# Intel® 860 Chipset: 82860 Memory Controller Hub (MCH)

## Datasheet

*May 2001*

Document Number: 290713-001

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

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

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

The Intel® 82860 Memory Controller Hub may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

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

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

Copies of documents which have an ordering number and are referenced in this document, or other Intel literature, may be obtained by calling

1-800-548-4725 or by visiting Intel's website at http://www.intel.com.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

# Contents

# Figures

## Tables

# Revision History

| Revision Number | Description | Revision Date |
|---|---|---|
| -001 | Initial Release. | May 2001 |

# Intel® 82860 MCH Features

- Supports Intel® Xeon™ processors
  - 100 MHz system bus
  - 200 MHz address bus
  - 400 MHz data bus
  - System bus interrupt delivery
  - AGTL+ bus driver technology with integrated AGTL termination resistors
- Direct RDRAM* Device Support
  - Two Rambus* Channels operating in lock-step at 300 MHz and 400 MHz
  - Maximum memory bandwidth of 3.2 GB/s
  - 128-/144-Mb (1-KB page size) and 256-/288-Mb (2-KB page size) Direct RDRAM device densities
  - Supports a maximum memory address decode space of 16 GB
  - Maximum memory
    - -1 GB using 128-/144-Mb Direct RDRAM devices
    - -2 GB using 256-/288-Mb Direct RDRAM devices
    - -4 GB using two Intel® 82803AA MRH-Rs
  - Up to 8 simultaneous open pages
  - Direct RDRAM device subsystem thermal management
  - ECC operation (single-bit error correction and multiple-bit error detection)
- Hub Interface_A to ICH2
  - 266 MB/s point-to-point hub interface to ICH2 with parity
  - Interrupt related messages
  - Power management events as messages
  - SMI, SCI, and SERR error indication messages
  - Supports normal and enhanced termination modes
- Hub Interface_B and Hub Interface_C
  - 533 MB/s point-to-point 16-bit hub interfaces with parity
  - 66 MHz base clock running 4x (533MB/s) data transfers
  - 36-bit addressing on inbound transactions only (maximum 16-GB memory decode space)
- Accelerated Graphics Port (AGP) Interface
  - Supports a single AGP device (either via a connector or on the motherboard)
  - Supports AGP 2.0 including 4x AGP data transfers and 2x/4x fast write protocol
  - 1.5 V AGP signaling levels
  - 32 deep AGP request queue
  - AGP address translation mechanism with integrated fully associative 20 entry TLB
  - Delayed transaction support for AGP-to-DRAM FRAME# semantic reads
- System Interrupts
  - Supports only system bus interrupt delivery mechanism
  - Supports interrupts signaled as upstream Memory Writes from hub interface_A–C
  - Supports peer MSI between hub interface_A–C
  - Provides redirection for IPI and upstream interrupts to the system bus
- Power Management
  - SMRAM space remapping to A0000h
  - Supports extended SMRAM space above 256 MB, additional TSEG from Top of Low Memory
  - SMRAM accesses from AGP or hub interfaces are not allowed
  - PC99/2001 suspend to DRAM support
  - ACPI Rev 1.0 compliant power management
  - APM Rev 1.2 compliant power management
- Package
  - 42.5 x 42.5 mm 1012OLGA
- I/O Device Support
  - ICH2
  - Intel® P64H (16-bit hub interface-to-optional 64-bit/66 MHz PCI Bus Hub)

This page is intentionally left blank.

# 1 Introduction

The Intel® 860 chipset is a high-bandwidth chipset designed for workstation platforms based on the Intel® Xeon™ processor. The chipset contains two main components and additional optional components that provide expansion capability. The Intel® 82860 Memory Controller Hub (MCH) provides the chipset's system bus interface, memory controller, AGP interface, hub interface for I/O, and two hub interface ports for PCI bus expansion. This document describes the Intel 82860 Memory Controller Hub (MCH). Section 1.3, *Intel® 860 Chipset System Architecture*, provides an overview of each of the components of the Intel 860 chipset.

## 1.1 Terminology

| Term | Description |
|---|---|
| AGP | Accelerated Graphics Port. The MCH contains an AGP that supports AGP 2.0 compliant components only with 1.5 V signaling level. PIPE# and SBA addressing cycles and their associated data phases are generally referred to as AGP transactions. FRAME# cycles over the AGP bus are generally referred to as AGP/PCI transactions. |
| Core | The internal base logic in the MCH. |
| DBI | Dynamic Bus Inversion |
| DP | Dual-Processor |
| Full Reset | A Full MCH Reset is defined in this document when RSTIN# is asserted. |
| GART | Graphics Aperture Re-map Table. Table in memory containing the page re-map information used during AGP aperture address translations. |
| GTLB | Graphics Translation Look-aside Buffer. A cache used to store frequently used GART entries. |
| Host | This term is used synonymously with processor. |
| Hub Interface_A | The proprietary hub interface that ties the MCH to the ICH2. In this document hub interface cycles originating from or destined for the primary PCI interface on the ICH2 is generally referred to as Hub Interface_A cycles. |
| Hub Interface_B and Hub Interface_C | The proprietary hub interface that ties the MCH to the Intel P64H. Cycles originating from or destined for any target on one of these hub interfaces are described as Hub Interface_B and Hub Interface_C cycles respectively. |
| ICH2 | Intel® 82801BA I/O Controller Hub (ICH2). The I/O Controller Hub component that contains the primary PCI interface, LPC interface, USB, ATA-100, AC'97, and other I/O functions. It communicates with the MCH over a proprietary interface called Hub Interface_A. |
| IPI | Inter Processor Interrupt |
| MCH | The Memory Controller Hub component that contains the processor interface, DRAM controller, and AGP interface. It communicates with the I/O controller hub (ICH2) and other I/O controller hubs over proprietary interfaces called the hub interface. |

| Term | Description |
|---|---|
| Intel® MRH-R | The Memory Repeater Hub (for Direct RDRAM* devices) component that allows the system to expand the number of available Rambus* Channels. Each Intel MRH-R connects one primary Rambus Channel to two subordinate Rambus Channels**.** |
| MSI | Message Signaled Interrupts. MSI allows a device to request interrupt service via a standard Memory Write transaction instead of through a hardware signal. |
| Normal vs Enhanced Mode | The normal routing distance for a hub interface is a few inches. Enhanced mode allows the user the flexibility to have the hub interface farther apart. |
| Intel® P64H | The Bus Controller Hub component that has a 16-bit hub interface on its primary side and a configurable 64-bit, 66 MHz PCI interface on the secondary side. It connects to any one of the Intel 82860 MCH's 16-bit hub interfaces. |
| PCI_A | The physical PCI bus that is driven directly by the ICH2 component. It supports 5 V, 32-bit, 33 MHz PCI 2.2 compliant components. Communication between PCI_A and the MCH occurs over Hub Interface_A.<br>**Note:** Even though it is referred to as PCI_A it is not PCI Bus #0 from a configuration standpoint. |
| RAC | Rambus* ASIC Cell. The RAC is a library macrocell used in ASIC controller designs to interface the core logic of a CMOS ASIC device to the Rambus Channel. It is the embedded cell designed by Rambus that interfaces with the Direct RDRAM* devices using RSL signaling. The RAC communicates with the RMC. |
| RMC | Rambus* Memory Controller. The RMC is a block of digital logic residing on a Rambus-based controller IC to drive and manage the memory transactions of a Rambus memory system. This is the logic that directly interfaces to the RAC. |
| RSL | Rambus Signaling Level. RSL is a multi-drop, bidirectional bus connection signaling technology. Operating up to a GHz transfer rate, RSL uses low swing signaling, a common reference voltage and precise clocking to transfer two bits per clock cycle. |
| Rambus* Channel | The Rambus Channel consists of a two-byte wide data path capable of transferring data and address information at rates of 800MHz and beyond. The Rambus Channel has defined mechanical and electrical interfaces and consists of a memory controller, RDRAM devices, DRCG and all interconnect components. |
| System Bus | Processor-to-MCH interface. The system bus runs at 400 MHz from a 100 MHz quad-pumped clock. It includes source synchronous transfers for address and data, and system bus interrupt delivery. |
| UP | Uni-Processor |

## 1.2 Reference Documents

| Document | Document Number |
|---|---|
| *Intel® Xeon™ Processor and Intel® 860 Chipset Platform Design Guide* | 298252 |
| *Intel® 82801BA I/O Controller Hub* (ICH2) *and Intel® 82801BAM I/O Controller Hub* (ICH2-M) *Datasheet* | 290687 |
| *Intel® 860 Chipset Thermal Considerations Application Note (AP-721)* | 292269 |
| *Intel® 82806AA PCI 64 Hub (P64H)* Datasheet | 298025 |
| *Intel® 82803AA Memory Repeater Hub for RDRAM (MRH-R) Datasheet* | 298022 |
| *Intel® 82802AB/AC Firmware Hub (FWH) Datasheet* | 290658 |
| *Intel® Xeon™ Processor Datasheet* | |

***Note:*** See the *Intel® Xeon™ Processor and Intel® 860 Chipset Platform Design Guide* for an expanded set of related documents.

## 1.3 Intel® 860 Chipset System Architecture

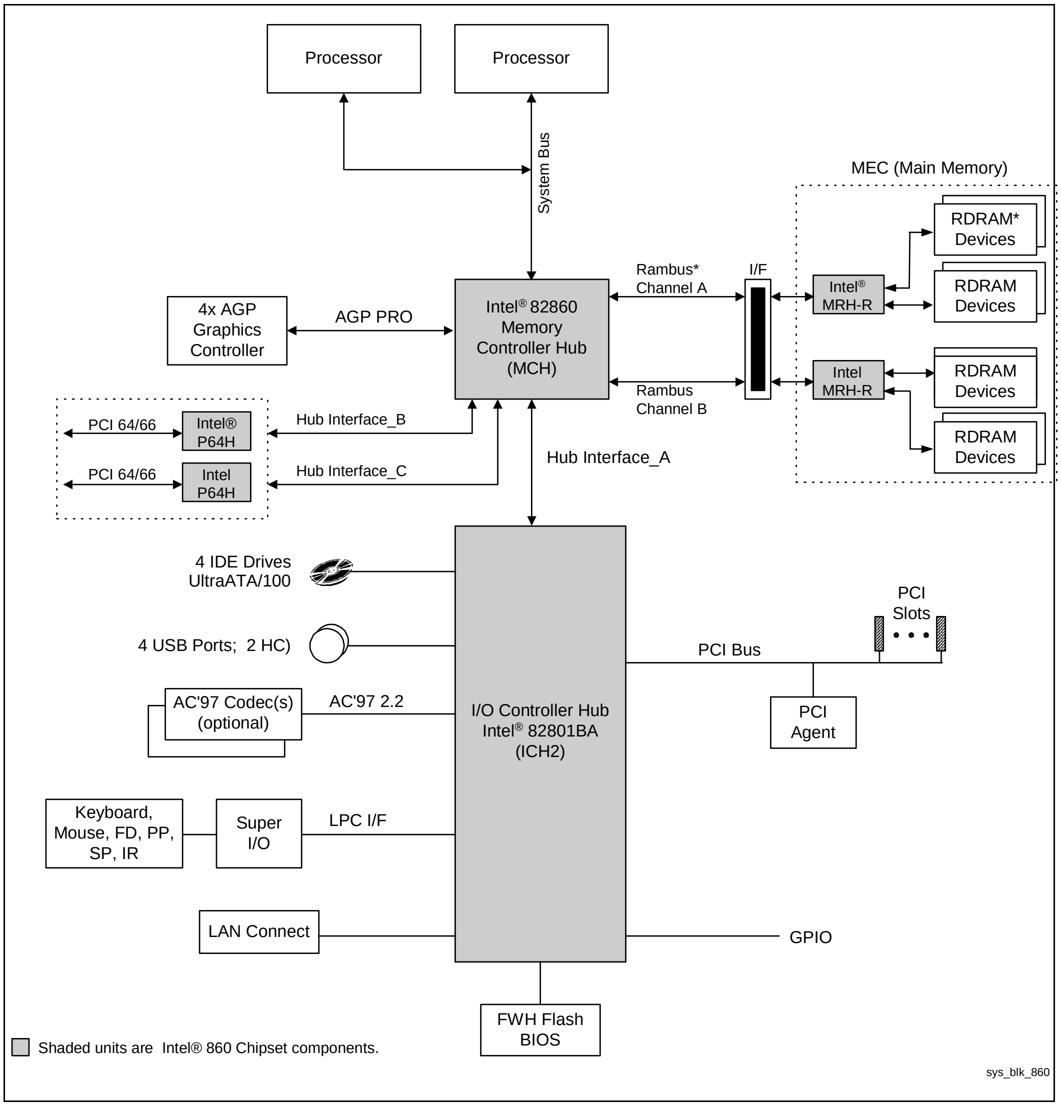
The Intel 860 chipset is optimized for the Intel Xeon processor. The Intel 860 chipset allows flexibility for dual-processor configurations with a 100 MHz system bus (400 MHz data bus). The Intel 860 chipset consists of two main components: Intel 82860 Memory Controller Hub (MCH), and Intel® 82801BA I/O Controller Hub (ICH2).

Architectural expansion is provided with the memory expansion card and PCI 64-bit Hub. The Intel 82803AA Memory Repeater Hub (Intel MRH-R) provides memory expansion capabilities for Rambus* Channels. The Intel® 82806AA PCI 64 Hub (Intel P64H) provides PCI bridging functions between the hub interface_B–C and PCI Bus. The Intel 860 chipset components are interconnected via an interface called “hub interface” providing efficient communication between the chipset components.

Additional hardware platform features, supported by Intel 860 chipset, include AGP 4X, Direct RDRAM* devices, Ultra DMA/100/66/33, Low Pin Count interface (LPC), integrated LAN Controller, and Universal Serial Bus (USB). The Intel 860 chipset architecture removes the requirement for the ISA expansion bus that was traditionally integrated into the I/O subsystem of PCIsets/AGPsets. This eliminates many conflicts experienced when installing legacy ISA hardware and drivers.

