		
AC28	FBD0NBIP[2]		
AC29	VSS		
AC30	FBD0NBIN[1]		
AC31	FBD0NBIP[1]		
AC32	VSS		
AC33	FBD0SBOP[0]		
AC34	FBD0SBON[0]		
AC35	VSS		
AC36	FBD0SBOP[7]		
AC37	FBD0SBON[7]		
AC38	VSS		
AC39	VSS		
AD1	VSS		
AD2	PE0RP[1]		
AD3	PE4TN[2]		
AD4	VSS		
AD5	PE0RP[3]		
AD6	PE4RN[2]		
AD7	VSS		
AD8	PE0TP[0]		
AD9	PE4TN[3]		
AD10	VSS		
AD11	PE0RP[0]		
AD12	VCCAPE125		
AD13	VCCAPE125		
AD14	VCCRPE125		
AD15	VCCRPE125		
AD16	VCC		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 13 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AD17	VSS		
AD18	VCC		
AD19	VSS		
AD20	VCC		
AD21	VSS		
AD22	VCC		
AD23	VSS		
AD24	VCC		
AD25	VSS		
AD26	VCCAFBD		
AD27	VCCAFBD		
AD28	VSS		
AD29	FBD0NBIN[0]		
AD30	FBD0NBIP[0]		
AD31	VSS		
AD32	FSB0D_N[40]		
AD33	FSB0DBI_N[2]		
AD34	VSS		
AD35	FBD0SBOP[8]		
AD36	FBD0SBON[8]		
AD37	VSS		
AD38	FSB0D_N[63]		
AD39	FSB0D_N[62]		
AE1	PE0TP[1]		
AE2	PE0RN[1]		
AE3	VSS		
AE4	PE0TP[2]		
AE5	PE0RN[3]		
AE6	VSS		
AE7	VSS		
AE8	PE0TN[0]		
AE9	VSS		
AE10	PE0TP[3]		
AE11	PE0RN[0]		
AE12	VSS		
AE13	VCCSF		
AE14	VCCSF		
AE15	VTT		
AE16	VTT		
AE17	VTT		
AE18	VTT		
AE19	VTT		
AE20	VTT		
AE21	VTT		
AE22	VTT		
AE23	VTT		
AE24	VTT		
AE25	VTT		
AE26	VCCSF		
AE27	VSS		
AE28	VCCSF		
AE29	FSB0D_N[46]		
AE30	VSS		
AE31	FSB0STBN_N[2]		
AE32	FSB0STBP_N[2]		
AE33	VSS		
AE34	FSB0D_N[39]		
AE35	FSB0D_N[58]		
AE36	VSS		
AE37	FSB0D_N[59]		
AE38	FSB0D_N[48]		
AE39	VSS		
AF1	PE0TN[1]		
AF2	VSS		
AF3	PE0RP[2]		
AF4	PE0TN[2]		
AF5	VSS		
AF6	FSB1A_N[14]		
AF7	FSB1A_N[16]		
AF8	VSS		
AF9	FSB1A_N[13]		
AF10	PE0TN[3]		
AF11	VSS		
AF12	VCCSF		
AF13	VCCSF		
AF14	VCCSF		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 14 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AF15	VTT		
AF16	VTT		
AF17	VTT		
AF18	VTT		
AF19	VTT		
AF20	VTT		
AF21	VTT		
AF22	VTT		
AF23	VTT		
AF24	VTT		
AF25	VTT		
AF26	VCCSF		
AF27	VCCSF		
AF28	VCCSF		
AF29	VSS		
AF30	FSB0D_N[42]		
AF31	FSB0D_N[44]		
AF32	VSS		
AF33	FSB0D_N[41]		
AF34	FSB0D_N[38]		
AF35	VSS		
AF36	FSB0D_N[54]		
AF37	FSB0D_N[60]		
AF38	VSS		
AF39	FSB0D_N[61]		
AG1	VSS		
AG2	VSS		
AG3	PE0RN[2]		
AG4	VSS		
AG5	FSB1A_N[10]		
AG6	FSB1A_N[15]		
AG7	VSS		
AG8	FSB1A_N[11]		
AG9	FSB1A_N[36]		
AG10	VSS		
AG11	FSB1REQ_N[4]		
AG12	VCCSF		
AG13	VSS		
AG14	FSB1DP_N[3]		
AG15	FSB1RESET_N		
AG16	VSS		
AG17	FSB1D_N[62]		
AG18	FSB1D_N[58]		
AG19	VTT		
AG20	VTT		
AG21	VTT		
AG22	VTT		
AG23	FSB0A_N[15]		
AG24	FSB0A_N[14]		
AG25	VSS		
AG26	FSB0A_N[16]		
AG27	VCCSF		
AG28	VSS		
AG29	FSB0D_N[43]		
AG30	FSB0D_N[45]		
AG31	VSS		
AG32	FSB0D_N[35]		
AG33	FSB0D_N[37]		
AG34	VSS		
AG35	FSB0DBI_N[3]		
AG36	FSB0D_N[57]		
AG37	VSS		
AG38	FSB0STBN_N[3]		
AG39	FSB0STBP_N[3]		
AH1	FSB1A_N[34]		
AH2	VSS		
AH3	VSS		
AH4	FSB1A_N[35]		
AH5	FSB1A_N[12]		
AH6	VSS		
AH7	FSB1A_N[9]		
AH8	FSB1A_N[37]		
AH9	VSS		
AH10	FSB1REQ_N[3]		
AH11	FSB1REQ_N[1]		
AH12	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 15 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
AH13	FSB1DP_N[1]			AJ12	FSB1HIT_N		
AH14	FSB1DP_N[2]			AJ13	VCC		
AH15	VSS			AJ14	VSS		
AH16	FSB1D_N[59]			AJ15	FSB1DP_N[0]		
AH17	FSB1D_N[63]			AJ16	FSB1D_N[54]		
AH18	VSS			AJ17	VSS		
AH19	VTT			AJ18	FSB1D_N[48]		
AH20	VTT			AJ19	VTT		
AH21	VTT			AJ20	VTT		
AH22	VTT			AJ21	VTT		
AH23	VCC			AJ22	VTT		
AH24	VSS			AJ23	VSS		
AH25	VCC			AJ24	FSB0A_N[13]		
AH26	FSB0A_N[10]			AJ25	FSB0A_N[12]		
AH27	VSS			AJ26	VSS		
AH28	FSB0REQ_N[4]			AJ27	FSB0REQ_N[1]		
AH29	FSB0D_N[47]			AJ28	FSB0REQ_N[3]		
AH30	VSS			AJ29	VSS		
AH31	FSB0D_N[34]			AJ30	FSB0D_N[32]		
AH32	FSB0D_N[33]			AJ31	FSB0D_N[0]		
AH33	VSS			AJ32	VSS		
AH34	FSB0D_N[36]			AJ33	FSB0D_N[14]		
AH35	FSB0D_N[56]			AJ34	FSB0D_N[13]		
AH36	VSS			AJ35	VSS		
AH37	FSB0D_N[49]			AJ36	FSB0D_N[50]		
AH38	FSB0D_N[55]			AJ37	FSB0D_N[51]		
AH39	VSS			AJ38	VSS		
AJ1	FSB1A_N[32]			AJ39	FSB0D_N[53]		
AJ2	VSS			AK1	VSS		
AJ3	FSB1A_N[29]			AK2	FSB1A_N[30]		
AJ4	FSB1A_N[33]			AK3	FSB1A_N[31]		
AJ5	VSS			AK4	VSS		
AJ6	FSB1ADSTB_N[0]			AK5	FSB1A_N[27]		
AJ7	FSB1A_N[8]			AK6	FSB1A_N[4]		
AJ8	VSS			AK7	VSS		
AJ9	FSB1A_N[6]			AK8	FSB1A_N[5]		
AJ10	FSB1REQ_N[0]			AK9	FSB1A_N[3]		
AJ11	VSS			AK10	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 16 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AK11	FSB1TRDY_N		
AK12	FSB1HITM_N		
AK13	VSS		
AK14	FSB1D_N[16]		
AK15	FSB1BPRI_N		
AK16	VSS		
AK17	FSB1D_N[61]		
AK18	FSB1D_N[60]		
AK19	VTT		
AK20	VTT		
AK21	VTT		
AK22	VTT		
AK23	VSS		
AK24	FSB0A_N[9]		
AK25	VSS		
AK26	FSB0A_N[11]		
AK27	FSB0A_N[6]		
AK28	VSS		
AK29	FSB0AP_N[0]		
AK30	FSB0AP_N[1]		
AK31	VSS		
AK32	FSB0D_N[1]		
AK33	FSB0D_N[9]		
AK34	VSS		
AK35	FSB0FSBVREF[4]		
AK36	FSB0D_N[28]		
AK37	VSS		
AK38	FSB0D_N[30]		
AK39	FSB0D_N[52]		
AL1	FSB1A_N[28]		
AL2	FSB1A_N[39]		
AL3	VSS		
AL4	FSB1FSBVREF[0]		
AL5	FSB1ADSTB_N[1]		
AL6	VSS		
AL7	FSB1REQ_N[2]		
AL8	FSB1A_N[7]		
AL9	VSS		
AL10	FSB1RS_N[0]		
AL11	FSB1RS_N[2]		
AL12	VSS		
AL13	FSB1DEFER_N		
AL14	FSB1D_N[17]		
AL15	VSS		
AL16	FSB1D_N[57]		
AL17	FSB1DBI_N[3]		
AL18	VSS		
AL19	VTT		
AL20	VTT		
AL21	VTT		
AL22	VTT		
AL23	RESERVED[1]		
AL24	VSS		
AL25	FSB0A_N[37]		
AL26	FSB0A_N[3]		
AL27	VSS		
AL28	FSB0REQ_N[2]		
AL29	FSB0MCERR_N		
AL30	VSS		
AL31	FSB0D_N[2]		
AL32	FSB0D_N[3]		
AL33	VSS		
AL34	FSB0D_N[11]		
AL35	FSB0D_N[15]		
AL36	VSS		
AL37	FSB0D_N[25]		
AL38	FSB0D_N[31]		
AL39	VSS		
AM1	FSB1A_N[22]		
AM2	VSS		
AM3	FSB1A_N[38]		
AM4	FSB1A_N[25]		
AM5	VSS		
AM6	FSB1BREQ0_N		
AM7	FSB1BPM_N[4]		
AM8	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 17 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AM9	FSB1LOCK_N		
AM10	FSB1D_N[19]		
AM11	VSS		
AM12	FSB1D_N[18]		
AM13	FSB1D_N[20]		
AM14	VSS		
AM15	FSB1D_N[28]		
AM16	FSB1D_N[56]		
AM17	VSS		
AM18	FSB1STBP_N[3]		
AM19	VTT		
AM20	VTT		
AM21	VTT		
AM22	VTT		
AM23	VSS		
AM24	FSB0ADSTB_N[0]		
AM25	FSB0A_N[36]		
AM26	VSS		
AM27	FSB0A_N[5]		
AM28	FSB0REQ_N[0]		
AM29	VSS		
AM30	FSB0BINIT_N		
AM31	FSB0BREQ0_N		
AM32	VSS		
AM33	FSB0D_N[8]		
AM34	FSB0D_N[10]		
AM35	VSS		
AM36	FSB0D_N[27]		
AM37	FSB0D_N[29]		
AM38	VSS		
AM39	FSB0D_N[26]		
AN1	VSS		
AN2	FSB1A_N[26]		
AN3	FSB1A_N[24]		
AN4	VSS		
AN5	FSB1BINIT_N		
AN6	FSB1RS_N[1]		
AN7	VSS		
AN8	FSB1BNR_N		
AN9	FSB1DBSY_N		
AN10	VSS		
AN11	FSB1D_N[23]		
AN12	FSB1D_N[21]		
AN13	VSS		
AN14	FSB1D_N[25]		
AN15	FSB1D_N[30]		
AN16	VSS		
AN17	FSB1D_N[52]		
AN18	FSB1STBN_N[3]		
AN19	VTT		
AN20	VTT		
AN21	VTT		
AN22	VTT		
AN23	CORECLKP		
AN24	FSB0A_N[8]		
AN25	VSS		
AN26	FSB0A_N[4]		
AN27	FSB0FSBVREF[0]		
AN28	VSS		
AN29	FSB0RSP_N		
AN30	FSB0FSBVREF[2]		
AN31	VSS		
AN32	FSB0D_N[5]		
AN33	FSB0DBI_N[0]		
AN34	VSS		
AN35	FSB0D_N[16]		
AN36	FSB0D_N[17]		
AN37	VSS		
AN38	FSB0D_N[24]		
AN39	FSB0DBI_N[1]		
AP1	FSB1A_N[21]		
AP2	FSB1A_N[17]		
AP3	VSS		
AP4	FSB1A_N[18]		
AP5	FSB1RSP_N		
AP6	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 18 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AP7	FSB1ADS_N		
AP8	FSB1BPM_N[5]		
AP9	VSS		
AP10	FSB1STBP_N[1]		
AP11	FSB1D_N[22]		
AP12	VSS		
AP13	FSB1D_N[29]		
AP14	FSB1D_N[31]		
AP15	VSS		
AP16	FSB1D_N[50]		
AP17	FSB1D_N[51]		
AP18	VSS		
AP19	VTT		
AP20	VTT		
AP21	VTT		
AP22	VTT		
AP23	CORECLKN		
AP24	VSS		
AP25	FSB0A_N[18]		
AP26	FSB0A_N[7]		
AP27	VSS		
AP28	FSB0A_N[19]		
AP29	FSB0RS_N[1]		
AP30	VSS		
AP31	FSB0RESET_N		
AP32	FSB0D_N[4]		
AP33	VSS		
AP34	FSB0STBN_N[0]		
AP35	FSB0STBP_N[0]		
AP36	VSS		
AP37	FSB0D_N[19]		
AP38	FSB0D_N[23]		
AP39	VSS		
AR1	FSB1A_N[23]		
AR2	VSS		
AR3	FSB1A_N[19]		
AR4	FSB1MCERR_N		
AR5	VSS		
AR6	FSB1BREQ1_N		
AR7	FSB1DRDY_N		
AR8	VSS		
AR9	FSB1D_N[24]		
AR10	FSB1STBN_N[1]		
AR11	VSS		
AR12	FSB1DBI_N[1]		
AR13	FSB1D_N[26]		
AR14	VSS		
AR15	FSB1D_N[43]		
AR16	FSB1D_N[47]		
AR17	VSS		
AR18	FSB1D_N[49]		
AR19	VTT		
AR20	VTT		
AR21	VTT		
AR22	VTT		
AR23	VSS		
AR24	FSB0A_N[34]		
AR25	FSB0A_N[20]		
AR26	VSS		
AR27	FSB0A_N[23]		
AR28	FSB0A_N[21]		
AR29	VSS		
AR30	FSB0BPM_N[4]		
AR31	FSB0DP_N[3]		
AR32	VSS		
AR33	FSB0D_N[6]		
AR34	FSB0D_N[7]		
AR35	VSS		
AR36	FSB0D_N[20]		
AR37	FSB0D_N[21]		
AR38	VSS		
AR39	FSB0STBN_N[1]		
AT1	VSS		
AT2	FSB1A_N[20]		
AT3	FSB1AP_N[1]		
AT4	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 19 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AT5	FSB1D_N[0]		
AT6	FSB1D_N[2]		
AT7	VSS		
AT8	FSB1STBN_N[0]		
AT9	FSB1D_N[11]		
AT10	VSS		
AT11	FSB1D_N[36]		
AT12	FSB1D_N[27]		
AT13	VSS		
AT14	FSB1D_N[38]		
AT15	FSB1D_N[40]		
AT16	VSS		
AT17	FSB1D_N[53]		
AT18	FSB1D_N[55]		
AT19	VTT		
AT20	VTT		
AT21	VTT		
AT22	VTT		
AT23	VCCDCPLL		
AT24	FSB0A_N[32]		
AT25	VSS		
AT26	FSB0A_N[17]		
AT27	FSB0A_N[22]		
AT28	VSS		
AT29	FSB0BREQ1_N		
AT30	FSB0BPM_N[5]		
AT31	VSS		
AT32	FSB0DP_N[2]		
AT33	FSB0DP_N[0]		
AT34	VSS		
AT35	FSB0D_N[12]		
AT36	FSB0D_N[18]		
AT37	VSS		
AT38	FSB0D_N[22]		
AT39	FSB0STBP_N[1]		
AU1	VSS		
AU2	FSB1AP_N[0]		
AU3	VSS		
AU4	FSB1D_N[3]		
AU5	FSB1D_N[5]		
AU6	VSS		
AU7	FSB1D_N[12]		
AU8	FSB1STBP_N[0]		
AU9	VSS		
AU10	FSB1D_N[14]		
AU11	FSB1D_N[32]		
AU12	VSS		
AU13	FSB1D_N[35]		
AU14	FSB1DBI_N[2]		
AU15	VSS		
AU16	FSB1D_N[41]		
AU17	FSB1D_N[42]		
AU18	VSS		
AU19	VTT		
AU20	VTT		
AU21	VTT		
AU22	VTT		
AU23	VCCACPLL		
AU24	VSS		
AU25	FSB0A_N[31]		
AU26	FSB0A_N[27]		
AU27	VSS		
AU28	FSB0A_N[25]		
AU29	FSB0A_N[24]		
AU30	VSS		
AU31	FSB0LOCK_N		
AU32	FSB0DP_N[1]		
AU33	VSS		
AU34	FSB0DEFER_N		
AU35	FSB0HIT_N		
AU36	VSS		
AU37	RESERVED[0]		
AU38	FSBDRVCRES		
AU39	VSS		
AV1	VSS		
AV2	VSS		