The Intel 860 chipset is also ACPI compliant and supports Full-On, Stop Grant, Suspend to Disk, and Soft-Off power management states. Through the use of an appropriate LAN device, the Intel 860 chipset also supports wake-on-LAN* for remote administration and troubleshooting.

### Figure 1. Intel® 860 Chipset System Block Diagram



## 82801BA I/O Controller Hub (ICH2)

The ICH is a highly integrated multifunctional I/O Controller Hub that provides the interface to the PCI Bus and integrates many of the functions needed in today's PC platforms. The MCH and ICH communicate over a dedicated hub interface. Intel 82801BA (ICH2) Functions and capabilities include:

- PCI Rev 2.2 compliant with support for 33 MHz PCI operations
- Supports up to 6 Request/Grant pairs (PCI Slots)
- Power management logic support
- Enhanced DMA controller, interrupt controller, and timer functions
- Integrated IDE controller; Ultra ATA/100/66/33
- USB host interface; 2 host controllers and supports 4 USB ports
- Integrated LAN controller

- System Management Bus (SMBus) compatible with most I$^2$C devices; ICH2 has both bus master and slave capability
- Intel® *Audio Codec '97 Component Specification v2.2*-compliant (AC’97) link for audio and telephony codecs; up to 6 channels (ICH2)
- Low Pin Count (LPC) interface
- FWH Interface (FWH Flash BIOS support)
- Alert on LAN* (AOL and AOL2)

### Intel® 82803AA Memory Repeater Hub (Intel® MRH-R)

The Intel MRH-R supports multiple Rambus Channels from an “expansion channel.” Expansion channel is the interconnect between the MCH and the Intel MRH-R. Each Intel MRH-R can support up to two “stick” channels. The Intel MRH-R acts as a pass-through logic with fixed delay for read and write accesses from expansion channels to Rambus Channels. The Intel MRH-R features include:

- Maximum of 1 GB memory per channel
- Refresh and Precharge on a channel upon request from memory controller
- Core logic gating to minimize power consumption
- Clock generation for Direct Rambus* Clock Generator (DRCG)
- Integrated SMBus controller to read/write data from/to SPD EEPROM on the RIMMs

### Intel® 82806AA PCI 64 Hub (Intel® P64H)

The PCI-64 Hub (Intel P64H)) is a peripheral chip that performs PCI bridging functions between the hub interface and the PCI Bus and is used as an integral part of the Intel 860 chipset. The Intel P64H has a 16-bit primary hub interface to the MCH and a secondary 64-bit PCI Bus interface. The 64-bit interfaces inter-operate transparently with either 64-bit or 32-bit devices. The Intel P64H is fully compliant with the *PCI Local Bus Specification, Revision 2.2*. The Intel P64H functions include:

- Integrated PCI low skew clock driver
- I/O APIC

## 1.4 Intel® 82860 MCH Overview

The Intel 82860 Memory Controller Hub (MCH) provides the processor interface, DRAM interface, AGP interface, and hub interfaces in an Intel 860 chipset-based platform. The MCH uses a 1012 OLGA package and its capabilities include:

- Supports single or dual Intel® Xeon™ processor configurations at 100 MHz (400 MHz data bus)
- Parity protection on the system data, address/request, and response bus signals
- AGTL+ host bus with integrated termination supporting 32-bit host addressing
- Supports IOQ depth of 8
- Dual Rambus Channels support 300 and 400 MHz Direct RDRAM device operation

- 4-GB Direct RDRAM device support
- 1.5 Volt AGP interface with 4x SBA/Data Transfer and 2x/4x Fast Write capability
- AGP SERR# signal
- 8-bit, 66 MHz 4x Hub Interface_A to ICH2
- Two 16-bit, 66 MHz 4x hub interfaces
- Advanced power management logic
- Distributed arbitration for highly concurrent operation

### 1.4.1 Processor Interface

The Intel 82860 MCH supports the Intel Xeon processor system bus interface. The primary enhancements over the P6 bus protocol are:

- Source synchronous double-pumped address
- Source synchronous quad-pumped data
- System bus interrupt delivery

The MCH supports a 64-byte cache line size. Up to two processors can be used at a system bus frequency of 100 MHz (400 MHz data bus). The MCH supports a 1:1 Host-to-Direct RDRAM device frequency ratio (400 MHz data bus to 400 MHz Direct RDRAM device). The MCH integrates AGTL+ termination resistors on all of the AGTL+ signals. System bus Dynamic Bus Inversion (DBI) is supported. The MCH provides 36-bit host addressing, allowing the processor to access the entire 16 GB of the MCH’s memory address space. The MCH has an 8-deep In-Order Queue permitting up to eight outstanding pipelined address requests on the host bus.

Host-initiated I/O cycles are positively decoded to AGP, Hub Interface_B, Hub Interface_C, or MCH configuration space. Host-initiated I/O cycles are subtractively decoded to Hub Interface_A. Host-initiated memory cycles are positively decoded to AGP, Hub Interface_B, Hub Interface_C, or main memory and are again subtractively decoded to Hub Interface_A if under 4 GB. AGP semantic memory accesses initiated from AGP to main memory are not snooped on the host bus. Main memory accesses initiated from AGP using PCI semantics and from any hub interface to main memory are snooped on the system bus. Memory accesses whose addresses lie within the AGP aperture are translated using the AGP address translation table, regardless of the originating interface.

The Intel 82860 MCH generates and checks parity for data, address/request, and response signals on the processor bus. The type of error protection and the responses of the MCH are described in the following table.

| Signal Name | Protection | Error Response |
|---|---|---|
| DEP[3:0]# | Parity | **Host Data Parity:** The Intel 82860 MCH can be configured to generate an SERR message when it detects a host data parity error. |
| AP[1:0]# | Parity | **Address Parity:** The Intel 82860 MCH can be configured to generate an SERR message when it detects a host address/request parity error. Since the MCH does not implement the system bus error phase, the erroneous transaction will proceed to completion. |
| RSP# | Parity | **Response Parity:** The Intel 82860 MCH does not detect errors on the response signals, since they are always MCH-driven. Correct response parity is driven in all phases. |

### 1.4.2 Memory Interface

The MCH directly supports two channels of Direct RDRAM device memory operating in lock-step using RSL technology. The MCH Rambus Channels run at 300 MHz and 400 MHz and supports 128/144 and 256/288Mb technology Direct RDRAM devices. The page size for 128/144 Mb Direct RDRAM devices is 1 KB; for 256/288Mb devices, the page size is 2 KB. Up to eight pages can be open simultaneously. A maximum of 64 Direct RDRAM devices are supported on the paired channels without external logic. Each expander adds two stick channels to the main channel, which yields a total of eight Rambus Channels. Table 1 shows the maximum Direct RDRAM device array size and the minimum increment size for the various Direct RDRAM device densities supported.

***Warning:*** Memory Repeater Hubs run at 400 MHz only.

**Table 1. Maximum Memory Supported**

| Direct RDRAM* Device Technology | Directly Supported Maximum | Supported via Expanders[1] (max 1 per channel) Maximum |
|---|---|---|
| 128/144 Mbit | 1 GB | 2 GB |
| 256/288 Mbit | 2 GB | 4 GB |

The MCH provides optional ECC error checking for Direct RDRAM device data integrity. During Direct RDRAM device writes, ECC is generated on a QWord (64-bit) basis. During Direct RDRAM device reads and the read of the data that underlies partial writes, the MCH supports detection of single-bit and multiple-bit errors, and will correct single-bit errors when correction is enabled.

## Table 2. Supported Direct RDRAM* Devices

| Device Tech | Device Quantity | No. of Banks | Page Size |
|---|---|---|---|
| 128 Mbit | 4,8,16 | 16d | 1 KB |
| 128/144 Mbit | 4,8,16 | 2x16d | 1 KB |
| 288 Mbit | 4,8,16 | 16d | 2 KB |
| 256/288 Mbit | 2,4,8,16 | 2x16d | 2 KB |

### Direct RDRAM* Device Thermal Management

The relatively high power dissipation needs of Direct RDRAM device devices necessitate a MCH mechanism capable of putting a number of memory devices into a power-saving mode. Direct RDRAM devices may be in one of three power-management states: Active, Standby, or Nap. The Intel 82860 MCH implements Direct RDRAM device nap mode.

In “pool” mode, two queues are used inside the MCH: the “A” pool contains references to device pairs that are currently in the active mode while the “B” pool contains references to device pairs that are in the standby mode. All devices that are found in neither pool are napping or in standby. The “A” pool may hold between 1 and 8 device pairs, while the “B” pool may be configured to contain between 1 and 16 device pairs. This allows the power consumption to be tuned.

The Intel 82860 MCH also implements a mode in which all devices are turned on and it is assumed that proper system design will provide adequate cooling. This means that all devices that are Not in pool “A” or “B” are in standby mode. Two failsafe mechanisms are supported that protect the Direct RDRAM devices from thermal overload. One mechanism relies on external thermal sensors to assert the OVERT# pin. The other mechanism polls the thermal indicator bits in the Direct RDRAM devices themselves. When either mechanism is activated, the MCH immediately exits the “all devices on” mode and reverts to whatever pool mode has been programmed by system software.

In summary the MCH Direct RDRAM device thermal management includes:

- Pool mode keeps Direct RDRAM device power dissipation within pre-configured bounds
- From 1–8 device pairs in active pool
- From 0–16 device pairs in standby pool
- Remainder of device pairs in nap or standby pool
- SW may change pool size dynamically
- Overtemp condition detected based on external signal or polling thermal sensor bits in Direct RDRAM devices

### 1.4.3 AGP Interface

A single AGP component or connector (not both) is supported by the Intel 82860 MCH's AGP interface.

The AGP interface supports AGP 2.0 including 1x/2x/4x AGP signaling and 2x/4x Fast Writes. AGP semantic cycles to DRAM are not snooped on the host bus. PCI semantic cycles to DRAM are snooped on the host bus. The MCH supports PIPE# or SBA[7:0] AGP address mechanisms, but not both simultaneously. Either the PIPE# or the SBA[7:0] mechanism must be selected during system initialization. AGP semantic (PIPE# or SBA initiated) accesses to memory are not snooped.

PCI semantic (FRAME# initiated) accesses to memory are snooped. There is delayed transaction support for AGP-to-main memory FRAME# semantic reads that cannot be serviced immediately.

Both upstream and downstream addressing is limited to 32 bits for AGP and AGP/PCI transactions. The MCH contains a 32-deep AGP requests queue. High priority accesses are supported. All accesses from the AGP interface that fall within the Graphics Aperture address range pass through an address translation mechanism with a fully associative 20 entry TLB. Accesses between AGP and Hub Interface_A are limited to memory writes originating from Hub Interface_A destined for the AGP bus.

The AGP interface is clocked from a dedicated 66 MHz clock (66IN). The AGP-to-host/core interface is asynchronous.

***Note:*** The AGP buffers operate only in 1.5 V mode. They are not 3.3 V safe.

### 1.4.4 Hub Interface_A

The 8-bit Hub Interface_A connects the MCH to the ICH2. Virtually all communication between the MCH and the ICH2 occurs over Hub Interface_A. Hub Interface_A runs at 66 MHz; this provides a 266MB/s point-to-point hub interface to ICH2 with parity. In addition to the normal traffic types (e.g., Hub Interface_A -to- AGP memory writes, Hub Interface_A-to-DRAM, and processor-to-Hub Interface_A) the following communication also occurs over Hub Interface_A:

- Interrupt related messages
- Power management events as messages
- SMI, SCI, and SERR error indication messages

It is assumed that Hub Interface_A is always connected to an ICH2.

## 1.4.5 Hub Interface_B and Hub Interface_C

The MCH supports two full time 16-bit hub interfaces. The two dedicated 16-bit hub interfaces (Interface_B, Interface_C) run at 66 MHz and provide 266 MB/s (533MB/s) bandwidth with parity. Peer-to-peer accesses between any 16-bit hub interface (B–C) are limited to memory writes.

The 16-bit hub interfaces may or may not be connected to a device. The MCH detects the presence of a device by sampling the HLx[11] input signals. If a hub interface device is not present, its configuration register space is hidden from configuration software. If Hub Interface_C is used, then Hub Interface_B **must** be populated with an Intel 82806AA (P64H) component.

Interface_B and Interface_C traffic types include:

- Memory writes between any 16-bit hub interfaces
- Hub Interface_B and Hub Interface_C - to - AGP memory writes
- Hub Interface_B and Hub Interface_C - to - DRAM
- Processor-to- Hub Interface_B and Hub Interface_C
- Messaging
- MSI Interrupt messages
- SERR error indication

## 1.4.6 MCH Clocking

The MCH has the following clock input pins:

- Differential BCLK0/BCLK1 for the host interface
- 66 MHz clock input for the AGP and Hub Interface_A
- Differential CTM/CTM# and CFM/CFM# for each of the two RACs.

Clock Synthesizer chip(s) are responsible for generating the system host clocks, AGP and hub interface clocks, PCI clocks, and Direct RDRAM device clocks. The MCH provides two pairs of feedback signals to the Direct Rambus* Clock Generator (DRCG) chips to keep the Host and Direct RDRAM device clocks aligned. The Host speed is 100 MHz (300/400 MHz data bus). The speed for Direct RDRAM device is 300 MHz or 400 MHz. The MCH does not require any relationship between the HCLKIN host clock and the 66 MHz clock generated for AGP and hub interfaces; they are asynchronous to each other. The AGP and Hub Interfaces (A–C) run at a constant 66 MHz base frequency. The Hub Interfaces run at 4x; AGP transfers can be 1x/2x/4x. Table 3 and Table 4 indicate the frequency ratios between the various interfaces.