Table 8-1. Intel 5400 Chipset MCH Signals By Ball Number (Sheet 20 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AV3	FSB1D_N[1]		
AV4	FSB1D_N[4]		
AV5	VSS		
AV6	FSB1D_N[7]		
AV7	FSB1FSBVREF[2]		
AV8	VSS		
AV9	FSB1D_N[15]		
AV10	FSB1D_N[9]		
AV11	VSS		
AV12	FSB1D_N[34]		
AV13	FSB1D_N[37]		
AV14	VSS		
AV15	FSB1STBP_N[2]		
AV16	FSB1FSBVREF[4]		
AV17	VSS		
AV18	FSB1D_N[46]		
AV19	VTT		
AV20	VTT		
AV21	VTT		
AV22	VTT		
AV23	VSS		
AV24	FSB0A_N[33]		
AV25	FSB0A_N[30]		
AV26	VSS		
AV27	FSB0A_N[39]		
AV28	FSB0A_N[38]		
AV29	VSS		
AV30	FSB0DRDY_N		
AV31	FSB0BNR_N		
AV32	VSS		
AV33	FSB0RS_N[2]		
AV34	FSB0TRDY_N		
AV35	VSS		
AV36	FSBODTCRES		
AV37	FSBSLWCRES		
AV38	VSS		
AV39	VSS		
AW1	VSS		
AW2	VSS		
AW3	VSS		
AW4	VSS		
AW5	FSB1D_N[6]		
AW6	FSB1DBI_N[0]		
AW7	VSS		
AW8	FSB1D_N[10]		
AW9	FSB1D_N[8]		
AW10	VSS		
AW11	FSB1D_N[13]		
AW12	FSB1D_N[33]		
AW13	VSS		
AW14	FSB1D_N[39]		
AW15	FSB1STBN_N[2]		
AW16	VSS		
AW17	FSB1D_N[44]		
AW18	FSB1D_N[45]		
AW19	VTT		
AW20	VTT		
AW21	VTT		
AW22	VTT		
AW23	FSB0A_N[35]		
AW24	FSB0A_N[29]		
AW25	VSS		
AW26	FSB0A_N[28]		
AW27	FSB0ADSTB_N[1]		
AW28	VSS		
AW29	FSB0A_N[26]		
AW30	FSB0ADS_N		
AW31	VSS		
AW32	FSB0DBSY_N		
AW33	FSB0RS_N[0]		
AW34	VSS		
AW35	FSB0HITM_N		
AW36	FSB0BPRI_N		
AW37	VSS		
AW38	VSS		
AW39	VSS		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 1 of 20)