### Table 3. MCH Processor System Bus-to-RAC Ratio

| Direct RDRAM* Device Speed (MHz) | RAC I/F Frequency (MHz) | Processor System Bus Frequency Intel® Xeon™ Processor 100 MHz (400 MHz data bus) |
|---|---|---|
| 300 | 100 | 3:4 |
| 400 | 100 | 1:1 |

**Table 4. MCH Processor-to-AGP/Hub Interface Ratio**

| AGP/Hub Interface Unit Frequency (MHz) | Processor System Bus Frequency<br>Intel® Xeon™ Processor<br>100 MHz (400 MHz data bus) |
|---|---|
| AGP: 66 MHz | Asynchronous |
| Hub Interface_A: 66 MHz | Asynchronous |
| Hub Interface_B–C: 66 MHz | Asynchronous |

## 1.4.7 System Interrupts

The MCH supports both Intel® 8259 and Intel Xeon processor system bus interrupt delivery mechanisms. The serial APIC interrupt mechanism is not supported. The Intel 8259 support consists of flushing inbound Hub Interface_A write buffers when an Interrupt Acknowledge cycle is forwarded from the system bus to Hub Interface_A.

Support for the Intel Xeon processor system bus interrupt delivery is new to the Intel 82860 MCH. IOxAPIC and PCI MSI interrupts are generated as memory writes. The MCH decodes upstream memory writes to the range 0FEE0_0000h–0FEEF_FFFFh from any of the Hub Interface_A–C interfaces as messaged based interrupts. The MCH forwards these memory writes, along with the associated write data, to the system bus as an Interrupt Message transaction. Note that since this address does not decode as part of main memory, the write cycle and the write data do not get forwarded to Direct RDRAM device via the write buffer. The MCH provides the response and TRDY# for all Interrupt Message cycles including the ones originating from the MCH. The MCH supports interrupt re-direction for inter-processor interrupts (IPIs) as well as upstream interrupt memory writes.

For message based interrupts system write buffer coherency is maintained by relying on strict ordering of memory writes. The MCH ensures that all memory writes received from a given interface prior to an interrupt message memory write are delivered to the system bus for snooping in the same order that they occur on the given interface.

## 1.4.8 Powerdown Flow

Since the MCH is powered down during STR, the MCH cannot maintain any state information when exiting STR. This means that the entire initialization process when exiting STR must be performed by the BIOS via accesses to the RICM register.

Entry into STR (ACPI S3) is initiated by the Operating System (OS) based on detecting a lack of system activity. The OS unloads all system device drivers as part of the process of entering STR. The OS then writes to the PM1_CNT I/O register in the ICH2 to actually trigger the transition into STR. The ICH2 responds by eventually generating the Go C3 message to the MCH via Hub Interface_A.

# 2 Signal Description

This section provides a detailed description of MCH signals. The signals are arranged in functional groups according to their associated interface (See Figure 2). The states of all of the signals during reset are provided in Section 2.11, *“Pin States during Reset”*.

The “#” 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 “#” is not present after the signal name the signal is asserted when at the high voltage level.

The following notations are used to describe the signal type:

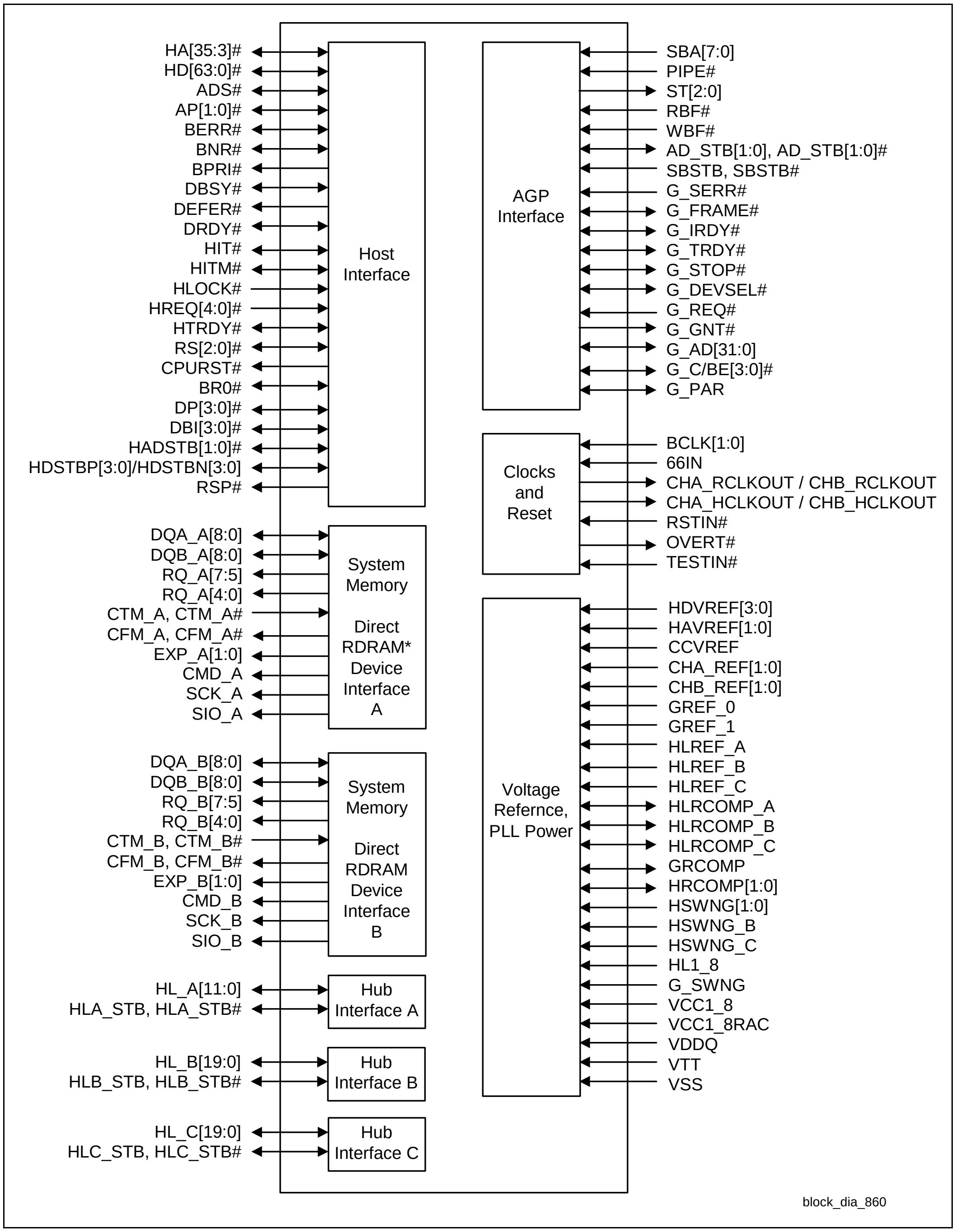
| | |
|---|---|
| I | Input pin |
| O | Output pin |
| I/O | Bi-directional Input/Output pin |
| s/t/s | Sustained Tri-State. This pin is driven to its Inactive state prior to tri-stating. |
| as/t/s | Active Sustained Tri-State. This applies to some of the hub interface signals. This pin is weakly driven to its last driven value. |

The signal description also includes 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. |
| AGP | AGP interface signals. These signals are compatible with AGP 2.0 1.5v Signaling Environment DC and AC Specifications. The buffers are not 3.3 V tolerant. |
| CMOS | CMOS buffers. |
| RSL | Rambus Signaling Level interface signal. Refer to the latest *Direct RDRAM* Component Specification* published by Rambus for complete details. |

***Note:*** Processor address and data bus signals are logically inverted signals (i.e., the actual values are inverted from what appears on the system bus). All system bus control signals follow normal convention; that is, a 0 indicates an active level (low voltage) if the signal is followed by # symbol and a 1 indicates an active level (high voltage) if the signal has no # suffix.

## Figure 2. MCH Signal Diagram



## 2.1 Host Interface Signals

| Signal Name | Type | Description |
|---|---|---|
| ADS# | I/O<br>AGTL+ | **Address Strobe:** The system bus owner asserts ADS# to indicate the first of two cycles of a request phase. |
| AP[1:0]# | I/O<br>AGTL+ | **Address Parity:** The AP[1:0]# lines are driven by the request initiator and provide parity protection for the Request Phase signals. AP[1:0]# are common clock signals and are driven one common clock after the Request Phase.<br>**First Add. Sub-phase** **Second Add. Sub-phase**<br>AP0# HA[35:24]# HA[23:3]#, HREQ[4:0]#<br>AP1# HA[23:3]#, HREQ[4:0]# HA[35:24]#<br>Address parity is correct if there is an even number of electrically low signals (low voltage) in the set consisting of the covered signals plus the parity signal. Note that the MCH only connects to HA[35:3]#. The MCH assumes HA[43:36]# to be electrically high (high voltage) when checking and generating address parity.<br>The MCH may be configured to send a SERR message to the ICH2 over Hub Interface_A when it detects an error on one of the AP[1:0]# signals. |
| BERR# | I/O<br>AGTL+ | **Bus Error:** This signal is not functional. The BERR# pin on the MCH does contain internal pull-ups; thus, it should be connected to the processor BERR# signal to provide system bus termination. |
| BNR# | I/O<br>AGTL+ | **Block Next Request:** Used to block the current request bus owner from issuing a new request. This signal is used to dynamically control the system bus pipeline depth. |
| BPRI# | O<br>AGTL+ | **Bus Priority Request:** The MCH is the only Priority Agent on the system bus. It asserts this signal to obtain the ownership of the address bus. This signal has priority over symmetric bus requests and will cause the current symmetric owner to stop issuing new transactions unless the HLOCK# signal was asserted. |
| BR0# | I/O<br>AGTL+ | **Bus Request 0#:** The MCH pulls the processor bus' BR0# signal low during CPURST#. The processor samples this signal on the active-to-inactive transition of CPURST#. The minimum setup time for this signal is 4 HCLKs. The minimum hold time is 2 clocks and the maximum hold time is 20 HCLKs. BR0# should be tristated after the hold time requirement has been satisfied. |
| CPURST# | O<br>AGTL+ | **CPU Reset:** The CPURST# pin is an output from the MCH. The MCH asserts CPURST# while RSTIN# (PCIRST# from ICH2) is asserted and for approximately 1 ms after RSTIN# is deasserted. The CPURST# allows the processors to begin execution in a known state.<br>**Note:** The ICH2 must provide processor frequency select strap set-up and hold times around CPURST#. This requires strict synchronization between MCH CPURST# deassertion and ICH2 driving the straps. |
| DBSY# | I/O<br>AGTL+ | **Data Bus Busy:** Used by the data bus owner to hold the data bus for transfers requiring more than one cycle. |
| DEFER# | O<br>AGTL+ | **Defer:** Signals that the MCH will terminate the transaction currently being snooped with either a deferred response or with a retry response. |

<table>
<tr><th>Signal Name</th><th>Type</th><th>Description</th></tr>
<tr><td>DP[3:0]#</td><td>I/O<br>AGTL+<br>4x</td><td><b>Host Data Parity:</b> The DP[3:0]# signals provide parity protection for HD[63:0]#. The DP[3:0]# signals are common clock signals and are driven one common clock after the data phases they cover. DP[3:0]# are driven by the same agent driving HD[63:0]#.
<table>
<tr><th></th><th>1st Data Phase</th><th>2nd Data Phase</th><th>3rd Data Phase</th><th>4th Data Phase</th></tr>
<tr><td>HD[15:0]#, DBI0#</td><td>DP3#</td><td>DP2#</td><td>DP1#</td><td>DP0#</td></tr>
<tr><td>HD[31:16]#, DBI1#</td><td>DP0#</td><td>DP3#</td><td>DP2#</td><td>DP1#</td></tr>
<tr><td>HD[47:32]#, DBI2#</td><td>DP1#</td><td>DP0#</td><td>DP3#</td><td>DP2#</td></tr>
<tr><td>HD[63:48]#, DBI3#</td><td>DP2#</td><td>DP1#</td><td>DP0#</td><td>DP3#</td></tr>
</table>
Data parity is correct if there is an even number of electrically low signals (low voltage) in the set consisting of the covered signals plus the parity signal.</td></tr>
<tr><td>DBI[3:0]#</td><td>I/O<br>AGTL+<br>4x</td><td><b>Dynamic Bus Inversion:</b> These signals are driven along with the HD[63:0]# signals. They indicate if the associated signals are inverted. DBI[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.<br>DBI3# = HD[63:48]#<br>DBI2# = HD[47:32]#<br>DBI1# = HD[31:16]#<br>DBI0# = HD[15:0]#</td></tr>
<tr><td>DRDY#</td><td>I/O<br>AGTL+</td><td><b>Data Ready:</b> Asserted for each cycle that data is transferred.</td></tr>
<tr><td>HA[35:3]#</td><td>I/O<br>AGTL+<br>2x</td><td><b>Host Address Bus:</b> HA[35:3]# connect to the processor address bus. During processor cycles, the HA[35:3]# are inputs. The MCH drives HA[35:3]# during snoop cycles on behalf of hub interface and AGP/Secondary PCI initiators. HA[35:3]# are transferred at 2x rate. Note that the address is inverted on the system bus.<br>After reset, the value on HA7# is sampled by all system bus agents, including the MCH, on the rising edge of CPURST#. Its latched value determines the maximum IOQ depth mode supported on the system bus. If HA7# is sampled low, the IOQ depth on the bus is one. If HA7# is sampled high, the IOQ depth on the bus is the maximum of 12. When the IOQ depth on the bus is set to 12, the MCH limits the number of queued transactions by asserting BNR#, since the MCH has an IOQ of depth 8.</td></tr>
<tr><td>HADSTB[1:0]#</td><td>I/O<br>AGTL+<br>2x</td><td><b>Host Address Strobe:</b> HADSTB[1:0]# are the source synchronous strobes used to transfer HA[35:3]# and HREQ[4:0]# at the 2x transfer rate.<br>HADSTB0# = AP0#, HA[16:3]#, HREQ[4:0]#<br>HADSTB1# = AP1#, HA[35:17]#</td></tr>
<tr><td>HD[63:0]#</td><td>I/O<br>AGTL+<br>4x</td><td><b>Host Data:</b> These signals are connected to the processor data bus. In enhanced mode HD[63:0]# are transferred at 4x rate. Note that the data signals are inverted on the system bus.</td></tr>
</table>