Ball No.	Signal Name	Buffer Type	Direction
A15	CFGSMBCLK		
C14	CFGSMBDATA		
AP23	CORECLKN		
AN23	CORECLKP		
D14	ERR0_N		
E14	ERR1_N		
F15	ERR2_N		
U34	FBD01CLKN		
U33	FBD01CLKP		
AD29	FBD0NBIN[0]		
AC30	FBD0NBIN[1]		
W35	FBD0NBIN[10]		
V36	FBD0NBIN[11]		
W29	FBD0NBIN[12]		
Y31	FBD0NBIN[13]		
AB28	FBD0NBIN[2]		
AA29	FBD0NBIN[3]		
AB32	FBD0NBIN[4]		
AA33	FBD0NBIN[5]		
V30	FBD0NBIN[6]		
W32	FBD0NBIN[7]		
V33	FBD0NBIN[8]		
Y34	FBD0NBIN[9]		
AD30	FBD0NBIP[0]		
AC31	FBD0NBIP[1]		
W34	FBD0NBIP[10]		
V35	FBD0NBIP[11]		
W28	FBD0NBIP[12]		
Y30	FBD0NBIP[13]		
AC28	FBD0NBIP[2]		
AB29	FBD0NBIP[3]		
AB31	FBD0NBIP[4]		
AA32	FBD0NBIP[5]		
V29	FBD0NBIP[6]		
W31	FBD0NBIP[7]		
V32	FBD0NBIP[8]		
Y33	FBD0NBIP[9]		
AC34	FBD0SBON[0]		
V39	FBD0SBON[1]		
W38	FBD0SBON[2]		
Y37	FBD0SBON[3]		
AB35	FBD0SBON[4]		
AA39	FBD0SBON[5]		
AB38	FBD0SBON[6]		
AC37	FBD0SBON[7]		
AD36	FBD0SBON[8]		
AA36	FBD0SBON[9]		
AC33	FBD0SBOP[0]		
U39	FBD0SBOP[1]		
V38	FBD0SBOP[2]		
W37	FBD0SBOP[3]		
AB34	FBD0SBOP[4]		
AA38	FBD0SBOP[5]		
AB37	FBD0SBOP[6]		
AC36	FBD0SBOP[7]		
AD35	FBD0SBOP[8]		
AA35	FBD0SBOP[9]		
R32	FBD1NBIN[0]		
U30	FBD1NBIN[1]		
P34	FBD1NBIN[10]		
N35	FBD1NBIN[11]		
M30	FBD1NBIN[12]		
N29	FBD1NBIN[13]		
T31	FBD1NBIN[2]		
R29	FBD1NBIN[3]		
P30	FBD1NBIN[4]		
N31	FBD1NBIN[5]		
L31	FBD1NBIN[6]		
K32	FBD1NBIN[7]		
J33	FBD1NBIN[8]		
M32	FBD1NBIN[9]		
R33	FBD1NBIP[0]		
U31	FBD1NBIP[1]		
P33	FBD1NBIP[10]		
N34	FBD1NBIP[11]		
M29	FBD1NBIP[12]		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 2 of 20)

Ball No.	Signal Name	Buffer Type	Direction
N28	FBD1NBIP[13]		
T32	FBD1NBIP[2]		
T29	FBD1NBIP[3]		
R30	FBD1NBIP[4]		
P31	FBD1NBIP[5]		
L30	FBD1NBIP[6]		
K31	FBD1NBIP[7]		
J32	FBD1NBIP[8]		
N32	FBD1NBIP[9]		
J39	FBD1SBON[0]		
K38	FBD1SBON[1]		
L37	FBD1SBON[2]		
L39	FBD1SBON[3]		
N38	FBD1SBON[4]		
R36	FBD1SBON[5]		
R38	FBD1SBON[6]		
T37	FBD1SBON[7]		
T35	FBD1SBON[8]		
P37	FBD1SBON[9]		
H39	FBD1SBOP[0]		
J38	FBD1SBOP[1]		
K37	FBD1SBOP[2]		
M39	FBD1SBOP[3]		
M38	FBD1SBOP[4]		
P36	FBD1SBOP[5]		
R39	FBD1SBOP[6]		
T38	FBD1SBOP[7]		
R35	FBD1SBOP[8]		
N37	FBD1SBOP[9]		
F29	FBD23CLKN		
F30	FBD23CLKP		
D29	FBD2NBIN[0]		
C30	FBD2NBIN[1]		
C36	FBD2NBIN[10]		
G32	FBD2NBIN[11]		
A35	FBD2NBIN[12]		
B34	FBD2NBIN[13]		
B31	FBD2NBIN[2]		
A32	FBD2NBIN[3]		
E31	FBD2NBIN[4]		
D32	FBD2NBIN[5]		
B37	FBD2NBIN[6]		
C33	FBD2NBIN[7]		
E34	FBD2NBIN[8]		
D35	FBD2NBIN[9]		
D28	FBD2NBIP[0]		
C29	FBD2NBIP[1]		
C35	FBD2NBIP[10]		
G31	FBD2NBIP[11]		
A34	FBD2NBIP[12]		
B33	FBD2NBIP[13]		
B30	FBD2NBIP[2]		
A31	FBD2NBIP[3]		
E30	FBD2NBIP[4]		
D31	FBD2NBIP[5]		
B36	FBD2NBIP[6]		
C32	FBD2NBIP[7]		
E33	FBD2NBIP[8]		
D34	FBD2NBIP[9]		
D38	FBD2SBON[0]		
F39	FBD2SBON[1]		
G38	FBD2SBON[2]		
E37	FBD2SBON[3]		
F36	FBD2SBON[4]		
H37	FBD2SBON[5]		
J36	FBD2SBON[6]		
K35	FBD2SBON[7]		
L34	FBD2SBON[8]		
G35	FBD2SBON[9]		
C38	FBD2SBOP[0]		
E39	FBD2SBOP[1]		
F38	FBD2SBOP[2]		
D37	FBD2SBOP[3]		
E36	FBD2SBOP[4]		
G37	FBD2SBOP[5]		
H36	FBD2SBOP[6]		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 3 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
J35	FBD2SBOP[7]			K22	FBD3SBON[7]		
K34	FBD2SBOP[8]			L21	FBD3SBON[8]		
F35	FBD2SBOP[9]			K25	FBD3SBON[9]		
G25	FBD3NBIN[0]			G28	FBD3SBOP[0]		
G22	FBD3NBIN[1]			J30	FBD3SBOP[1]		
B28	FBD3NBIN[10]			K29	FBD3SBOP[2]		
A29	FBD3NBIN[11]			H28	FBD3SBOP[3]		
C26	FBD3NBIN[12]			J27	FBD3SBOP[4]		
D25	FBD3NBIN[13]			L25	FBD3SBOP[5]		
F23	FBD3NBIN[2]			J24	FBD3SBOP[6]		
E24	FBD3NBIN[3]			K23	FBD3SBOP[7]		
D23	FBD3NBIN[4]			L22	FBD3SBOP[8]		
C24	FBD3NBIN[5]			K26	FBD3SBOP[9]		
F27	FBD3NBIN[6]			H30	FBDICOMP		
E28	FBD3NBIN[7]			AH26	FSB0A_N[10]		
B25	FBD3NBIN[8]			AK26	FSB0A_N[11]		
A26	FBD3NBIN[9]			AJ25	FSB0A_N[12]		
G26	FBD3NBIP[0]			AJ24	FSB0A_N[13]		
H22	FBD3NBIP[1]			AG24	FSB0A_N[14]		
B27	FBD3NBIP[10]			AG23	FSB0A_N[15]		
A28	FBD3NBIP[11]			AG26	FSB0A_N[16]		
D26	FBD3NBIP[12]			AT26	FSB0A_N[17]		
E25	FBD3NBIP[13]			AP25	FSB0A_N[18]		
G23	FBD3NBIP[2]			AP28	FSB0A_N[19]		
F24	FBD3NBIP[3]			AR25	FSB0A_N[20]		
D22	FBD3NBIP[4]			AR28	FSB0A_N[21]		
C23	FBD3NBIP[5]			AT27	FSB0A_N[22]		
F26	FBD3NBIP[6]			AR27	FSB0A_N[23]		
E27	FBD3NBIP[7]			AU29	FSB0A_N[24]		
B24	FBD3NBIP[8]			AU28	FSB0A_N[25]		
A25	FBD3NBIP[9]			AW29	FSB0A_N[26]		
G29	FBD3SBON[0]			AU26	FSB0A_N[27]		
J29	FBD3SBON[1]			AW26	FSB0A_N[28]		
K28	FBD3SBON[2]			AW24	FSB0A_N[29]		
H27	FBD3SBON[3]			AL26	FSB0A_N[3]		
J26	FBD3SBON[4]			AV25	FSB0A_N[30]		
L24	FBD3SBON[5]			AU25	FSB0A_N[31]		
J23	FBD3SBON[6]			AT24	FSB0A_N[32]		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 4 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AV24	FSB0A_N[33]		
AR24	FSB0A_N[34]		
AW23	FSB0A_N[35]		
AM25	FSB0A_N[36]		
AL25	FSB0A_N[37]		
AV28	FSB0A_N[38]		
AV27	FSB0A_N[39]		
AN26	FSB0A_N[4]		
AM27	FSB0A_N[5]		
AK27	FSB0A_N[6]		
AP26	FSB0A_N[7]		
AN24	FSB0A_N[8]		
AK24	FSB0A_N[9]		
AW30	FSB0ADS_N		
AM24	FSB0ADSTB_N[0]		
AW27	FSB0ADSTB_N[1]		
AK29	FSB0AP_N[0]		
AK30	FSB0AP_N[1]		
AM30	FSB0BINIT_N		
AV31	FSB0BNR_N		
AR30	FSB0BPM_N[4]		
AT30	FSB0BPM_N[5]		
AW36	FSB0BPRI_N		
AM31	FSB0BREQ0_N		
AT29	FSB0BREQ1_N		
AJ31	FSB0D_N[0]		
AK32	FSB0D_N[1]		
AM34	FSB0D_N[10]		
AL34	FSB0D_N[11]		
AT35	FSB0D_N[12]		
AJ34	FSB0D_N[13]		
AJ33	FSB0D_N[14]		
AL35	FSB0D_N[15]		
AN35	FSB0D_N[16]		
AN36	FSB0D_N[17]		
AT36	FSB0D_N[18]		
AP37	FSB0D_N[19]		
AL31	FSB0D_N[2]		