| Signal Name | Type | Description |
|---|---|---|
| HDSTBP[3:0]#<br>HDSTBN[3:0]# | I/O<br>AGTL+<br>4x | **Differential Host Data Strobes:** These are the differential source synchronous strobes used to transfer HD[63:0]# and DBI[3:0]# at the 4x transfer rate.<br>HDSTBP3#, HDSTBN3# = HD[63:48]#, DBI3#<br>HDSTBP2#, HDSTBN2# = HD[47:32]#, DBI2#<br>HDSTBP1#, HDSTBN1# = HD[31:16]#, DBI1#<br>HDSTBP0#, HDSTBN0# = HD[15:0]#, DBI0# |
| HIT# | I/O<br>AGTL+ | **Hit:** This signal indicates that a caching agent holds an unmodified version of the requested line. It is, also, driven in conjunction with HITM# by the target to extend the snoop window. |
| HITM# | I/O<br>AGTL+ | **Hit Modified:** HITM# indicates that a caching agent holds a modified version of the requested line and that this agent assumes responsibility for providing the line. HITM# is, also, driven in conjunction with HIT# to extend the snoop window. |
| HLOCK# | I<br>AGTL+ | **Host Lock:** All system bus cycles sampled with the assertion of HLOCK# and ADS#, until the negation of HLOCK#, must be atomic (i.e., no hub interface or AGP snoopable access to DRAM are allowed when HLOCK# is asserted by the processor). |
| HREQ[4:0]# | I/O<br>AGTL+<br>2x | **Host Request Command:** These signals define the attributes of the request. In enhanced mode HREQ[4:0]# are transferred at 2x rate. HREQ[4:0]# 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. |
| HTRDY# | O<br>AGTL+ | **Host Target Ready:** HTRDY# indicates that the target of the processor transaction is able to enter the data transfer phase. |
| RS[2:0]# | O<br>AGTL+ | **Response Status:** RS[2:0]# indicate the type of response according to the following the table:<br>000 = Idle state<br>001 = Retry response<br>010 = Deferred response<br>011 = Reserved (not driven by MCH)<br>100 = Hard Failure (not driven by MCH)<br>101 = No data response<br>110 = Implicit writeback<br>111 = Normal data response |
| RSP# | O<br>AGTL+ | **Response Parity:** RSP# provides parity protection for the RS[2:0]# signals. It is always driven by the MCH and must be valid on all clocks. Response Parity is correct if there are an even number of low signals (low voltage) in the set consisting of the RS[2:0]# signals and the RSP# signal itself.<br>The MCH may be configured to send an SERR message to the ICH2 over Hub Interface_A when it detects an error on the RSP# signal. |

## 2.2 Rambus* Channel A

| Signal Name | Type | Description |
|---|---|---|
| DQA_A[8:0] | I/O<br>RSL | **Direct RDRAM* Device Data (A):** Data signals used for read and write operations on Rambus* Channel A. |
| DQB_A[8:0] | I/O<br>RSL | **Direct RDRAM Device Data (A):** Data signals used for read and write operations on Rambus Channel A. |
| RQ_A[7:5] | O<br>RSL | **Row Access Control (A):** Three request package pins containing control and address information for row accesses.<br>**Note:** RQ_A[7:5] are sometimes referred to as the “ROW_A[2:0]” signals. |
| RQ_A[4:0] | O<br>RSL | **Column Access Control (A):** Five request package pins containing control and address information for column accesses.<br>**Note:** RQ_A[4:0] are sometimes referred to as the “COL_A[4:0]” signals. |
| CTM_A | I<br>RSL | **Clock To Master (A):** CTM_A is one of the two differential transmit clock signals used for Direct RDRAM device operations on Rambus Channel A. It is an input to the MCH and is generated from an external clock synthesizer. |
| CTM_A# | I<br>RSL | **Clock To Master Compliment (A):** CTM_A# is one of the two differential transmit clock signals used for Direct RDRAM device operations on Rambus Channel A. It is an input to the MCH and is generated from an external clock synthesizer. |
| CFM_A | O<br>RSL | **Clock From Master (A):** CFM_A is one of the two differential receive clock signals used for Direct RDRAM device operations on Rambus Channel A. It is an output from the MCH. |
| CFM_A# | O<br>RSL | **Clock From Master Compliment( A):** CFM_A# is one of the two differential receive clock signals used for Direct RDRAM device operations on Rambus Channel A. It is an output from the MCH. |
| EXP_A[1:0] | O<br>RSL | **Expansion (A):** These signals are used to communicate to an external Direct RDRAM device repeater on Rambus Channel A. The repeater increases the maximum memory size supported by the MCH. |
| CMD_A | O<br>CMOS | **Command (A):** Command output to the Direct RDRAM device devices used for power mode control, configuring the SIO daisy chain, and framing SIO operations. |
| SCK_A | O<br>CMOS | **Serial Clock (A):** This signal provides clocking for register accesses and selects Rambus Channel A devices for power management. |
| SIO_A | I/O<br>CMOS | **Serial Input/Output (A):** This signal is a bi-directional serial data signal used for device initialization, register operations, power mode control, and device reset. |

## 2.3 Rambus* Channel B

| Signal Name | Type | Description |
|---|---|---|
| DQA_B[8:0] | I/O RSL | **Direct RDRAM* Device Data (B):** Data signals used for read and write operations on Rambus* Channel B. |
| DQB_B[8:0] | I/O RSL | **Direct RDRAM Device Data (B):** Data signals used for read and write operations on Rambus Channel B. |
| RQ_B[7:5] | O<br>RSL | **Row Access Control (B):** Three request package pins containing control and address information for row accesses.<br>**Note:** RQ_B[7:5] are sometimes referred to as the “ROW_B[2:0]” signals. |
| RQ_B[4:0] | O<br>RSL | **Column Access Control (B):** Five request package pins containing control and address information for column accesses.<br>**Note:** RQ_B[4:0] are sometimes referred to as the “COL_B[4:0]” signals. |
| CTM_B | I<br>RSL | **Clock To Master (B):** CTM_B is one of the two differential transmit clock signals used for Direct RDRAM device operations on Rambus Channel B. It is an input to the MCH and is generated from an external clock synthesizer. |
| CTM_B# | I<br>RSL | **Clock To Master Compliment (B):** CTM_B# is one of the two differential transmit clock signals used for Direct RDRAM device operations on Rambus Channel B. It is an input to the MCH and is generated from an external clock synthesizer. |
| CFM_B | O<br>RSL | **Clock From Master (B):** CFM_B is one of the two differential receive clock signals used for Direct RDRAM device operations on Rambus Channel. It is an output from the MCH. |
| CFM_B# | O<br>RSL | **Clock From Master Compliment (B):** CFM_B# is one of the two differential receive clock signals used for Direct RDRAM device operations on Rambus Channel B. It is an output from the MCH. |
| EXP_B[1:0] | O<br>RSL | **Expansion (B):** These signals are used to communicate to an external Direct RDRAM device repeater on Rambus Channel B. The repeater increases the maximum memory size supported by the MCH. |
| CMD_B | O<br>CMOS | **Command (B):** CMD_B is a command output to the Direct RDRAM device devices used for power mode control, configuring the SIO daisy chain, and framing SIO operations. |
| SCK_B | O<br>CMOS | **Serial Clock (B):** This signal provides clocking for register accesses and selects Rambus Channel B devices for power management. |
| SIO_B | I/O<br>CMOS | **Serial Input/Output (B):** SIO_B is a bi-directional serial data signal used for device initialization, register operations, power mode control, and device reset. |

## 2.4 Hub Interface_A Signals

| Signal Name | Type | Description |
|---|---|---|
| HL_A[11:0] | I/O<br>CMOS | **Hub Interface_A Signals:** Signals used for the hub interface. |
| HLA_STB | I/O<br>CMOS | **Hub Interface_A Strobe:** One of two differential strobe signals used to transmit or receive packet data over Hub Interface_A. |
| HLA_STB# | I/O<br>CMOS | **Hub Interface_A Strobe Compliment:** One of two differential strobe signals used to transmit or receive packet data over Hub Interface_A. |

## 2.5 Hub Interface_B

| Signal Name | Type | Description |
|---|---|---|
| HL_B[19:0] | I/O<br>CMOS | **Hub Interface_B Signals:** Signals used for the hub interface. |
| HLB_STB[1:0] | I/O<br>CMOS | **Hub Interface_B Strobe:** One of two differential strobe signals used to transmit or receive packet data over Hub Interface_B. |
| HLB_STB[1:0]# | I/O<br>CMOS | **Hub Interface_B Strobe Compliment:** One of two differential strobe signals used to transmit or receive packet data over Hub Interface_B. |

## 2.6 Hub Interface_C

| Signal Name | Type | Description |
|---|---|---|
| HL_C[19:0] | I/O<br>CMOS | **Hub Interface_C Signals:** Signals used for the hub interface. |
| HLC_STB[1:0] | I/O<br>CMOS | **Hub Interface_C Strobe:** One of two differential strobe signals used to transmit or receive packet data over Hub Interface_C. |
| HLC_STB[1:0]# | I/O<br>CMOS | **Hub Interface_C Strobe Compliment:** One of two differential strobe signals used to transmit or receive packet data over Hub Interface_C. |

## 2.7 AGP Interface Signals

### 2.7.1 AGP Addressing Signals

| Signal Name | Type | Description |
|---|---|---|
| PIPE# | I<br>AGP | **Pipeline:**<br>**During PIPE# Operation:** This signal is asserted by the AGP master to indicate a full width address is to be enqueued on by the target using the AD bus. One address is placed in the AGP request queue on each rising clock edge while PIPE# is asserted. When PIPE# is deasserted, no new requests are queued across the AD bus.<br>**During SBA Operation:** Not used.<br>**During FRAME# Operation:** Not used.<br>N**ote:** Initial AGP designs may not use PIPE# (i.e., PCI only 66 MHz). Therefore, an 8 KΩ pull-up resistor connected to this pin is required on the motherboard. |
| SBA[7:0] | I<br>AGP | **Side-band Addressing:**<br>**During PIPE# Operation:** Not used.<br>**During SBA Operation:** These signals are used by the AGP master (graphics component) to place addresses in the AGP request queue. The SBA bus and AD bus operate independently (i.e., transaction can proceed on the SBA bus and the AD bus simultaneously).<br>**During FRAME# Operation:** Not used.<br>**Note:** These signals implement internal pull-ups with a nominal value of 8 kΩ. When AGP is not enabled, these pull-ups are disabled. |

**NOTE:** The above table contains two mechanisms to queue requests by the AGP master. Note that the master can only use one mechanism. The master may not switch methods without a full reset of the system. When PIPE# is used to queue addresses, the master is not allowed to queue addresses using the SBA bus. For example, during configuration time, if the master indicates that it can use either mechanism, the configuration software will indicate which mechanism the master will use. Once this choice has been made, the master will continue to use the mechanism selected until the master is reset (and reprogrammed) to use the other mode. This change of modes is not a dynamic mechanism; rather, it is a static decision when the device is first being configured after reset.

## 2.7.2 AGP Flow Control Signals

| Signal Name | Type | Description |
|---|---|---|
| RBF# | I<br>AGP | **Read Buffer Full:**<br>**During PIPE# and SBA Operation:** Read buffer full indicates if the master is ready to accept previously requested low priority read data. When RBF# is asserted, the MCH is not allowed to initiate the return low priority read data. That is, the MCH can finish returning the data for the request currently being serviced; however, it cannot begin returning data for the next request. RBF# is only sampled at the beginning of a cycle.<br>If the AGP master is always ready to accept return read data, then it is not required to implement this signal.<br>**During FRAME# Operation:** Not used. |
| WBF# | I<br>AGP | **Write-Buffer Full:**<br>**During PIPE# and SBA Operation:** Write buffer full indicates if the master is ready to accept Fast Write data from the MCH. When WBF# is asserted, the MCH is not allowed to drive Fast Write data to the AGP master. WBF# is only sampled at the beginning of a cycle.<br>If the AGP master is always ready to accept fast write data, then it is not required to implement this signal.<br>**During FRAME# Operation:** Not used. |

## 2.7.3 AGP Status Signals

| Signal Name | Type | Description |
|---|---|---|
| ST[2:0] | O<br>AGP | **Status Bus:**<br>**During PIPE# and SBA Operation:** Provides information from the arbiter to an AGP Master on what it may do. ST[2:0] only have meaning to the master when its G_GNT# is asserted. When G_GNT# is deasserted, these signals have no meaning and must be ignored. Refer to the *AGP Interface Specification Revision 2.0* for further explanation of the ST[2:0] values and their meanings.<br>**During FRAME# Operation:** These signals are not used during FRAME#-based operation, except that a ‘111’ indicates that the master may begin a FRAME# transaction. |

### 2.7.4 AGP Strobes

| Signal Name | Type | Description |
|---|---|---|
| AD_STB0 | I/O<br>(s/t/s)<br>AGP | **AD Bus Strobe-0:**<br>**During 1X Operation:** Not used.<br>**During 2X Operation:** During 2X operation, this signal provides timing for the G_AD[15:0] and G_C/BE[1:0]# signals. The agent that is providing the data will drive this signal.<br>**During 4X Operation:** During 4X operation, this is one-half of a differential strobe pair that provides timing information for the G_AD[15:0] and G_C/BE[1:0]# signals. |
| AD_STB0# | I/O<br>(s/t/s)<br>AGP | **AD Bus Strobe-0 Compliment:**<br>**During 1X Operation:** Not used.<br>**During 2X Operation:** Not used.<br>**During 4X Operation:** During 4X operation, this is one-half of a differential strobe pair that provides timing information for the G_AD[15:0] and G_C/BE[1:0]# signals. The agent that is providing the data will drive this signal. |
| AD_STB1 | I/O<br>(s/t/s)<br>AGP | **AD Bus Strobe-1:**<br>**During 1X Operation:** Not used.<br>**During 2X Operation:** During 2X operation, this signal provides timing for the G_AD[16:31] and G_C/BE[2:3]# signals. The agent that is providing the data will drive this signal.<br>**During 4X Operation:** During 4X operation, this is one-half of a differential strobe pair that provides timing information for the G_AD[16:31] and G_C/BE[2:3]# signals. The agent that is providing the data will drive this signal. |
| AD_STB1# | I/O<br>(s/t/s)<br>AGP | **AD Bus Strobe-1 Compliment**<br>**During 1X Operation:** Not used.<br>**During 2X Operation:** Not used.<br>**During 4X Operation:** During 4X operation, this is one-half of a differential strobe pair that provides timing information for the G_AD[16:31] and G_C/BE[2:3]# signals. The agent that is providing the data will drive this signal. |
| SB_STB | I<br>AGP | **SBA Bus Strobe:**<br>**During 1X Operation:** Not used.<br>**During 2X Operation:** During 2X operation, this signal provides timing for the SBA bus signals. The agent that is driving the SBA bus will drive this signal.<br>**During 4X Operation:** During 4X operation, this is one-half of a differential strobe pair that provides timing information for the SBA bus signals. The agent that is driving the SBA bus will drive this signal. |

| Signal Name | Type | Description |
|---|---|---|
| SB_STB# | I<br>AGP | **SBA Bus Strobe Compliment:**<br>**During 1X Operation:** Not used.<br>**During 2X Operation:** Not used.<br>**During 4X Operation:** During 4X operation, this is one-half of a differential strobe pair that provides timing information for the SBA bus signals. The agent that is driving the SBA bus will drive this signal. |

## 2.7.5 AGP/PCI Signals-Semantics

For transactions on the AGP interface carried using AGP FRAME# protocol these signals operate similar to their semantics in the PCI 2.1 specification. The role of all AGP FRAME# signals is described below.

| Signal Name | Type | Description |
|---|---|---|
| G_FRAME# | I/O<br>s/t/s<br>AGP | **FRAME:**<br>**During PIPE# and SBA Operation:** Not used.<br>**During Fast Write Operation:** G_FRAME# is used to frame transactions as an output from the MCH during Fast Writes.<br>**During FRAME# Operation:** G_FRAME# is an output when the MCH acts as an initiator on the AGP Interface. G_FRAME# is asserted by the MCH to indicate the beginning and duration of an access. G_FRAME# is an input when the MCH acts as a FRAME#-based AGP target. As a FRAME#-based AGP target, the MCH latches the G_C/BE[3:0]# and the G_AD[31:0] signals on the first clock edge on which it samples G_FRAME# active. |
| G_IRDY# | I/O<br>s/t/s<br>AGP | **Initiator Ready:**<br>**During PIPE# and SBA Operation:** Not used while enqueueing requests via AGP SBA and PIPE#, but used during the data phase of PIPE# and SBA transactions.<br>**During FRAME# Operation:** G_IRDY# is an output when MCH acts as a FRAME#-based AGP initiator and an input when the MCH acts as a FRAME#-based AGP target. The assertion of G_IRDY# indicates the current FRAME#-based AGP bus initiator's ability to complete the current data phase of the transaction.<br>**During Fast Write Operation:** G_IRDY# indicates the AGP compliant master is ready to provide all write data for the current transaction. Once G_IRDY# is asserted for a write operation, the master is not allowed to insert wait states. The master is never allowed to insert a wait-state during the initial data transfer (32 bytes) of a write transaction. However, it may insert wait states after each 32-byte block is transferred. |

| Signal Name | Type | Description |
|---|---|---|
| G_TRDY# | I/O<br>s/t/s<br>AGP | **Target Ready:**<br>**During PIPE# and SBA Operation:** Not used while enqueueing requests via AGP SBA and PIPE#, but used during the data phase of PIPE# and SBA transactions.<br>**During FRAME# Operation:** G_TRDY# is an input when the MCH acts as an AGP initiator and an output when the MCH acts as a FRAME#-based AGP target. The assertion of G_TRDY# indicates the target's ability to complete the current data phase of the transaction.<br>**During Fast Write Operation:** G_TRDY# indicates the AGP compliant target is ready to receive write data for the entire transaction (when the transfer size is less than or equal to 32 bytes) or is ready to transfer the initial or subsequent block (32 bytes) of data when the transfer size is greater than 32 bytes. The target is allowed to insert wait states after each block (32 bytes) is transferred on write transactions.<br>**Note:** For AGP Fast Writes, If an AGP master (acting as a PCI target) asserts G_TRDY# before the first throttling point, the system will hang. |
| G_STOP# | I/O<br>s/t/s<br>AGP | **Stop:**<br>**During PIPE# and SBA Operation:** Not used.<br>**During FRAME# Operation:** G_STOP# is an input when the MCH acts as a FRAME#-based AGP initiator and an output when the MCH acts as a FRAME#-based AGP target. G_STOP# is used for disconnect, retry, and abort sequences on the AGP interface. |
| G_DEVSEL# | I/O<br>s/t/s<br>AGP | **Device Select:**<br>**During PIPE# and SBA Operation:** Not used.<br>**During FRAME# Operation:** G_DEVSEL#, when asserted, indicates that a FRAME#-based AGP target device has decoded its address as the target of the current access. The MCH asserts G_DEVSEL# based on the DRAM address range being accessed by a PCI initiator. As an input it indicates whether any device on the bus has been selected.<br>**During Fast Write Operation:** G_DEVSEL# is used when the transaction cannot complete during the block data transfer. |
| G_REQ# | I<br>AGP | **Request:**<br>**During SBA Operation:** Not used.<br>**During PIPE# and FRAME# Operation:** G_REQ#, when asserted, indicates that a FRAME# or PIPE# based AGP master is requesting use of the AGP interface. This signal is an input into the MCH. |
| G_GNT# | O<br>AGP | **Grant:**<br>**During SBA, PIPE# and FRAME# Operation:** G_GNT# along with the information on the ST[2:0] signals (status bus) indicates how the AGP interface will be used next. Refer to the AGP Interface Specification revision 2.0 for further explanation of the ST[2:0] values and their meanings.<br>This signal requires an external pull-up of 8.2 kΩ. |

| Signal Name | Type | Description |
|---|---|---|
| G_AD[31:0] | I/O<br>AGP | **Address/Data Bus:**<br>**During PIPE# and FRAME# Operation:** G_AD[31:0] are used to transfer both address and data information on the AGP interface.<br>**During SBA Operation:** G_AD[31:0] are used to transfer data on the AGP interface. |
| G_C/BE[3:0]# | I/O<br>AGP | **Command/Byte Enable:**<br>**During FRAME# Operation:** During the address phase of a transaction, G_C/BE[3:0]# define the Bus command. During the data phase, G_C/BE[3:0]# are used as byte enables. The byte enables determine which byte lanes carry meaningful data. The commands issued on the G_C/BE[3:0]# signals during FRAME# based AGP are the same C/BE[3:0]# command described in the PCI 2.1 specification.<br>**During PIPE# Operation:** When an address is enqueued using PIPE#, the G_C/BE[3:0]# signals carry command information. Refer to the *AGP 2.0 Interface Specification, Revision 2.0* for the definition of these commands. The command encoding used during PIPE# based AGP is **Different** than the command encoding used during FRAME# based AGP cycles (or standard PCI cycles on a PCI bus).<br>**During SBA Operation:** Not used. |
| G_PAR | I/O<br>AGP | **Parity:**<br>**During FRAME# Operation:** This signal is driven by the MCH when it acts as a FRAME#-based AGP initiator during address and data phases for a write cycle, and during the address phase for a read cycle. G_PAR is driven by the MCH when it acts as a FRAME# based AGP target during each data phase of a FRAME# based AGP memory read cycle. Even parity is generated across G_AD[31:0] and G_C/BE[3:0]#.<br>**During SBA and PIPE# Operation:** This signal is not used during SBA and PIPE# operation. |
| G_SERR# | I<br>AGP | **SERR#:** The G_SERR# signal is used by a PCI device to signal an error on a PCI device attached to AGP/PCI. The MCH may be configured to send a SERR message to the ICH2 upon the assertion of G_SERR#. |

**NOTE:** PCIRST# from the ICH2 is connected to RSTIN# and is used to reset AGP interface logic within the MCH. The AGP agent will also use PCIRST# provided by the ICH2 as an input to reset its internal logic. The LOCK# signal is **not supported** on the AGP interface (even for PCI operations). PERR# signal is **not supported** on the AGP interface.

## 2.8 Clocks, Reset, and Miscellaneous

| Signal Name | Type | Description |
|---|---|---|
| BCLK[1:0] | I<br>CMOS | **Differential Host Clock In:** These pins receive a 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. |
| 66IN | I<br>CMOS | **66 MHz Clock In:** This pin receives a 66 MHz clock from the clock synthesizer. The AGP and Hub Interface_A–C clock domains use this clock.<br>**Note:** This clock input is 3.3 V tolerant. |
| CHA_RCLKOUT/<br>CHB_RCLKOUT | O<br>CMOS | **Direct RDRAM* Device Clock Out:** This pin provides divided down versions of the Direct RDRAM device clock as feedback to the Direct RDRAM device clock synthesizers for phase alignment.<br>**Note:** This pin will only be driven to 1.8 V. |
| CHA_HCLK0UT/<br>CHB_HCLKOUT | O<br>CMOS | **Host Clock Out:** This pin provides divided down versions of the host clock as feedback to the Direct RDRAM device clock synthesizers for phase alignment.<br>**Note:** This pin will only be driven to 1.8 V. |
| RSTIN# | I<br>CMOS | **Reset In:** When asserted this signal will asynchronously reset the MCH logic. This signal is connected to the PCIRST# output of the ICH2. All AGP output and bi-directional signals will also tri-state compliant to *PCI Revision 2.0 and 2.1* specifications.<br>This input should have a Schmidt trigger to avoid spurious resets.<br>**Note:** This input needs to be 3.3 V tolerant. |
| TESTIN# | I<br>CMOS | **Test Input:** TESTIN# is used for manufacturing and board level test purposes. This signal is internally pulled up to VDDQ. |
| OVERT# | I<br>CMOS | **Overtemperature Condition:** This signal, when asserted, indicates that the Direct RDRAM devices have exceeded the system designer’s target maximum temperature. The MCH’s response to this signal is to shift the Direct RDRAM device controller into the throttling/pool mode placed into the DPS register. Software should program this register with a value that is more conservative than the current value in order to make the OVERT# pin useful. It is internally pulled up to VDDQ. |

## 2.9 Voltage References, PLL Power

| Signal Name | Type | Description |
|---|---|---|
| HDVREF[3:0] | | **Host Data Reference Voltage.** Reference voltage input for the 4x Data Signals of the Host AGTL+ interface. Connect to 2/3 VTT with 2% tolerance. |
| HAVREF[1:0] | | **Host Address Reference Voltage.** Reference voltage input for the 2x address signals of the host AGTL+ interface. Connect to 2/3 VTT with 2% tolerance. |
| CCVREF | | **Host Common Clock Reference Voltage.** Reference voltage input for the common clock signals of the host AGTL+ interface. Connect to 2/3 VTT with 2% tolerance. |
| CHA_REF[1:0] | | **Rambus* Channel A Reference:** Reference voltage input for the Rambus Channel A RSL interface. |
| CHB_REF[1:0] | | **Rambus Channel B Reference:** Reference voltage input for the Rambus Channel B RSL interface. |
| GREF_0 | | **AGP:** Reference voltage input for the AGP interface. |
| GREF_1 | | **AGP:** Reference voltage input for the AGP interface. |
| HLREF_A | | **Hub Interface_A Reference:** Reference voltage input for the Hub Interface_A.<br>**Normal Mode:** Connect to 1/2 VCC1_8 with 2% tolerance<br>**Enhanced Mode:** Connect to 2/3 VCC1_8 with 2% tolerance |
| HLREF_B | | **Hub Interface_B Reference:** Reference voltage input for the Hub Interface_B. Connect to 2/3 VCC1_8 with 2% tolerance. |
| HLREF_C | | **Hub Interface_C Reference:** Reference voltage input for the Hub Interface_C. Connect to 2/3 VCC1_8 with 2% tolerance. |
| HLRCOMP_A | I/O<br>CMOS | **Compensation for Hub Interface_A:** This signal is used to calibrate the Hub Interface_A I/O buffers.<br>**Normal Mode:** Connect to 39 Ω 1% or 40 Ω 2% Pull-down<br>**Enhanced Mode:** Connect to 30 Ω 1% Pull-down |
| HLRCOMP_B | I/O<br>CMOS | **Compensation for Hub Interface_B:** This signal is used to calibrate the Hub Interface_B I/O buffers. Refer to the *Intel® Xeon™ Processor and Intel® 860 Chipset Platform Design Guide* for proper connections. |
| HLRCOMP_C | I/O<br>CMOS | **Compensation for Hub Interface_C:** This signal is used to calibrate the Hub Interface_C I/O buffers. Refer to the *Intel® Xeon™ Processor and Intel® 860 Chipset Platform Design Guide* for proper connections. |
| GRCOMP | I/O<br>CMOS | **Compensation for AGP:** This signal is used to calibrate AGP buffers. Connect to a 39 Ω with a 1% tolerance or a 40 Ω with a 2% tolerance Pull-down. |
| HRCOMP[1:0] | I/O<br>CMOS | **Compensation for Host:** This signal is used to calibrate the host AGTL+ I/O buffers. Connect to 20.75 Ω with a 1% tolerance pull-down. |
| HSWNG[1:0] | I<br>CMOS | **Host Compensation Reference Voltage:** Reference voltage input for the compensation logic. Connect to 1/3 VTT with a 2% tolerance. |

| Signal Name | Type | Description |
|---|---|---|
| HLSWNG_B | I<br>CMOS | **Hub Interface_B Compensation Reference Voltage:** Reference voltage input for the compensation logic. Connect to 1/3 VCC1_8 with a 2% tolerance. |
| HLSWNG_C | I<br>CMOS | **Hub Interface_C Compensation Reference Voltage:** Reference voltage input for the compensation logic. Connect to 1/3 VCC1_8 with a 2% tolerance. |
| HL1_8 | I<br>CMOS | **Connect to VCC1_8** |
| G_SWNG | I<br>CMOS | **AGP Compensation Reference Voltage:** Reference voltage input for the compensation logic. Hooked to GREF_0. |
| VCC1_8 | | **Power:** The 1.8 V Power input pins |
| VCC1_8RAC | | **Power:** The 1.8 V RAC Power pins |
| VDDQ | | **Power:** The power supply input for the AGP I/O supply (1.5 V) |
| VTT | | **Power:** The AGTL+ bus termination voltage inputs |
| VSS | | **Ground** |