Ball No.	Signal Name	Buffer Type	Direction
AR36	FSB0D_N[20]		
AR37	FSB0D_N[21]		
AT38	FSB0D_N[22]		
AP38	FSB0D_N[23]		
AN38	FSB0D_N[24]		
AL37	FSB0D_N[25]		
AM39	FSB0D_N[26]		
AM36	FSB0D_N[27]		
AK36	FSB0D_N[28]		
AM37	FSB0D_N[29]		
AL32	FSB0D_N[3]		
AK38	FSB0D_N[30]		
AL38	FSB0D_N[31]		
AJ30	FSB0D_N[32]		
AH32	FSB0D_N[33]		
AH31	FSB0D_N[34]		
AG32	FSB0D_N[35]		
AH34	FSB0D_N[36]		
AG33	FSB0D_N[37]		
AF34	FSB0D_N[38]		
AE34	FSB0D_N[39]		
AP32	FSB0D_N[4]		
AD32	FSB0D_N[40]		
AF33	FSB0D_N[41]		
AF30	FSB0D_N[42]		
AG29	FSB0D_N[43]		
AF31	FSB0D_N[44]		
AG30	FSB0D_N[45]		
AE29	FSB0D_N[46]		
AH29	FSB0D_N[47]		
AE38	FSB0D_N[48]		
AH37	FSB0D_N[49]		
AN32	FSB0D_N[5]		
AJ36	FSB0D_N[50]		
AJ37	FSB0D_N[51]		
AK39	FSB0D_N[52]		
AJ39	FSB0D_N[53]		
AF36	FSB0D_N[54]		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 5 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
AH38	FSB0D_N[55]			AP29	FSB0RS_N[1]		
AH35	FSB0D_N[56]			AV33	FSB0RS_N[2]		
AG36	FSB0D_N[57]			AN29	FSB0RSP_N		
AE35	FSB0D_N[58]			AP34	FSB0STBN_N[0]		
AE37	FSB0D_N[59]			AR39	FSB0STBN_N[1]		
AR33	FSB0D_N[6]			AE31	FSB0STBN_N[2]		
AF37	FSB0D_N[60]			AG38	FSB0STBN_N[3]		
AF39	FSB0D_N[61]			AP35	FSB0STBP_N[0]		
AD39	FSB0D_N[62]			AT39	FSB0STBP_N[1]		
AD38	FSB0D_N[63]			AE32	FSB0STBP_N[2]		
AR34	FSB0D_N[7]			AG39	FSB0STBP_N[3]		
AM33	FSB0D_N[8]			AV34	FSB0TRDY_N		
AK33	FSB0D_N[9]			AG5	FSB1A_N[10]		
AN33	FSB0DBI_N[0]			AG8	FSB1A_N[11]		
AN39	FSB0DBI_N[1]			AH5	FSB1A_N[12]		
AD33	FSB0DBI_N[2]			AF9	FSB1A_N[13]		
AG35	FSB0DBI_N[3]			AF6	FSB1A_N[14]		
AW32	FSB0DBSY_N			AG6	FSB1A_N[15]		
AU34	FSB0DEFER_N			AF7	FSB1A_N[16]		
AT33	FSB0DP_N[0]			AP2	FSB1A_N[17]		
AU32	FSB0DP_N[1]			AP4	FSB1A_N[18]		
AT32	FSB0DP_N[2]			AR3	FSB1A_N[19]		
AR31	FSB0DP_N[3]			AT2	FSB1A_N[20]		
AV30	FSB0DRDY_N			AP1	FSB1A_N[21]		
AN27	FSB0FSBVREF[0]			AM1	FSB1A_N[22]		
AN30	FSB0FSBVREF[2]			AR1	FSB1A_N[23]		
AK35	FSB0FSBVREF[4]			AN3	FSB1A_N[24]		
AU35	FSB0HIT_N			AM4	FSB1A_N[25]		
AW35	FSB0HITM_N			AN2	FSB1A_N[26]		
AU31	FSB0LOCK_N			AK5	FSB1A_N[27]		
AL29	FSB0MCERR_N			AL1	FSB1A_N[28]		
AM28	FSB0REQ_N[0]			AJ3	FSB1A_N[29]		
AJ27	FSB0REQ_N[1]			AK9	FSB1A_N[3]		
AL28	FSB0REQ_N[2]			AK2	FSB1A_N[30]		
AJ28	FSB0REQ_N[3]			AK3	FSB1A_N[31]		
AH28	FSB0REQ_N[4]			AJ1	FSB1A_N[32]		
AP31	FSB0RESET_N			AJ4	FSB1A_N[33]		
AW33	FSB0RS_N[0]			AH1	FSB1A_N[34]		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 6 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AH4	FSB1A_N[35]		
AG9	FSB1A_N[36]		
AH8	FSB1A_N[37]		
AM3	FSB1A_N[38]		
AL2	FSB1A_N[39]		
AK6	FSB1A_N[4]		
AK8	FSB1A_N[5]		
AJ9	FSB1A_N[6]		
AL8	FSB1A_N[7]		
AJ7	FSB1A_N[8]		
AH7	FSB1A_N[9]		
AP7	FSB1ADS_N		
AJ6	FSB1ADSTB_N[0]		
AL5	FSB1ADSTB_N[1]		
AU2	FSB1AP_N[0]		
AT3	FSB1AP_N[1]		
AN5	FSB1BINIT_N		
AN8	FSB1BNR_N		
AM7	FSB1BPM_N[4]		
AP8	FSB1BPM_N[5]		
AK15	FSB1BPRI_N		
AM6	FSB1BREQ0_N		
AR6	FSB1BREQ1_N		
AT5	FSB1D_N[0]		
AV3	FSB1D_N[1]		
AW8	FSB1D_N[10]		
AT9	FSB1D_N[11]		
AU7	FSB1D_N[12]		
AW11	FSB1D_N[13]		
AU10	FSB1D_N[14]		
AV9	FSB1D_N[15]		
AK14	FSB1D_N[16]		
AL14	FSB1D_N[17]		
AM12	FSB1D_N[18]		
AM10	FSB1D_N[19]		
AT6	FSB1D_N[2]		
AM13	FSB1D_N[20]		
AN12	FSB1D_N[21]		
AP11	FSB1D_N[22]		
AN11	FSB1D_N[23]		
AR9	FSB1D_N[24]		
AN14	FSB1D_N[25]		
AR13	FSB1D_N[26]		
AT12	FSB1D_N[27]		
AM15	FSB1D_N[28]		
AP13	FSB1D_N[29]		
AU4	FSB1D_N[3]		
AN15	FSB1D_N[30]		
AP14	FSB1D_N[31]		
AU11	FSB1D_N[32]		
AW12	FSB1D_N[33]		
AV12	FSB1D_N[34]		
AU13	FSB1D_N[35]		
AT11	FSB1D_N[36]		
AV13	FSB1D_N[37]		
AT14	FSB1D_N[38]		
AW14	FSB1D_N[39]		
AV4	FSB1D_N[4]		
AT15	FSB1D_N[40]		
AU16	FSB1D_N[41]		
AU17	FSB1D_N[42]		
AR15	FSB1D_N[43]		
AW17	FSB1D_N[44]		
AW18	FSB1D_N[45]		
AV18	FSB1D_N[46]		
AR16	FSB1D_N[47]		
AJ18	FSB1D_N[48]		
AR18	FSB1D_N[49]		
AU5	FSB1D_N[5]		
AP16	FSB1D_N[50]		
AP17	FSB1D_N[51]		
AN17	FSB1D_N[52]		
AT17	FSB1D_N[53]		
AJ16	FSB1D_N[54]		
AT18	FSB1D_N[55]		
AM16	FSB1D_N[56]		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 7 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AL16	FSB1D_N[57]		
AG18	FSB1D_N[58]		
AH16	FSB1D_N[59]		
AW5	FSB1D_N[6]		
AK18	FSB1D_N[60]		
AK17	FSB1D_N[61]		
AG17	FSB1D_N[62]		
AH17	FSB1D_N[63]		
AV6	FSB1D_N[7]		
AW9	FSB1D_N[8]		
AV10	FSB1D_N[9]		
AW6	FSB1DBI_N[0]		
AR12	FSB1DBI_N[1]		
AU14	FSB1DBI_N[2]		
AL17	FSB1DBI_N[3]		
AN9	FSB1DBSY_N		
AL13	FSB1DEFER_N		
AJ15	FSB1DP_N[0]		
AH13	FSB1DP_N[1]		
AH14	FSB1DP_N[2]		
AG14	FSB1DP_N[3]		
AR7	FSB1DRDY_N		
AL4	FSB1FSBVREF[0]		
AV7	FSB1FSBVREF[2]		
AV16	FSB1FSBVREF[4]		
AJ12	FSB1HIT_N		
AK12	FSB1HITM_N		
AM9	FSB1LOCK_N		
AR4	FSB1MCERR_N		
AJ10	FSB1REQ_N[0]		
AH11	FSB1REQ_N[1]		
AL7	FSB1REQ_N[2]		
AH10	FSB1REQ_N[3]		
AG11	FSB1REQ_N[4]		
AG15	FSB1RESET_N		
AL10	FSB1RS_N[0]		
AN6	FSB1RS_N[1]		
AL11	FSB1RS_N[2]		
AP5	FSB1RSP_N		
AT8	FSB1STBN_N[0]		
AR10	FSB1STBN_N[1]		
AW15	FSB1STBN_N[2]		
AN18	FSB1STBN_N[3]		
AU8	FSB1STBP_N[0]		
AP10	FSB1STBP_N[1]		
AV15	FSB1STBP_N[2]		
AM18	FSB1STBP_N[3]		
AK11	FSB1TRDY_N		
AU38	FSBDRVCRES		
AV36	FSBODTCRES		
AV37	FSBSLWCRES		
D21	FSBSLWCTRL		
A14	GPIOSMBCLK		
A13	GPIOSMBDATA		
W4	PE0CLKN		
W5	PE0CLKP		
Y10	PE0ICOMPI		
W10	PE0RCOMPO		
AE11	PE0RN[0]		
AE2	PE0RN[1]		
AG3	PE0RN[2]		
AE5	PE0RN[3]		
AD11	PE0RP[0]		
AD2	PE0RP[1]		
AF3	PE0RP[2]		
AD5	PE0RP[3]		
AE8	PE0TN[0]		
AF1	PE0TN[1]		
AF4	PE0TN[2]		
AF10	PE0TN[3]		
AD8	PE0TP[0]		
AE1	PE0TP[1]		
AE4	PE0TP[2]		
AE10	PE0TP[3]		
H4	PE1CLKN		
G4	PE1CLKP		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 8 of 20)

Ball No.	Signal Name	Buffer Type	Direction
L12	PE1ICOMPI		
L13	PE1RCOMPO		
P7	PE1RN[0]		
P4	PE1RN[1]		
R6	PE1RN[2]		
T11	PE1RN[3]		
N7	PE1RP[0]		
N4	PE1RP[1]		
P6	PE1RP[2]		
R11	PE1RP[3]		
R9	PE1TN[0]		
N1	PE1TN[1]		
R3	PE1TN[2]		
T5	PE1TN[3]		
P9	PE1TP[0]		
N2	PE1TP[1]		
P3	PE1TP[2]		
R5	PE1TP[3]		
T8	PE2RN[0]		
U1	PE2RN[1]		
U4	PE2RN[2]		
V9	PE2RN[3]		
R8	PE2RP[0]		
T1	PE2RP[1]		
T4	PE2RP[2]		
U9	PE2RP[3]		
U10	PE2TN[0]		
T2	PE2TN[1]		
V3	PE2TN[2]		
U7	PE2TN[3]		
T10	PE2TP[0]		
R2	PE2TP[1]		
U3	PE2TP[2]		
T7	PE2TP[3]		
U6	PE3RN[0]		
AA2	PE3RN[1]		
V8	PE3RN[2]		
AB11	PE3RN[3]		
V6	PE3RP[0]		
AA3	PE3RP[1]		
W8	PE3RP[2]		
AA11	PE3RP[3]		
W2	PE3TN[0]		
Y4	PE3TN[1]		
AB8	PE3TN[2]		
W11	PE3TN[3]		
V2	PE3TP[0]		
Y3	PE3TP[1]		
AA8	PE3TP[2]		
V11	PE3TP[3]		
AB5	PE4RN[0]		
AC4	PE4RN[1]		
AD6	PE4RN[2]		
AC10	PE4RN[3]		
AA5	PE4RP[0]		
AB4	PE4RP[1]		
AC6	PE4RP[2]		
AB10	PE4RP[3]		
AC7	PE4TN[0]		
AB1	PE4TN[1]		
AD3	PE4TN[2]		
AD9	PE4TN[3]		
AB7	PE4TP[0]		
AB2	PE4TP[1]		
AC3	PE4TP[2]		
AC9	PE4TP[3]		
C11	PE5RN[0]		
B10	PE5RN[1]		
A7	PE5RN[2]		
A4	PE5RN[3]		
D11	PE5RP[0]		
A10	PE5RP[1]		
A8	PE5RP[2]		
A5	PE5RP[3]		
F11	PE5TN[0]		
D10	PE5TN[1]		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 9 of 20)

Ball No.	Signal Name	Buffer Type	Direction
C9	PE5TN[2]		
B6	PE5TN[3]		
G11	PE5TP[0]		
E10	PE5TP[1]		
B9	PE5TP[2]		
B7	PE5TP[3]		
C8	PE6RN[0]		
E6	PE6RN[1]		
D4	PE6RN[2]		
C2	PE6RN[3]		
D8	PE6RP[0]		
F6	PE6RP[1]		
D5	PE6RP[2]		
C3	PE6RP[3]		
E9	PE6TN[0]		
D7	PE6TN[1]		
C5	PE6TN[2]		
B3	PE6TN[3]		
F9	PE6TP[0]		
E7	PE6TP[1]		
C6	PE6TP[2]		
B4	PE6TP[3]		
J9	PE7RN[0]		
F3	PE7RN[1]		
H1	PE7RN[2]		
H7	PE7RN[3]		
H9	PE7RP[0]		
E3	PE7RP[1]		
G1	PE7RP[2]		
H6	PE7RP[3]		
H10	PE7TN[0]		
D1	PE7TN[1]		
G8	PE7TN[2]		
J3	PE7TN[3]		
G10	PE7TP[0]		
D2	PE7TP[1]		
F8	PE7TP[2]		
H3	PE7TP[3]		
K8	PE8RN[0]		
K4	PE8RN[1]		
L1	PE8RN[2]		
L10	PE8RN[3]		
J8	PE8RP[0]		
K5	PE8RP[1]		
K1	PE8RP[2]		
K10	PE8RP[3]		
K11	PE8TN[0]		
J5	PE8TN[1]		
K2	PE8TN[2]		
L7	PE8TN[3]		
J11	PE8TP[0]		
J6	PE8TP[1]		
J2	PE8TP[2]		
K7	PE8TP[3]		
N10	PE9RN[0]		
M5	PE9RN[1]		
L4	PE9RN[2]		
N8	PE9RN[3]		
P10	PE9RP[0]		
N5	PE9RP[1]		
L3	PE9RP[2]		
M8	PE9RP[3]		
M11	PE9TN[0]		
L9	PE9TN[1]		
M3	PE9TN[2]		
M6	PE9TN[3]		
N11	PE9TP[0]		
M9	PE9TP[1]		
M2	PE9TP[2]		
L6	PE9TP[3]		
J14	PLLPWRDET		
K20	PSEL[0]		
G20	PSEL[1]		
F20	PSEL[2]		
B13	PWRGOOD		
AU37	RESERVED[0]		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 10 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AL23	RESERVED[1]		
T28	RESERVED[11]		
M33	RESERVED[12]		
L33	RESERVED[13]		
F32	RESERVED[24]		
E21	RESERVED[27]		
P1	RESERVED[41]		
AC1	RESERVED[42]		
U36	RESERVED[44]		
H25	RESERVED[45]		
U37	RESERVED[46]		
H24	RESERVED[47]		
Y9	RESERVED[48]		
Y7	RESERVED[49]		
W7	RESERVED[50]		
AA9	RESERVED[51]		
C13	RESETI_N		
K14	SPD0SMBCLK		
K13	SPD0SMBDATA		
G15	SPD1SMBCLK		
H15	SPD1SMBDATA		
G13	SPD2SMBCLK		
H13	SPD2SMBDATA		
E13	SPD3SMBCLK		
F13	SPD3SMBDATA		
D20	TCK		
C19	TDI		
B20	TDO		
A23	TESTLO[1]		
G21	TESTLO[10]		
E22	TESTLO[11]		
A22	TESTLO[2]		
H33	TESTLO[3]		
H34	TESTLO[4]		
L36	TESTLO[5]		
M36	TESTLO[6]		
M35	TESTLO[7]		
D19	TESTLO[8]		
F19	TESTLO[9]		
D15	THERMALERT_N		
L14	THERMTRIP_N		
A19	TMS		
A20	TRSTNN		
M17	VCC		
M18	VCC		
N17	VCC		
N19	VCC		
P18	VCC		
P20	VCC		
R17	VCC		
R19	VCC		
R21	VCC		
R23	VCC		
R25	VCC		
T16	VCC		
T18	VCC		
T20	VCC		
T22	VCC		
T24	VCC		
U17	VCC		
U19	VCC		
U21	VCC		
U23	VCC		
U25	VCC		
V16	VCC		
V18	VCC		
V20	VCC		
V22	VCC		
V24	VCC		
W17	VCC		
W19	VCC		
W21	VCC		
W23	VCC		
W25	VCC		
Y16	VCC		
Y18	VCC		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 11 of 20)

Ball No.	Signal Name	Buffer Type	Direction
Y20	VCC		
Y22	VCC		
Y24	VCC		
AA17	VCC		
AA19	VCC		
AA21	VCC		
AA23	VCC		
AA25	VCC		
AB16	VCC		
AB18	VCC		
AB20	VCC		
AB22	VCC		
AB24	VCC		
AC17	VCC		
AC19	VCC		
AC21	VCC		
AC23	VCC		
AC25	VCC		
AD16	VCC		
AD18	VCC		
AD20	VCC		
AD22	VCC		
AD24	VCC		
AH23	VCC		
AH25	VCC		
AJ13	VCC		
AA6	VCCABGPE033		
G5	VCCABGPE133		
AU23	VCCACPLL		
M20	VCCAFBD		
M21	VCCAFBD		
M22	VCCAFBD		
M23	VCCAFBD		
M24	VCCAFBD		
M25	VCCAFBD		
M26	VCCAFBD		
M27	VCCAFBD		
N20	VCCAFBD		
N21	VCCAFBD		
N22	VCCAFBD		
N23	VCCAFBD		
N24	VCCAFBD		
N25	VCCAFBD		
N26	VCCAFBD		
N27	VCCAFBD		
P22	VCCAFBD		
P23	VCCAFBD		
P24	VCCAFBD		
P25	VCCAFBD		
P26	VCCAFBD		
P27	VCCAFBD		
P28	VCCAFBD		
R26	VCCAFBD		
R27	VCCAFBD		
T26	VCCAFBD		
T27	VCCAFBD		
U26	VCCAFBD		
U27	VCCAFBD		
U28	VCCAFBD		
V26	VCCAFBD		
V27	VCCAFBD		
W26	VCCAFBD		
W27	VCCAFBD		
Y26	VCCAFBD		
Y27	VCCAFBD		
Y28	VCCAFBD		
AA26	VCCAFBD		
AA27	VCCAFBD		
AB26	VCCAFBD		
AB27	VCCAFBD		
AC26	VCCAFBD		
AC27	VCCAFBD		
AD26	VCCAFBD		
AD27	VCCAFBD		
P39	VCCAFBD01PLL		
T34	VCCAFBD01PLL18		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 12 of 20)