## 2.10 Strap Signals

This table indicates the strap options invoked by various MCH signal pins.

| Pin | Strap Name | Description |
|---|---|---|
| BUSPARK | System Bus Bus Parking | This signal is reflected on HA15# to configure the processor(s) in system bus parking enabled mode. This signal has an internal pull-up to VDDQ. Bus parking should be enabled for single processor systems and disabled for dual processor systems. |
| HLA_ENH# | Hub Interface_A Enhanced Mode Enable | This signal is used as the HL_A normal/enhanced mode operation strap. This signal has an internal pull-up to VDDQ. |

## 2.11 Pin States during Reset

Table 5 indicates the MCH signal pin states during reset assertion.

| | |
|---|---|
| Z | Tri-state Outputs |
| ISO | Isolate Inputs in Inactive State |
| S | Strap sampled on RSTIN# rising edge |
| H | Driven High |
| L | Driven Low |
| D | Drive Outputs to Functional Logic Level |
| I | Input Active |
| U | Undefined – Indeterminate |

### Table 5. Pin States during Reset

| Signal Name | State During RSTIN# Assertion |
|---|---|
| **Host Interface** | |
| CPURST# | L |
| HA[35:8,6:3]# | Z/I |
| HA7# | Z/I/S |
| HADSTB[1:0]# | Z/I |
| HD[63:0]# | Z/I |
| HDSTBP[3:0]# | Z/I |
| HDSTBN[3:0]# | Z/I |
| DBI[3:0]# | Z/I |
| ADS# | Z/I |
| BNR# | Z/I |
| BPRI# | Z/I |
| DBSY# | Z/I |
| DEFER# | Z/I |
| DRDY# | Z/I |
| HIT# | Z/I |
| HITM# | Z/I |
| HLOCK# | Z/I |
| HREQ[4:0]# | Z/I |
| HTRDY# | Z/I |
| RS[2:0]# | Z/I |
| BR0# | Z/I |
| AP[1:0]# | Z/I |
| RSP# | Z/I |
| DP[3:0]# | Z/I |
| HDVREF[3:0] | I |
| HAVREF[1:0] | I |
| CCVREF | I |
| HRCOMP[1:0] | Z |
| HSWNG[1:0] | I |
| **Rambus* Channel A** | |
| DQA_A[8:0] | Z |
| DQB_A[8:0] | Z |
| RQ_A[7:0] | Z |
| CTM_A | I |
| CTM_A# | I |
| CFM_A | Z |
| CFM_A# | Z |
| EXP_A[1:0] | Z |
| SCK_A | L |
| CMD_A | Z |
| SIO_A | Z |
| CHA_REF[1:0] | I |
| **Rambus Channel B** | |
| CHB_DQA[8:0] | Z |
| CHB_DQB[8:0] | Z |
| CHB_RQ[7:0] | Z |
| CHB_CTM | I |
| CHB_CTM# | I |
| CHB_CFM | Z |
| CHB_CFM# | Z |
| EXP_B[1:0] | Z |
| CHB_SCK | L |
| CHB_CMD | Z |
| CHB_SIO | Z |
| CHB_REF[1:0] | I |
| **AGP** | |
| PIPE# | I |
| SBA[7:0] | ISO |
| RBF# | I |
| WBF# | I |
| G_REQ# | I |
| ST[2:0] | L |
| G_GNT# | H |
| AD_STB[1:0] | Z |
| AD_STB[1:0]# | Z |
| SB_STB | I |
| SB_STB# | I |
| G_AD[31:0] | L/I |
| G_C/BE[3:0]# | L/I |
| G_FRAME# | Z/I |
| G_IRDY# | Z/I |
| G_TRDY# | Z/I |
| G_STOP# | Z/I |
| G_DEVSEL# | Z/I |
| G_PAR | L/I |
| G_SERR# | I |
| **Hub Interface_A (Normal Mode)** | |
| HL_A11 | L/I |
| HL_A10 | L/I |
| HL_A9 | L/I |
| HL_A8 | L |
| HL_A[7:4] | L/I |
| HL_A[3:0] | Z |
| HLA_STB | L/I |
| HLA_STB# | H |
| HLRCOMP_A | Z |
| HLREF_A | I |
| **Hub Interface_A (Enhanced Mode)** | |
| HL_A11 | Z/I |
| HL_A10 | Z/I |
| HL_A9 | L/I |
| HL_A8 | H |
| HL_A[7:4] | Z/I |
| HL_A[3:0] | Z/I |
| HLA_STB | Z/I |
| HLA_STB# | Z/I |
| HLRCOMP_A | Z |
| HLREF_A | I |

| Signal Name | State During RSTIN# Assertion |
|---|---|
| **Hub Interface_B (Enhanced Mode)** | |
| HL_B19 | Z/I |
| HL_B18 | Z/I |
| HL_B17 | L/I |
| HL_B16 | H |
| HL_B[15:0] | Z/I |
| HLB_STB[1:0] | Z/I |
| HLB_STB[1:0]# | Z/I |
| HLRCOMP_B | Z |
| HLREF_B | I |

| Signal Name | State During RSTIN# Assertion |
|---|---|
| **Hub Interface_C (Enhanced Mode)** | |
| HL_C19 | Z/I |
| HL_C18 | Z/I |
| HL_C17 | L/I |
| HL_C16 | H |
| HL_C[15:0] | Z/I |
| HLC_STB[1:0] | Z/I |
| HLC_STB[1:0]# | Z/I |
| HLRCOMP_C | Z |
| HLREF_C | I |

| Signal Name | State During RSTIN# Assertion |
|---|---|
| **Clocks and Misc.** | |
| BCLK[1:0] | I |
| 66IN | I |
| CHA_HCLKOUT | D |
| CHB_HCLKOUT | D |
| CHA_RCLKOUT | D |
| CHB_RCLKOUT | D |
| RSTIN# | I |
| TESTIN# | I |

This page is intentionally left blank.

# 3 Register Description

The MCH contains two sets of software accessible registers, accessed via the Host I/O address space:

- Control registers I/O mapped into the host I/O space, which control access to PCI and AGP configuration space (see section titled *I/O Mapped Registers*)
- Internal configuration registers residing within the MCH are partitioned into two logical device register sets ("logical" since they reside within a single physical device). The first register set is dedicated to Host-Hub Interface Bridge functionality (controls PCI_A such as DRAM configuration, other chip-set operating parameters and optional features). The second register block is dedicated to Host-AGP Bridge functions (controls AGP interface configurations and operating parameters).

The MCH supports PCI configuration space accesses using the mechanism denoted as Configuration Mechanism #1 in the PCI specification.

The MCH internal registers (I/O Mapped, and Configuration registers) are accessible by the host. The registers can be accessed as Byte, Word (16-bit), or DWord (32-bit) quantities, with the exception of CONF_ADDR, which can only be accessed as a DWord. All multi-byte numeric fields use "little-endian" ordering (i.e., lower addresses contain the least significant parts of the field).

## 3.1 Register Terminology

| Term | Description |
|---|---|
| RO | **Read Only**. If a register is read only, writes to this register have no effect. |
| R/W | **Read/Write**. A register with this attribute can be read and written. |
| R/W/L | **Read/Write/Lock**. A register with this attribute can be read, written, and locked. |
| R/WC | **Read/Write Clear**. A register bit with this attribute can be read and written. However, a write of a 1 clears (sets to 0) the corresponding bit and a write of a 0 has no effect. |
| R/WO | **Read/Write Once.** A register bit with this attribute can be written to only once after power up. After the first write, the bit becomes read only. |
| L | **Lock.** A register bit with this attribute becomes read only after a lock-bit is set. |
| Reserved Bits | Some of the MCH registers described in this section contain reserved bits. These bits are labeled "Reserved". Software must deal correctly with fields that are reserved. On reads, software must use appropriate masks to extract the defined bits and not rely on reserved bits being any particular value. On writes, software must ensure that the values of reserved bit positions are preserved. That is, the values of reserved bit positions must first be read, merged with the new values for other bit positions and then written back. Note that software does not need to perform read, merge, write operation for the Configuration Address (CONF_ADDR) register. |

| Term | Description |
|---|---|
| Reserved Registers | In addition to reserved bits within a register, the MCH contains address locations in the configuration space that are marked "Reserved". When a "Reserved" register location is read, a random value is returned. ("Reserved" registers can be 8-, 16-, or 32-bit in size). Registers that are marked as "Reserved" must not be modified by system software. Writes to "Reserved" registers may cause system failure. |
| Default Value upon a Reset | Upon a Full Reset, the MCH sets all of its internal configuration registers to predetermined **default** states. Some register values at reset are determined by external strapping options. The default state represents the minimum functionality feature set required to successfully bring up the system. Hence, it does not represent the optimal system configuration. It is the responsibility of the system initialization software (usually BIOS) to properly determine the DRAM configurations, operating parameters and optional system features that are applicable, and to program the MCH registers accordingly. |

## 3.2 PCI Configuration Space Access

Hub Interface_A physically connects the MCH and ICH2. From a configuration standpoint, Hub Interface_A is logically PCI bus #0. As a result, all devices internal to the MCH and ICH2 appear to be on PCI bus #0. The system's primary PCI expansion bus is physically attached to the ICH2 and, from a configuration perspective, appears to be a hierarchical PCI bus behind a PCI-to-PCI bridge and therefore has a programmable PCI Bus number. **Note that the primary PCI bus is referred to as PCI_A in this document and is not PCI bus #0 from a configuration standpoint.** The AGP and 16-bit hub interface ports appear to system software to be real PCI busses behind PCI-to-PCI bridges resident as devices on PCI bus #0.

The MCH contains up to four PCI devices within a single physical component. The configuration registers for the six devices are mapped as devices residing on PCI bus #0.

- Device 0: Host-Hub Interface_A Bridge/DRAM Controller. Logically this appears as a PCI device residing on PCI bus #0. Physically Device 0 contains the standard PCI registers, DRAM registers, the Graphics Aperture controller, and other MCH specific registers.
- Device 1: Host-AGP Bridge. Logically this appears as a "virtual" PCI-to-PCI bridge residing on PCI bus #0. Physically Device 1 contains the standard PCI-to-PCI bridge registers and the standard AGP configuration registers (including the AGP I/O and memory address mapping).
- Device 2: Host-Hub Interface_B Bridge. Logically this bridge appears to be a PCI-to-PCI bridge device residing on PCI bus #0. Physically, Device 2 contains the standard PCI-to-PCI registers.
- Device 3: Host-Hub Interface_C Bridge. Logically this bridge appears to be a PCI-to-PCI bridge device residing on PCI bus #0. Physically, Device 3 contains the standard PCI-to-PCI registers.

The following table shows the Device # assignment for the various internal MCH devices:

| MCH Function | Bus #0, Device # | Function # |
|---|---|---|
| DRAM Controller/8-bit Hub Interface_A Controller | Device 0 | Function #0 |
| Host-to-AGP Bridge (virtual P2P) | Device 1 | Function #0 |
| Host-to-16-bit Hub Interface_B Bridge (P2P) | Device 2 | Function #0 |
| Host-to-16-bit Hub Interface_C Bridge (P2P) | Device 3 | Function #0 |

The MCH automatically detects if devices are connected to Hub Interface_B or Hub Interface_C by sampling the HL[11] signal on the rising edge of RSTIN#. When a hub interface is unpopulated, the associated configuration register space is hidden, returning all 1s for all registers just as if the cycle terminated with a Master Abort on PCI. If Hub Interface_C is used, then Hub Interface_B **must** be populated with an Intel P64H.

***Note:*** Physical PCI bus #0 does not exist. The Hub Interface_A and the internal devices in the MCH and ICH2 logically constitute PCI Bus #0 to configuration software.

## Standard PCI Bus Configuration Mechanism

The PCI Bus defines a slot based "configuration space" that allows each device to contain up to eight functions with each function containing up to 256, 8-bit configuration registers. The PCI specification defines two bus cycles 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 specification defines two mechanisms to access configuration space, Mechanism #1 and Mechanism #2. **The MCH supports only Mechanism #1.**

The configuration access mechanism makes use of the CONF_ADDR register and CONF_DATA register. To reference a configuration register a DWord I/O write cycle is used to place a value into CONF_ADDR 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. CONF_ADDR[31] must be 1 to enable a configuration cycle. CONF_DATA then becomes a window into the four bytes of configuration space specified by the contents of CONF_ADDR. Any read or write to CONF_DATA will result in the MCH translating the CONF_ADDR into the appropriate configuration cycle.