Ball No.	Signal Name	Buffer Type	Direction
C27	VCCAFBD23PLL		
F33	VCCAFBD23PLL18		
W1	VCCAPE0PLL125		
V5	VCCAPE0PLL18		
M12	VCCAPE125		
M13	VCCAPE125		
N12	VCCAPE125		
N13	VCCAPE125		
N14	VCCAPE125		
P12	VCCAPE125		
P13	VCCAPE125		
P14	VCCAPE125		
R12	VCCAPE125		
R13	VCCAPE125		
R14	VCCAPE125		
T13	VCCAPE125		
T14	VCCAPE125		
U12	VCCAPE125		
U13	VCCAPE125		
U14	VCCAPE125		
V12	VCCAPE125		
V13	VCCAPE125		
V14	VCCAPE125		
W12	VCCAPE125		
W13	VCCAPE125		
W14	VCCAPE125		
Y12	VCCAPE125		
Y13	VCCAPE125		
Y14	VCCAPE125		
AA12	VCCAPE125		
AA13	VCCAPE125		
AA14	VCCAPE125		
AB13	VCCAPE125		
AB14	VCCAPE125		
AC12	VCCAPE125		
AC13	VCCAPE125		
AD12	VCCAPE125		
AD13	VCCAPE125		
F2	VCCAPE1PLL125		
E1	VCCAPE1PLL18		
AT23	VCCDCPLL		
E20	VCCMISC125		
G19	VCCMISC125		
G34	VCCMISC125		
J18	VCCMISC125		
F14	VCCMISC33		
H14	VCCMISC33		
H16	VCCMISC33		
J15	VCCMISC33		
L15	VCCRPE125		
M15	VCCRPE125		
N15	VCCRPE125		
N16	VCCRPE125		
P16	VCCRPE125		
R15	VCCRPE125		
T15	VCCRPE125		
U15	VCCRPE125		
W15	VCCRPE125		
Y15	VCCRPE125		
AA15	VCCRPE125		
AC14	VCCRPE125		
AC15	VCCRPE125		
AD14	VCCRPE125		
AD15	VCCRPE125		
L27	VCCSEN		
AE13	VCCSF		
AE14	VCCSF		
AE26	VCCSF		
AE28	VCCSF		
AF12	VCCSF		
AF13	VCCSF		
AF14	VCCSF		
AF26	VCCSF		
AF27	VCCSF		
AF28	VCCSF		
AG12	VCCSF		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 13 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
AG27	VCCSF			B35	VSS		
J12	VID[1]			B38	VSS		
H12	VID[2]			B39	VSS		
G12	VID[3]			C1	VSS		
E12	VID[4]			C4	VSS		
D12	VID[5]			C7	VSS		
C12	VID[6]			C10	VSS		
J19	VSCALEN			C15	VSS		
A2	VSS			C16	VSS		
A3	VSS			C20	VSS		
A6	VSS			C22	VSS		
A9	VSS			C25	VSS		
A11	VSS			C28	VSS		
A12	VSS			C31	VSS		
A17	VSS			C34	VSS		
A21	VSS			C37	VSS		
A24	VSS			C39	VSS		
A27	VSS			D3	VSS		
A30	VSS			D6	VSS		
A33	VSS			D9	VSS		
A36	VSS			D13	VSS		
A37	VSS			D18	VSS		
A38	VSS			D24	VSS		
A39	VSS			D27	VSS		
B1	VSS			D30	VSS		
B2	VSS			D33	VSS		
B5	VSS			D36	VSS		
B8	VSS			D39	VSS		
B11	VSS			E2	VSS		
B12	VSS			E4	VSS		
B14	VSS			E5	VSS		
B18	VSS			E8	VSS		
B21	VSS			E11	VSS		
B22	VSS			E15	VSS		
B23	VSS			E17	VSS		
B26	VSS			E19	VSS		
B29	VSS			E23	VSS		
B32	VSS			E26	VSS		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 14 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
E29	VSS			H29	VSS		
E32	VSS			H31	VSS		
E35	VSS			H32	VSS		
E38	VSS			H35	VSS		
F1	VSS			H38	VSS		
F4	VSS			J1	VSS		
F7	VSS			J4	VSS		
F10	VSS			J7	VSS		
F12	VSS			J10	VSS		
F21	VSS			J13	VSS		
F22	VSS			J16	VSS		
F25	VSS			J17	VSS		
F28	VSS			J20	VSS		
F31	VSS			J21	VSS		
F34	VSS			J22	VSS		
F37	VSS			J25	VSS		
G3	VSS			J28	VSS		
G6	VSS			J31	VSS		
G7	VSS			J34	VSS		
G9	VSS			J37	VSS		
G14	VSS			K3	VSS		
G16	VSS			K6	VSS		
G18	VSS			K9	VSS		
G24	VSS			K12	VSS		
G27	VSS			K15	VSS		
G30	VSS			K16	VSS		
G33	VSS			K19	VSS		
G36	VSS			K21	VSS		
G39	VSS			K24	VSS		
H2	VSS			K27	VSS		
H5	VSS			K30	VSS		
H8	VSS			K33	VSS		
H11	VSS			K36	VSS		
H19	VSS			K39	VSS		
H20	VSS			L2	VSS		
H21	VSS			L5	VSS		
H23	VSS			L8	VSS		
H26	VSS			L11	VSS		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 15 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
L16	VSS			P35	VSS		
L17	VSS			P38	VSS		
L20	VSS			R1	VSS		
L23	VSS			R4	VSS		
L26	VSS			R7	VSS		
L29	VSS			R10	VSS		
L32	VSS			R16	VSS		
L35	VSS			R18	VSS		
L38	VSS			R20	VSS		
M1	VSS			R22	VSS		
M4	VSS			R24	VSS		
M7	VSS			R28	VSS		
M10	VSS			R31	VSS		
M14	VSS			R34	VSS		
M16	VSS			R37	VSS		
M19	VSS			T3	VSS		
M28	VSS			T6	VSS		
M31	VSS			T9	VSS		
M34	VSS			T12	VSS		
M37	VSS			T17	VSS		
N3	VSS			T19	VSS		
N6	VSS			T21	VSS		
N9	VSS			T23	VSS		
N18	VSS			T25	VSS		
N30	VSS			T30	VSS		
N33	VSS			T33	VSS		
N36	VSS			T36	VSS		
N39	VSS			T39	VSS		
P2	VSS			U2	VSS		
P5	VSS			U5	VSS		
P8	VSS			U8	VSS		
P11	VSS			U11	VSS		
P15	VSS			U16	VSS		
P17	VSS			U18	VSS		
P19	VSS			U20	VSS		
P21	VSS			U22	VSS		
P29	VSS			U24	VSS		
P32	VSS			U29	VSS		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 16 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
U32	VSS			Y29	VSS		
U35	VSS			Y32	VSS		
U38	VSS			Y35	VSS		
V1	VSS			Y36	VSS		
V4	VSS			Y38	VSS		
V7	VSS			Y39	VSS		
V10	VSS			AA1	VSS		
V15	VSS			AA4	VSS		
V17	VSS			AA7	VSS		
V19	VSS			AA10	VSS		
V21	VSS			AA16	VSS		
V23	VSS			AA18	VSS		
V25	VSS			AA20	VSS		
V28	VSS			AA22	VSS		
V31	VSS			AA24	VSS		
V34	VSS			AA28	VSS		
V37	VSS			AA30	VSS		
W3	VSS			AA31	VSS		
W6	VSS			AA34	VSS		
W9	VSS			AA37	VSS		
W16	VSS			AB3	VSS		
W18	VSS			AB6	VSS		
W20	VSS			AB9	VSS		
W22	VSS			AB12	VSS		
W24	VSS			AB15	VSS		
W30	VSS			AB17	VSS		
W33	VSS			AB19	VSS		
W36	VSS			AB21	VSS		
W39	VSS			AB23	VSS		
Y2	VSS			AB25	VSS		
Y5	VSS			AB30	VSS		
Y8	VSS			AB33	VSS		
Y11	VSS			AB36	VSS		
Y17	VSS			AB39	VSS		
Y19	VSS			AC2	VSS		
Y21	VSS			AC5	VSS		
Y23	VSS			AC8	VSS		
Y25	VSS			AC11	VSS		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 17 of 20)

Ball No.	Signal Name	Buffer Type	Direction	Ball No.	Signal Name	Buffer Type	Direction
AC16	VSS			AF32	VSS		
AC18	VSS			AF35	VSS		
AC20	VSS			AF38	VSS		
AC22	VSS			AG1	VSS		
AC24	VSS			AG2	VSS		
AC29	VSS			AG4	VSS		
AC32	VSS			AG7	VSS		
AC35	VSS			AG10	VSS		
AC38	VSS			AG13	VSS		
AC39	VSS			AG16	VSS		
AD1	VSS			AG25	VSS		
AD4	VSS			AG28	VSS		
AD7	VSS			AG31	VSS		
AD10	VSS			AG34	VSS		
AD17	VSS			AG37	VSS		
AD19	VSS			AH2	VSS		
AD21	VSS			AH3	VSS		
AD23	VSS			AH6	VSS		
AD25	VSS			AH9	VSS		
AD28	VSS			AH12	VSS		
AD31	VSS			AH15	VSS		
AD34	VSS			AH18	VSS		
AD37	VSS			AH24	VSS		
AE3	VSS			AH27	VSS		
AE6	VSS			AH30	VSS		
AE7	VSS			AH33	VSS		
AE9	VSS			AH36	VSS		
AE12	VSS			AH39	VSS		
AE27	VSS			AJ2	VSS		
AE30	VSS			AJ5	VSS		
AE33	VSS			AJ8	VSS		
AE36	VSS			AJ11	VSS		
AE39	VSS			AJ14	VSS		
AF2	VSS			AJ17	VSS		
AF5	VSS			AJ23	VSS		
AF8	VSS			AJ26	VSS		
AF11	VSS			AJ29	VSS		
AF29	VSS			AJ32	VSS		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 18 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AJ35	VSS		
AJ38	VSS		
AK1	VSS		
AK4	VSS		
AK7	VSS		
AK10	VSS		
AK13	VSS		
AK16	VSS		
AK23	VSS		
AK25	VSS		
AK28	VSS		
AK31	VSS		
AK34	VSS		
AK37	VSS		
AL3	VSS		
AL6	VSS		
AL9	VSS		
AL12	VSS		
AL15	VSS		
AL18	VSS		
AL24	VSS		
AL27	VSS		
AL30	VSS		
AL33	VSS		
AL36	VSS		
AL39	VSS		
AM2	VSS		
AM5	VSS		
AM8	VSS		
AM11	VSS		
AM14	VSS		
AM17	VSS		
AM23	VSS		
AM26	VSS		
AM29	VSS		
AM32	VSS		
AM35	VSS		
AM38	VSS		
AN1	VSS		
AN4	VSS		
AN7	VSS		
AN10	VSS		
AN13	VSS		
AN16	VSS		
AN25	VSS		
AN28	VSS		
AN31	VSS		
AN34	VSS		
AN37	VSS		
AP3	VSS		
AP6	VSS		
AP9	VSS		
AP12	VSS		
AP15	VSS		
AP18	VSS		
AP24	VSS		
AP27	VSS		
AP30	VSS		
AP33	VSS		
AP36	VSS		
AP39	VSS		
AR2	VSS		
AR5	VSS		
AR8	VSS		
AR11	VSS		
AR14	VSS		
AR17	VSS		
AR23	VSS		
AR26	VSS		
AR29	VSS		
AR32	VSS		
AR35	VSS		
AR38	VSS		
AT1	VSS		
AT4	VSS		
AT7	VSS		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 19 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AT10	VSS		
AT13	VSS		
AT16	VSS		
AT25	VSS		
AT28	VSS		
AT31	VSS		
AT34	VSS		
AT37	VSS		
AU1	VSS		
AU3	VSS		
AU6	VSS		
AU9	VSS		
AU12	VSS		
AU15	VSS		
AU18	VSS		
AU24	VSS		
AU27	VSS		
AU30	VSS		
AU33	VSS		
AU36	VSS		
AU39	VSS		
AV1	VSS		
AV2	VSS		
AV5	VSS		
AV8	VSS		
AV11	VSS		
AV14	VSS		
AV17	VSS		
AV23	VSS		
AV26	VSS		
AV29	VSS		
AV32	VSS		
AV35	VSS		
AV38	VSS		
AV39	VSS		
AW1	VSS		
AW2	VSS		
AW3	VSS		
AW4	VSS		
AW7	VSS		
AW10	VSS		
AW13	VSS		
AW16	VSS		
AW25	VSS		
AW28	VSS		
AW31	VSS		
AW34	VSS		
AW37	VSS		
AW38	VSS		
AW39	VSS		
Y6	VSSABGPE0		
F5	VSSABGPE1		
Y1	VSSAPE0PLL		
G2	VSSAPE1PLL		
L28	VSSSEN		
AE15	VTT		
AE16	VTT		
AE17	VTT		
AE18	VTT		
AE19	VTT		
AE20	VTT		
AE21	VTT		
AE22	VTT		
AE23	VTT		
AE24	VTT		
AE25	VTT		
AF15	VTT		
AF16	VTT		
AF17	VTT		
AF18	VTT		
AF19	VTT		
AF20	VTT		
AF21	VTT		
AF22	VTT		
AF23	VTT		
AF24	VTT		

Table 8-2. Intel 5400 Chipset MCH Signals By Signal Name (Sheet 20 of 20)

Ball No.	Signal Name	Buffer Type	Direction
AF25	VTT		
AG19	VTT		
AG20	VTT		
AG21	VTT		
AG22	VTT		
AH19	VTT		
AH20	VTT		
AH21	VTT		
AH22	VTT		
AJ19	VTT		
AJ20	VTT		
AJ21	VTT		
AJ22	VTT		
AK19	VTT		
AK20	VTT		
AK21	VTT		
AK22	VTT		
AL19	VTT		
AL20	VTT		
AL21	VTT		
AL22	VTT		
AM19	VTT		
AM20	VTT		
AM21	VTT		
AM22	VTT		
AN19	VTT		
AN20	VTT		
AN21	VTT		
AN22	VTT		
AP19	VTT		
AP20	VTT		
AP21	VTT		
AP22	VTT		
AR19	VTT		
AR20	VTT		
AR21	VTT		
AR22	VTT		
AT19	VTT		
AT20	VTT		
AT21	VTT		
AT22	VTT		
AU19	VTT		
AU20	VTT		
AU21	VTT		
AU22	VTT		
AV19	VTT		
AV20	VTT		
AV21	VTT		
AV22	VTT		
AW19	VTT		
AW20	VTT		
AW21	VTT		
AW22	VTT		
K18	XDPCLK1X		
F17	XDPDATA_N[0]		
K17	XDPDATA_N[1]		
A16	XDPDATA_N[10]		
C18	XDPDATA_N[11]		
L19	XDPDATA_N[12]		
B15	XDPDATA_N[13]		
H18	XDPDATA_N[14]		
A18	XDPDATA_N[15]		
H17	XDPDATA_N[2]		
E16	XDPDATA_N[3]		
F16	XDPDATA_N[4]		
F18	XDPDATA_N[5]		
D16	XDPDATA_N[6]		
D17	XDPDATA_N[7]		
E18	XDPDATA_N[8]		
B16	XDPDATA_N[9]		
C17	XDPDSTBN_N		
B17	XDPDSTBP_N		
B19	XDPRDYACK_N		
L18	XDPRDYREQ_N		
G17	XDPVREF		
C21	XOROUT		