The MCH is responsible for translating and routing the processor's I/O accesses to the CONF_ADDR and CONF_DATA registers to internal MCH configuration registers, Hub Interface_A–C or AGP.

## Routing Configuration Accesses

The MCH supports up to four bus interfaces: Hub Interface_A–C, and AGP. PCI configuration cycles are selectively routed to one of these interfaces. The MCH is responsible for routing PCI configuration cycles to the proper interface. PCI configuration cycles to ICH2 internal devices and Primary PCI (including downstream devices) are routed to the ICH2 via Hub Interface_A. PCI configuration cycles to one of the 16-bit hub interfaces are routed to Hub Interface_B–C. AGP configuration cycles are routed to AGP. The AGP interface is treated as a separate PCI bus from the configuration point of view. Routing of configuration accesses to Hub Interface_B–C and AGP is controlled via the standard PCI-PCI bridge mechanism using information contained within the Primary Bus Number, the Secondary Bus Number, and the Subordinate Bus Number registers of the corresponding PCI-PCI bridge device.

### Logical PCI BUS #0 Configuration Mechanism

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

- The Host-Hub Interface_A Bridge entity within the MCH is hardwired as Device 0 on PCI Bus #0.
- The Host-AGP Bridge entity within the MCH is hardwired as Device 1 on PCI Bus #0.
- The Host-Hub Interface_B bridge entity within the MCH is hardwired as Device 2 on PCI Bus #0.
- The Host-Hub Interface_C bridge entity within the MCH is hardwired as Device 3 on PCI Bus #0.

Configuration cycles to any of the MCH's internal devices are confined to the MCH and not sent over Hub Interface_A. Accesses to disabled MCH internal devices, or devices #7 to #31 will be forwarded over Hub Interface_A as Type 0 configuration cycles.

### Primary PCI and Downstream Configuration Mechanism

If the Bus Number in the CONF_ADDR is non-zero, and is less than the values programmed into any of the internal MCH device's Secondary Bus Number registers or greater than the values programmed into the Subordinate Bus Number registers, the MCH will generate a Type 1 Hub Interface_A Configuration Cycle. The ICH2 compares the non-zero Bus Number with the Secondary Bus Number and Subordinate Bus Number registers of its P2P bridges to determine if the configuration cycle is meant for Primary PCI or a downstream PCI bus.

### AGP Bus Configuration Mechanism

From the chipset configuration perspective, AGP is seen as PCI bus interfaces residing on a Secondary Bus side of the "virtual" PCI-PCI bridges referred to as the MCH Host-AGP bridge. On the Primary bus side, the "virtual" PCI-PCI bridge is attached to PCI Bus #0. Therefore, the Primary Bus Number register is hardwired to 0. The "virtual" PCI-PCI bridge entity converts Type #1 PCI Bus Configuration cycles on PCI Bus #0 into Type 0 or Type 1 configuration cycles on the AGP interface. Type 1 configuration cycles on PCI Bus #0 that have a bus number that matches the Secondary Bus Number of one of the MCH's "virtual" P2P bridges will be translated into Type 0 configuration cycles on the AGP interface.

If the Bus Number is non-zero, greater than the value programmed into the Secondary Bus Number register, and less than or equal to the value programmed into the Subordinate Bus Number register, the MCH will generate a Type 1 PCI configuration cycle on AGP.

## 3.3 I/O Mapped Registers

The MCH contains a set of registers that reside in the Host I/O address space – the Configuration Address (CONF_ADDR) register and the Configuration Data (CONF_DATA) register. The Configuration Address register enables/disables the configuration space and determines what portion of configuration space is visible through the Configuration Data window.

### 3.3.1 CONF_ADDR—Configuration Address Register

| | |
|---|---|
| I/O Address: | 0CF8h Accessed as a DWord |
| Default Value: | 00000000h |
| Access: | Read/Write |
| Size: | 32 bits |

CONF_ADDR is a 32-bit register that can be accessed only as a DWord. A Byte or Word reference will “pass through” the Configuration Address register and Hub Interface_A onto the PCI bus as an I/O cycle. The CONF_ADDR register contains the Bus Number, Device Number, Function Number, and Register Number for which a subsequent configuration access is intended.

| Bit | Descriptions |
|---|---|
| 31 | **Configuration Enable (CFGE).**<br>1 = Enable<br>0 = Disable |
| 30:24 | **Reserved.** These bits are read only and have a value of 0. |
| 23:16 | **Bus Number.** When the Bus Number is programmed to 00h the target of the Configuration Cycle is a hub interface agent (MCH, ICH2, etc.).<br>The Configuration Cycle is forwarded to Hub Interface_A if the Bus Number is programmed to 00h and the MCH is not the target (the device number is ≥ 4).<br>If the Bus Number is non-zero and matches the value programmed into the Secondary Bus Number register of Device 1, a Type 0 PCI configuration cycle will be generated on AGP.<br>If the Bus Number is non-zero, greater than the value in the Secondary Bus Number register of Device 1 and less than or equal to the value programmed into the Subordinate Bus Number register of Device 1 a Type 1 PCI configuration cycle will be generated on AGP.<br>If the Bus Number is non-zero and matches the value programmed into the Secondary Bus Number register of Device 2–3 a Type 0 PCI configuration cycle will be generated on the corresponding Hub Interface_B–C.<br>If the Bus Number is non-zero, and less than or equal to the value programmed into the Subordinate Bus Number register of Device 2–3 a Type 1 PCI configuration cycle will be generated on the corresponding Hub Interface_B–C.<br>If the Bus Number is non-zero, and does not fall within the ranges enumerated by Device 1–3’s Secondary Bus Number or Subordinate Bus Number register, then a Hub Interface_A Type 1 Configuration Cycle is generated. |

| Bit | Descriptions |
|---|---|
| 15:11 | **Device Number.** This field selects one agent on the PCI bus selected by the Bus Number. When the Bus Number field is “00” the MCH decodes the Device Number field. The MCH is always Device Number 0 for the Host-Hub Interface_A bridge entity, Device Number 1 for the Host-AGP entity, and Device Number 2–3 for the Host-Hub Interface_B–C entities respectively. Therefore, when the Bus Number = 0 and the Device Number = 0–3, the internal MCH devices are selected.<br>If the Bus Number is non-zero and matches the value programmed into the Secondary Bus Number register, a Type 0 PCI configuration cycle will be generated on AGP. The MCH decodes the Device Number field [15:11] and asserts the appropriate GAD signal as an IDSEL. For PCI-to-PCI Bridge translation, one of the 16 IDSELs is generated. When bit 15 = 0, bits [14:11] are decoded to assert a signal AD[31:16] IDSEL. GAD16 is asserted to access Device 0, GAD17 for Device 1, and so forth up to Device 15 for which will assert AD31. All device numbers higher than 15 cause a type 0 configuration access with no IDSEL asserted, which will result in a Master Abort reported in the MCH’s “virtual” PCI-PCI bridge registers.<br>For Bus Numbers resulting in Hub Interface_A–C configuration cycles, the MCH propagates the device number field as A[15:11]. For bus numbers resulting in AGP/PCI_B type 1 configuration cycles, the device number is propagated as GAD[15:11]. |
| 10:8 | **Function Number.** This field is mapped to GAD[10:8] during AGP configuration cycles and A[10:8] during Hub Interface_A–C configuration cycles. This allows the configuration registers of a particular function in a multi-function device to be accessed. The MCH ignores configuration cycles to its internal devices if the function number is not equal to 0. |
| 7:2 | **Register Number**. This field selects one register within a particular bus, device, and function as specified by the other fields in the Configuration Address register. This field is mapped to GAD[7:2] during AGP configuration cycles and A[7:2] during Hub Interface_A–C configuration cycles. |
| 1:0 | **Reserved** |

## 3.3.2 CONF_DATA—Configuration Data Register

| | |
|---|---|
| I/O Address: | 0CFCh |
| Default Value: | 00000000h |
| Access: | Read/Write |
| Size: | 32 bits |

CONF_DATA is a 32-bit Read/Write window into configuration space. The portion of configuration space that is referenced by CONF_DATA is determined by the contents of CONF_ADDR.

| Bit | Descriptions |
|---|---|
| 31:0 | **Configuration Data Window (CDW).** If bit 31 of CONF_ADDR is 1, any I/O access to the CONF_DATA register will be mapped to configuration space using the contents of CONF_ADDR. |

## 3.4 Host-Hub Interface_A Bridge Device Registers (Device 0)

Table 6 shows the address map and describes the access attributes for the Device 0 configuration space. An **"s"** in the Default Value field means that a strap determines the power-up default value for that bit.

### Table 6. MCH Configuration Space (Device 0)

| Address Offset | Register Symbol | Register Name | Default Value | Access |
|---|---|---|---|---|
| 00–01h | VID | Vendor Identification | 8086h | RO |
| 02–03h | DID | Device Identification | 2531h | RO |
| 04–05h | PCICMD | PCI Command Register | 0006h | RO, R/W |
| 06–07h | PCISTS | PCI Status Register | 0090h | RO, R/WC |
| 08h | RID | Revision Identification | 04h | RO |
| 0Ah | SUBC | Sub-Class Code | 00h | RO |
| 0Bh | BCC | Base Class Code | 06h | RO |
| 0Dh | MLT | Master Latency Timer | 00h | RO |
| 0Eh | HDR | Header Type | 00h | RO |
| 10–13h | APBASE | Aperture Base Configuration | 00000008h | RO, R/W |
| 2C–2Dh | SVID | Subsystem Vendor Identification | 0000h | R/WO |
| 2E–2Fh | SID | Subsystem Identification | 0000h | R/WO |
| 34h | CAPPTR | Capabilities Pointer | A0h | RO |
| 40–4Fh | GAR[0:15] | RDRAM* Device Group Architecture Register [0:15] | 80h | RO, R/W |
| 50–51h | MCHCFG | MCH Configuration | 0000000000000s00b | RO, R/W |
| 52–57h | — | Reserved | — | — |
| 58h | FDHC | Fixed DRAM Hole Control | 00h | RO, R/W |
| 59–5Fh | PAM[0:6] | Programmable Attribute Map [0:6] | 00h | RO, R/W |
| 60–61h | GBA0 | RDRAM Device Group Boundary Address 0 | 0001h | RO, R/W |
| 62–63h | GBA1 | RDRAM Device Group Boundary Address 1 | 0001h | RO, R/W |
| 64–65h | GBA2 | RDRAM Device Group Boundary Address 2 | 0001h | RO, R/W |
| 66–67h | GBA3 | RDRAM Device Group Boundary Address 3 | 0001h | RO, R/W |
| 68–69h | GBA4 | RDRAM Device Group Boundary Address 4 | 0001h | RO, R/W |
| 6A–6Bh | GBA5 | RDRAM Device Group Boundary Address 5 | 0001h | RO, R/W |
| 6C–6Dh | GBA6 | RDRAM Device Group Boundary Address 6 | 0001h | RO, R/W |
| 6E–6Fh | GBA7 | RDRAM Device Group Boundary Address 7 | 0001h | RO, R/W |

| Address Offset | Register Symbol | Register Name | Default Value | Access |
|---|---|---|---|---|
| 70–71h | GBA8 | RDRAM Device Group Boundary Address 8 | 0001h | RO, R/W |
| 72–73h | GBA9 | RDRAM Device Group Boundary Address 9 | 0001h | RO, R/W |
| 74–75h | GBA10 | RDRAM Device Group Boundary Address A | 0001h | RO, R/W |
| 76–77h | GBA11 | RDRAM Device Group Boundary Address B | 0001h | RO, R/W |
| 78–79h | GBA12 | RDRAM Device Group Boundary Address C | 0001h | RO, R/W |
| 7A–7Bh | GBA13 | RDRAM Device Group Boundary Address D | 0001h | RO, R/W |
| 7C–7Dh | GBA14 | RDRAM Device Group Boundary Address E | 0001h | RO, R/W |
| 7E–7Fh | GBA15 | RDRAM Device Group Boundary Address F | 0001h | RO, R/W |
| 80–87h | — | Reserved | — | — |
| 88h | RDPS | RDRAM Device Pool Sizing Register | 10h | RO, WO, L |
| 90–93h | DRD | RDRAM Device Register Data | 00000000h | R/W |
| 94–97h | RICM | RDRAM Device Initialization Control Management | 00000000h | RO, R/W |
| 98–9Bh | — | Reserved | — | — |
| 9Dh | SMRAM | System Management RAM Control | 02h | RO, R/W, L |
| 9Eh | ESMRAMC | Extended System Management RAM Control | 38h | RO, R/W, R/WC, L |
| 9Fh | — | Reserved | — | — |
| A0–A3h | ACAPID | AGP Capability Identifier | 00200002h | RO |
| A4–A7h | AGPSTAT | AGP Status Register | 1F000217h | RO |
| A8–ABh | AGPCMD | AGP Command Register | 00000000h | RO, R/W |
| B0–B3h | AGPCTRL | AGP Control Register | 00000000h | RO, R/W |
| B4h | APSIZE | Aperture Size | 00h | RO, R/W |
| B8–BBh | ATTBASE | Aperture Translation Table | 00000000h | RO, R/W |
| BCh | AMTT | AGP MTT Control Register | 00h | RO, R/W |
| BDh | LPTT | AGP Low Priority Transaction Timer Register | 00h | RO, R/W |
| BEh | RDT | RDRAM Device Timing | 00h | R/W |
| BF–C3h | — | Reserved | — | — |
| C4–C5h | TOM | Top of Low Memory Register | 0000h | R/W |
| C6–C7h | — | Reserved | — | — |
| C8–C9h | ERRSTS | Error Status Register | 0000h | R/WC |
| CA–CBh | ERRCMD | Error Command Register | 0000h | R/W |
| CC–CDh | SMICMD | SMI Command Register | 0000h | RO, R/W |
| CE–CFh | SCICMD | SCI Command Register | 0000h | RO, R/W |
| D0–DBh | — | Reserved | — | — |
| DC–DDh | DRAMRC | RDRAM Device Refresh Control | 0000h | RO, R/W |

| Address Offset | Register Symbol | Register Name | Default Value | Access |
|---|---|---|---|---|
| DE–DFh | SKPD | Scratchpad Data | 0000h | R/W |
| E2–E3 | DERRCTL | DRAM Error Control Register | 0000h | RO |
| E4–E7h | EAP | DRAM Error Data Register | 00000000h | RO |
| E8–F3h | — | Reserved | — | — |
| F4–F7h | MISC_CNTL | Miscellaneous Control Register | 0000F874h | R/W |
| F8–FFh | — | Reserved | — | — |