8.2 Package Information

Figure 8-5. Bottom View

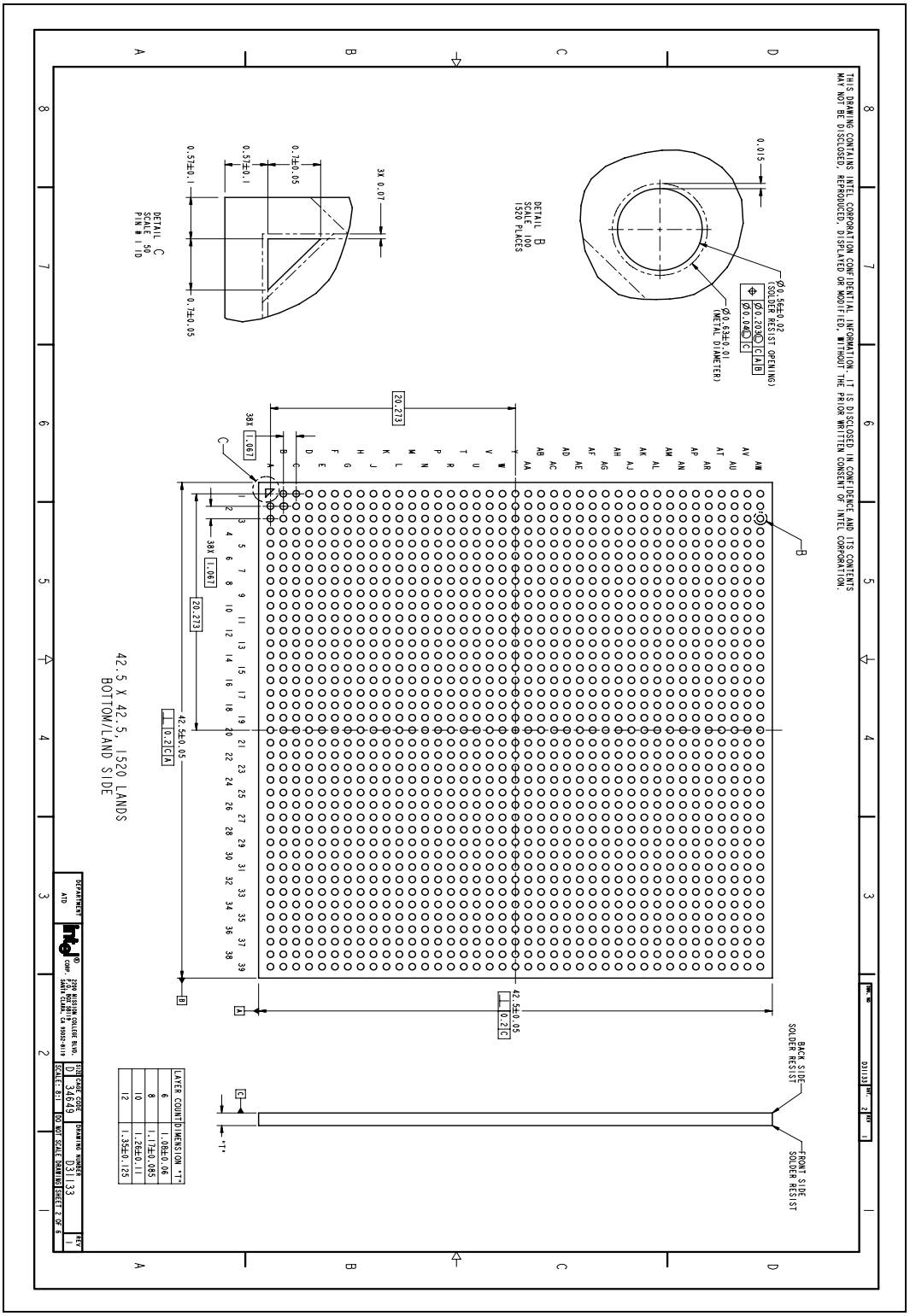


Figure 8-6. Top View

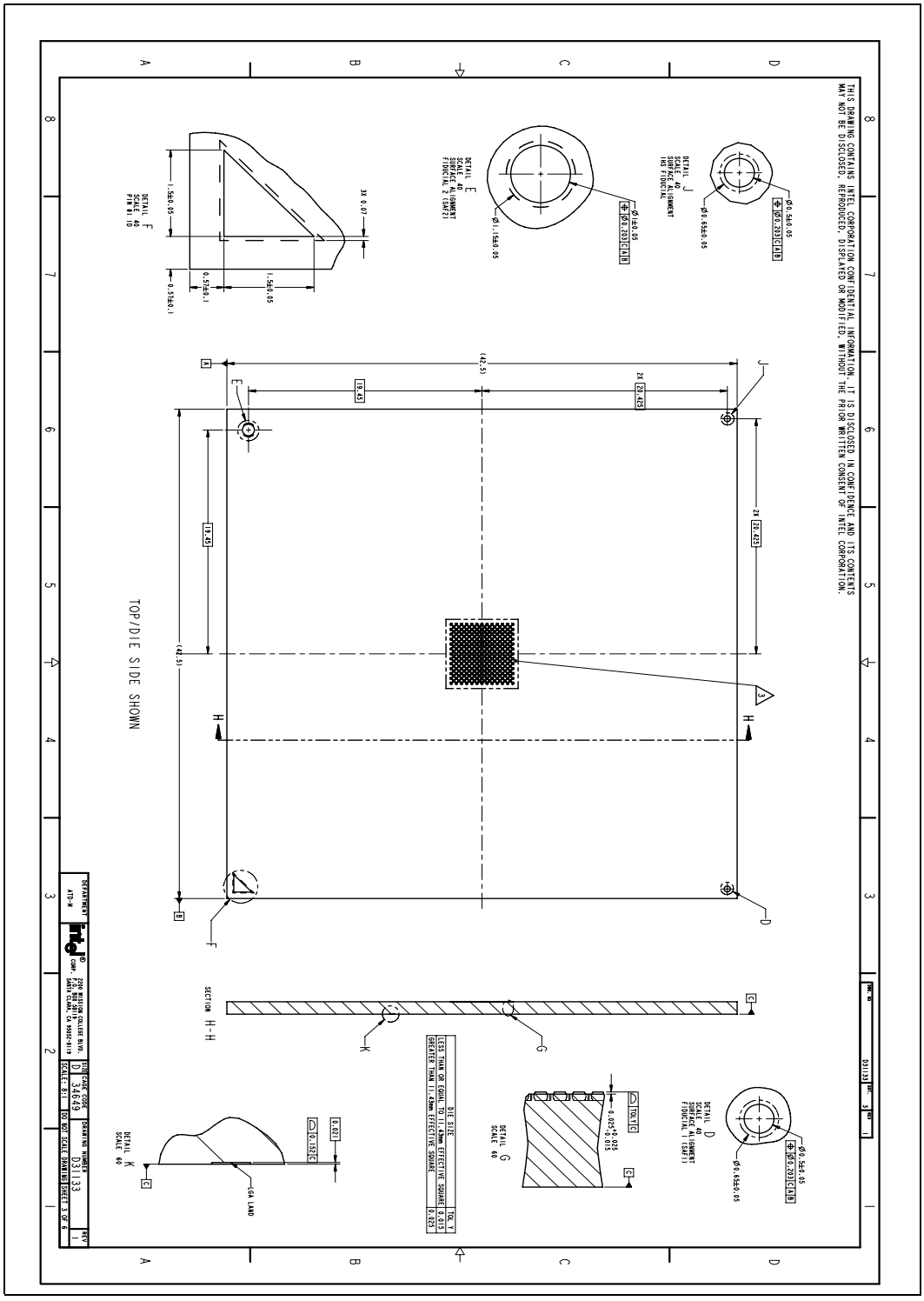


Figure 8-7. Stackup

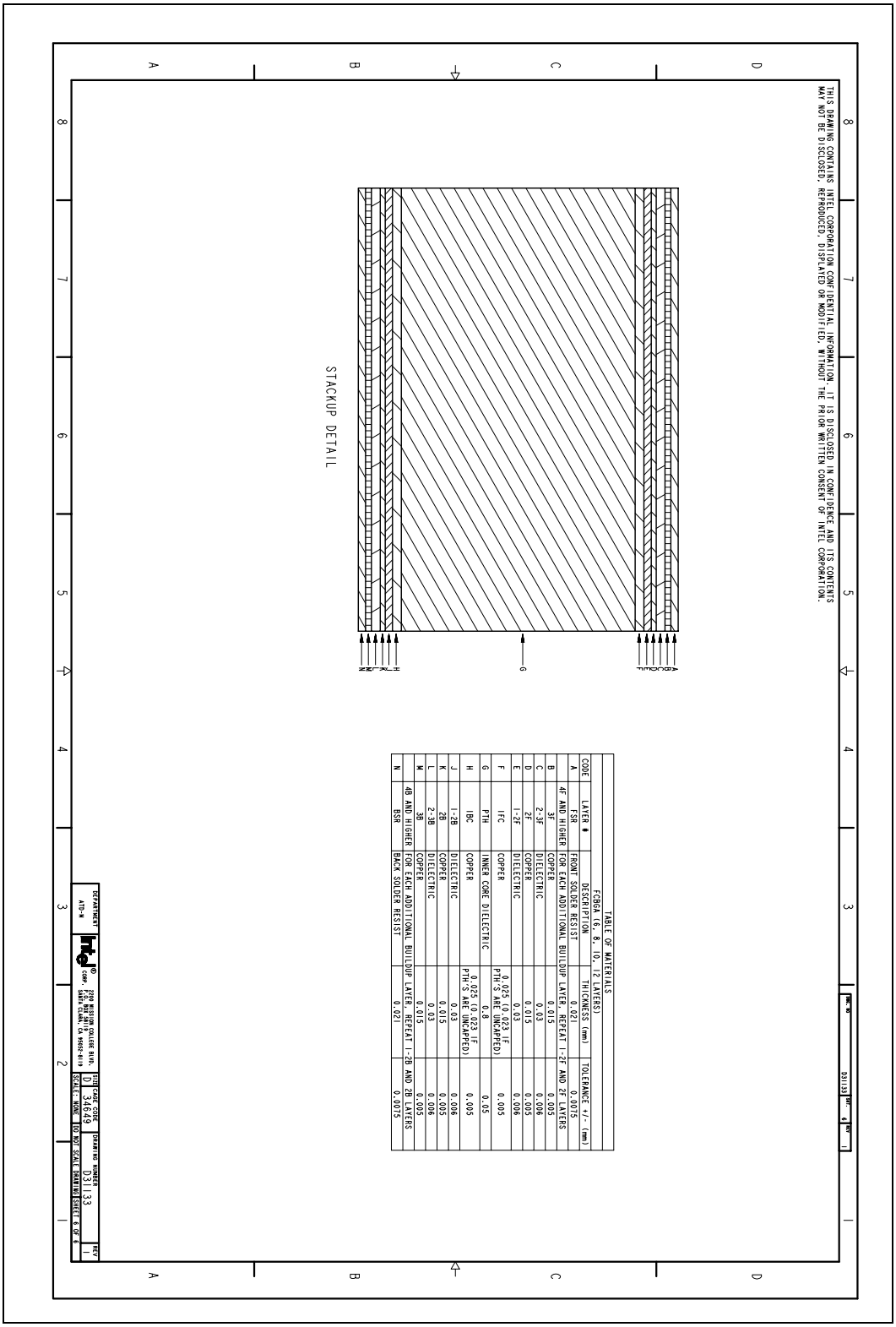


Figure 8-8. Surface Handling Exclusion Zone

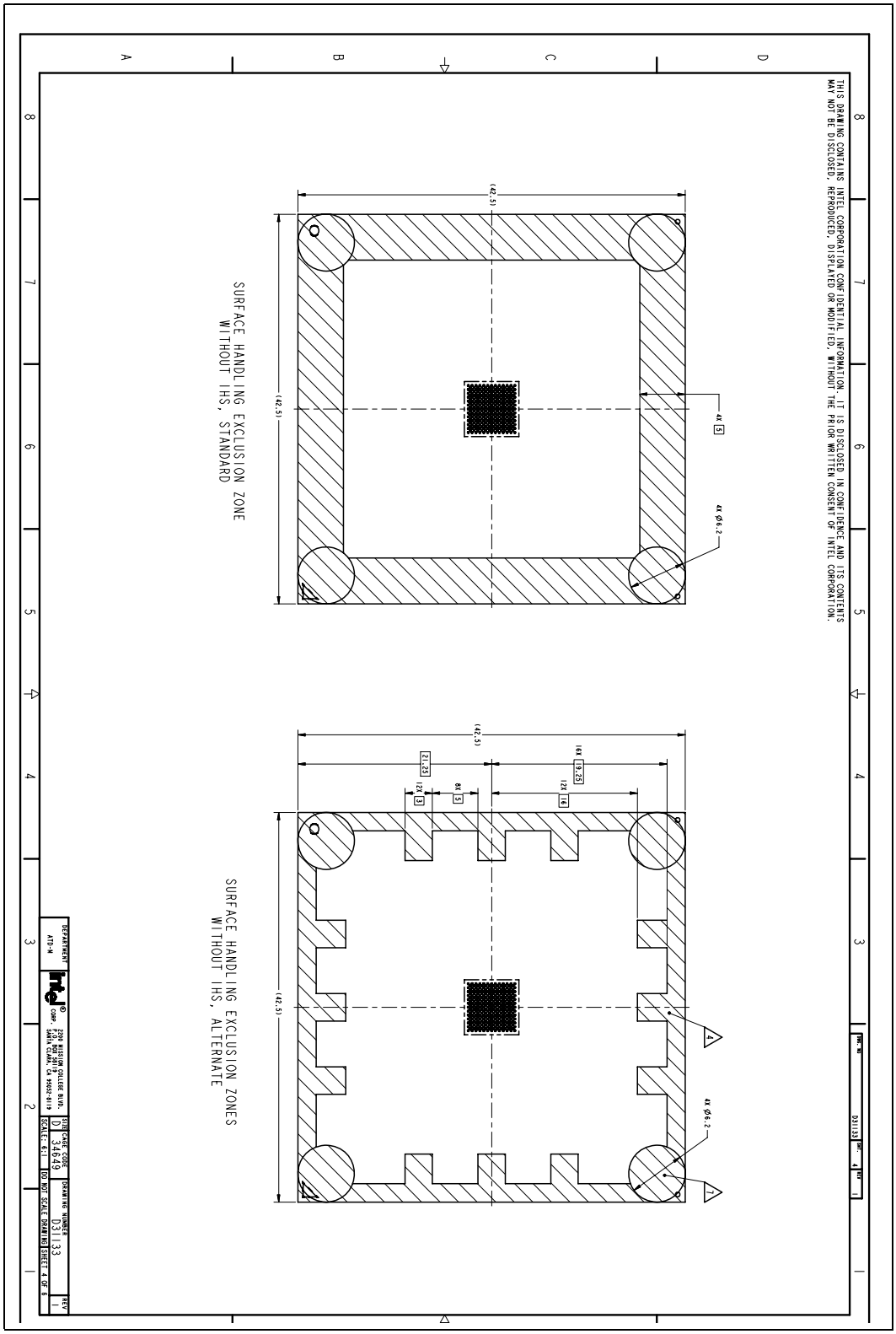
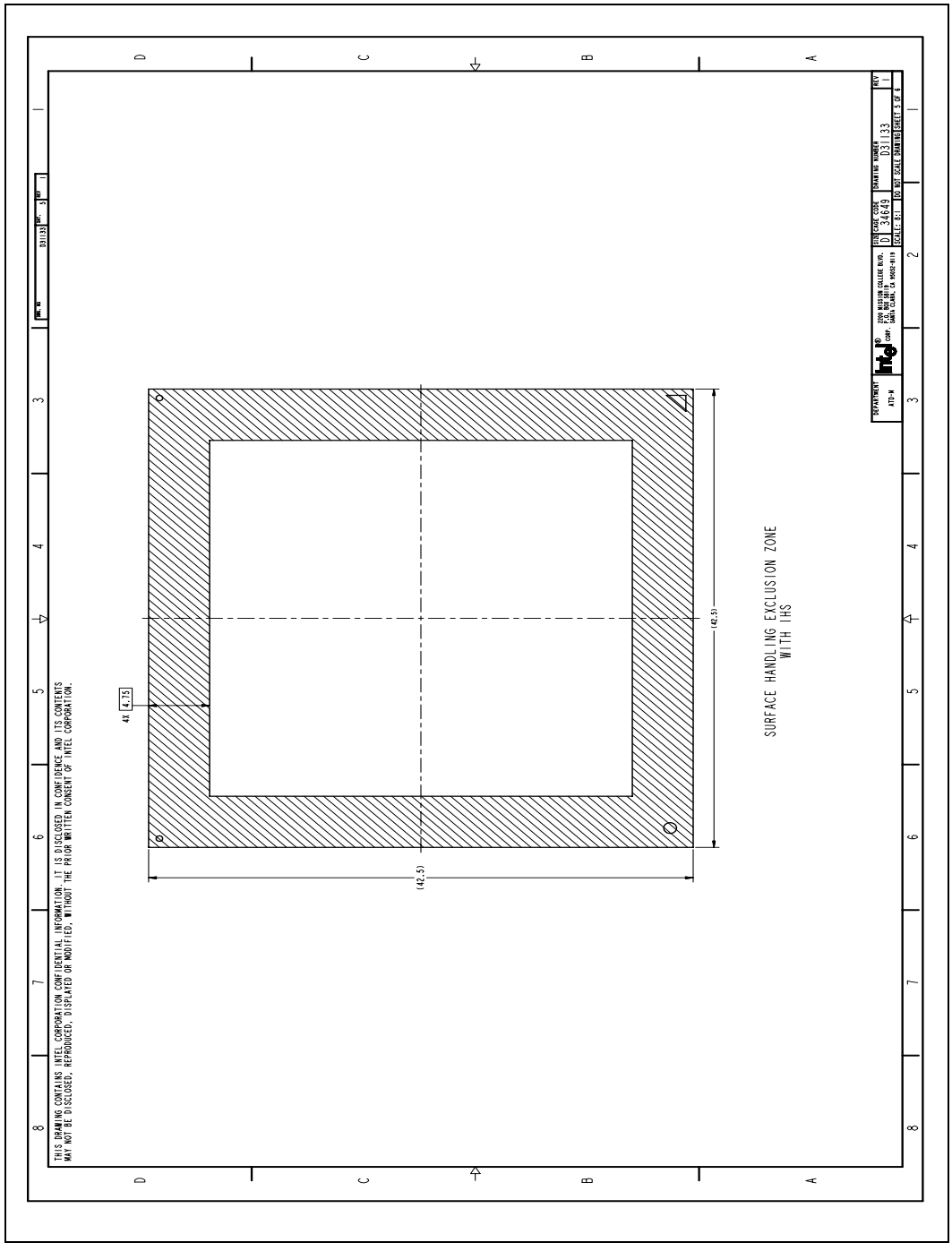


Figure 8-9. Surface Handle Exclusion Zone with IHS



§