## 3.4.1 VID—Vendor Identification Register (Device 0)

| | |
|---|---|
| Address Offset: | 00–01h |
| Default Value: | 8086h |
| Attribute: | Read Only |
| Size: | 16 bits |

The VID register contains the vendor identification number. This 16-bit register, combined with the Device Identification register, uniquely identifies any PCI device. Writes to this register have no effect.

| Bit | Description |
|---|---|
| 15:0 | **Vendor Identification Number.** This is a 16-bit value assigned to Intel. Intel VID = 8086h. |

## 3.4.2 DID—Device Identification Register (Device 0)

| | |
|---|---|
| Address Offset: | 02–03h |
| Default Value: | 2531h |
| Attribute: | Read Only |
| Size: | 16 bits |

This 16-bit register combined with the Vendor Identification register uniquely identifies any PCI device. Writes to this register have no effect.

| Bit | Description |
|---|---|
| 15:0 | **Device Identification Number.** This is a 16-bit value assigned to the MCH Host-Hub Interface_A Bridge Function #0. |

### 3.4.3 PCICMD—PCI Command Register (Device 0)

| | |
|---|---|
| Address Offset: | 04–05h |
| Default: | 0006h |
| Access: | Read/Write, Read Only |
| Size | 16 bits |

Since MCH Device 0 does not physically reside on PCI0 many of the bits are not implemented. Writes to bits that are not implemented have no affect.

| Bit | Descriptions |
|---|---|
| 15:10 | **Reserved** |
| 9 | **Fast Back-to-Back—RO.** Not implemented; hardwired to 0. This bit controls whether or not the master can do fast back-to-back write. Since Device 0 is strictly a target this bit is not implemented. |
| 8 | **SERR Enable (SERRE)—R/W.** This bit is a global enable bit for Device 0 SERR messaging. The MCH does not have an SERR# signal. The MCH communicates the SERR# condition by sending an SERR message to the ICH2.<br>1 = Enable. MCH is enabled to generate SERR messages over Hub Interface_A for specific Device 0 error conditions that are individually enabled in the ERRCMD register. The error status is reported in the ERRSTS and PCISTS registers.<br>0 = Disable.<br>**NOTE:** This bit only controls SERR message for the Device 0. Devices 1-5 have their own SERRE bits to control error reporting for error conditions occurring on their respective devices. |
| 7 | **Address/Data Stepping—RO.** Not implemented; hardwired to 0. |
| 6 | **Parity Error Enable (PERRE)—R/W.**<br>1 = MCH will generate an SERR message over Hub Interface_A to the ICH2 when an address or data parity error is detected by the MCH on Hub Interface_A (DPE set in PCISTS).<br>0 = MCH does not take any action when it detects a parity error on Hub Interface_A. |
| 5 | **VGA Palette Snoop—RO.** Not implemented; hardwired to 0. |
| 4 | **Memory Write and Invalidate Enable (MWIE)—RO.** Not implemented; hardwired to 0. |
| 3 | **Special Cycle Enable(SCE)—RO.** Not implemented; hardwired to 0. |
| 2 | **Bus Master Enable (BME)—RO.** Not implemented; hardwired to 1. The MCH is always enabled as a master on Hub Interface_A. |
| 1 | **Memory Access Enable (MAE)—RO.** Not implemented; hardwired to 1.The MCH always allows access to main memory. |
| 0 | **I/O Access Enable (IOAE)—RO.** Not implemented; hardwired to 0. |

### 3.4.4 PCISTS—PCI Status Register (Device 0)

| | |
|---|---|
| Address Offset: | 06–07h |
| Default Value: | 0090h |
| Access: | Read Only, Read/Write Clear |
| Size: | 16 bits |

PCISTS is a 16-bit status register that reports the occurrence of error events on devices on the hub interface (Device 0s). Since MCH Device 0 is the Host-to-Hub Interface_A bridge, many of the bits are not implemented.

| Bit | Description |
|---|---|
| 15 | **Detected Parity Error (DPE)—R/WC**.<br>1 = MCH detects a parity error on Hub Interface_A.<br>0 = Software clears this bit by writing a 1 to it. |
| 14 | **Signaled System Error (SSE)—R/WC**.<br>1 = Device 0 generates an SERR message over Hub Interface_A for any enabled Device 0 error condition. Device 0 error conditions are enabled in the PCICMD and ERRCMD registers. Device 0 error flags are read/reset from the PCISTS or ERRSTS registers.<br>0 **=** Software sets SSE to 0 by writing a 1 to this bit. |
| 13 | **Received Master Abort Status (RMAS)—R/WC**.<br>1 = MCH generates a Hub Interface_A request that receives a Master Abort completion packet or Master Abort Special Cycle.<br>0 **=** Software sets SSE to 0 by writing a 1 to this bit. |
| 12 | **Received Target Abort Status (RTAS)—R/WC.**<br>1 = MCH generates a Hub Interface_A request that receives a Target Abort completion packet or Target Abort Special Cycle.<br>0 **=** Software sets SSE to 0 by writing a 1 to this bit. |
| 11 | **Signaled Target Abort Status (STAS)—RO.** Not implemented; hardwired to 0. The MCH will not generate a Target Abort Hub Interface_A completion packet or Special Cycle. |
| 10:9 | **DEVSEL Timing (DEVT)**. Hardwired to 00. Hub interface does not comprehend DEVSEL# protocol. |
| 8 | **Master Data Parity Error Detected (DPD)—RO.** Hardwired to 0. PERR signaling and messaging are not implemented by the MCH. |
| 7 | **Fast Back-to-Back (FB2B)—RO**. Hardwired to 1. |
| 6:5 | **Reserved** |
| 4 | **Capability List (CLIST)—RO.**<br>1 = Indicates to the configuration software that this device/function implements a list of new capabilities. A list of new capabilities is accessed via register CAPPTR at configuration address offset 34h. Register CAPPTR contains an offset pointing to the start address within configuration space of this device where the AGP Capability standard register resides. |
| 3:0 | **Reserved** |

## 3.4.5 RID—Revision Identification Register (Device 0)

| | |
|---|---|
| Address Offset: | 08h |
| Default Value: | 04h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the revision number of the MCH Device 0. These bits are read only and writes to this register have no effect.

| Bit | Description |
|---|---|
| 7:0 | **Revision Identification Number.** This is an 8-bit value that indicates the revision identification number for the MCH Device 0.<br>A-3 Stepping = 04h. |

## 3.4.6 SUBC—Sub-Class Code Register (Device 0)

| | |
|---|---|
| Address Offset: | 0Ah |
| Default Value: | 00h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the Sub-Class Code for the MCH Device 0.

| Bit | Description |
|---|---|
| 7:0 | **Sub-Class Code (SUBC)**. This is an 8-bit value that indicates the category of bridge for the MCH.<br>00h = Host Bridge. |

## 3.4.7 BCC—Base Class Code Register (Device 0)

| | |
|---|---|
| Address Offset: | 0Bh |
| Default Value: | 06h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the Base Class Code of the MCH Device 0.

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BASEC)**. This is an 8-bit value that indicates the Base Class Code for the MCH.<br>06h = Bridge device. |

### 3.4.8 MLT—Master Latency Timer Register (Device 0)

| | |
|---|---|
| Address Offset: | 0Dh |
| Default Value: | 00h |
| Access: | Read Only |
| Size: | 8 bits |

The hub interface does not comprehend the concept of Master Latency Timer. Therefore, this register is not implemented.

| Bit | Description |
|---|---|
| 7:0 | These bits are hardwired to 0. Writes have no effect. |

### 3.4.9 HDR—Header Type Register (Device 0)

| | |
|---|---|
| Offset: | 0Eh |
| Default: | 00h |
| Access: | Read Only |
| Size: | 8 bits |

This register identifies the header layout of the configuration space. No physical register exists at this location.

| Bit | Description |
|---|---|
| 7:0 | This read only field always returns 0s when read and writes have no affect. |

### 3.4.10 APBASE—Aperture Base Configuration Register (Device 0)

| | |
|---|---|
| Offset: | 10–13h |
| Default: | 00000008h |
| Access: | Read/Write, Read Only |
| Size: | 32 bits |

The APBASE is a standard PCI Base Address register that is used to set the base of the Graphics Aperture. The standard PCI Configuration mechanism defines the base address configuration register such that only a fixed amount of space can be requested (dependent on which bits are hardwired to 0 or behave as hardwired to 0). To allow for flexibility (of the aperture), an additional register called APSIZE is used as a “back-end” register to control which bits of the APBASE will behave as hardwired to 0. This register will be programmed by the MCH specific BIOS code that will run before any of the generic configuration software is run.

***Note:*** Bit 9 of the MCHCFG register is used to prevent accesses to the aperture range before configuration software initializes this register and the appropriate translation table structure has been established in the main memory.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>31:28</td><td><b>Upper Programmable Base Address bits—R/W.</b> These bits are used to locate the range size selected via lower bits 27:4.<br>Default = 0000</td></tr>
<tr><td>27:22</td><td><b>Lower “Hardwired”/Programmable Base Address bits—R/W.</b> These bits behave as a “hardwired” or as programmable depending on the contents of the APSIZE register as defined below:
<table>
<tr><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>Aperture Size</th></tr>
<tr><td>r/w</td><td>r/w</td><td>r/w</td><td>r/w</td><td>r/w</td><td>r/w</td><td>4 MB</td></tr>
<tr><td>r/w</td><td>r/w</td><td>r/w</td><td>r/w</td><td>r/w</td><td>0</td><td>8 MB</td></tr>
<tr><td>r/w</td><td>r/w</td><td>r/w</td><td>r/w</td><td>0</td><td>0</td><td>16 MB</td></tr>
<tr><td>r/w</td><td>r/w</td><td>r/w</td><td>0</td><td>0</td><td>0</td><td>32 MB</td></tr>
<tr><td>r/w</td><td>r/w</td><td>0</td><td>0</td><td>0</td><td>0</td><td>64 MB</td></tr>
<tr><td>r/w</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>128 MB</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>256 MB</td></tr>
</table>
Bits 27:22 are controlled by the bits 5:0 of the APSIZE register in the following manner:
<ul>
<li>If bit APSIZE[5]=0, APBASE[27]=0. If APSIZE[5]=1, APBASE[27]=R/W. The same applies correspondingly to other bits.</li>
<li>Default for APSIZE[5:0]=000000b forces default APBASE[27:22] =000000b (i.e., all bits respond as hardwired to 0). This provides a default to the maximum aperture size of 256 MB. The MCH specific BIOS is responsible for selecting smaller size (if required) before PCI configuration software runs and establishes the system address map.</li>
</ul></td></tr>
<tr><td>21:4</td><td><b>Hardwired to 0.</b> This forces minimum aperture size selected by this register to be 4 MB.</td></tr>
<tr><td>3</td><td><b>Prefetchable—RO.</b> This bit is hardwired to 1 to identify the graphics aperture range as prefetchable (i.e., there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables, and the MCH may merge processor writes into this range without causing errors).</td></tr>
<tr><td>2:1</td><td><b>Type—RO.</b> These bits determine addressing type and they are hardwired to 00 to indicate that the address range defined by the upper bits of this register can be located anywhere in the 32-bit address space.</td></tr>
<tr><td>0</td><td><b>Memory Space Indicator—RO.</b> Hardwired to 0 to identify aperture range as a memory range.</td></tr>
</table>

### 3.4.11 SVID—Subsystem Vendor ID Register (Device 0)

| | |
|---|---|
| Offset: | 2C–2Dh |
| Default: | 0000h |
| Access: | Read/Write Once |
| Size: | 16 bits |

This value is used to identify the vendor of the subsystem.

| Bit | Description |
|---|---|
| 15:0 | **Subsystem Vendor ID.** The default value is 00h. This field should be programmed during boot-up. After this field is written once, it becomes read only. |

## 3.4.12 SID—Subsystem ID Register (Device 0)

| | |
|---|---|
| Offset: | 2E–2Fh |
| Default: | 0000h |
| Access: | Read/Write Once |
| Size: | 16 bits |

This value is used to identify a particular subsystem.

| Bit | Description |
|---|---|
| 15:0 | **Subsystem ID (R/WO).** The default value is 00h. This field should be programmed during boot-up. After this field is written once, it becomes read only. |

## 3.4.13 CAPPTR—Capabilities Pointer Register (Device 0)

| | |
|---|---|
| Offset: | 34h |
| Default: | AGP–A0h |
| Access: | Read Only |
| Size: | 8 bits |

The CAPPTR provides the offset that is the pointer to the location where the AGP standard registers are located.

| Bit | Description |
|---|---|
| 7:0 | **Pointer to the start of AGP standard register block.** This pointer indicates where software can find the beginning of the AGP register block. The value in this field is A0h when the AGP interface is configured for AGP mode. |

## 3.4.14 GAR[0:15]—RDRAM* Device Group Architecture Register (Device 0)

| | |
|---|---|
| Address Offset: | 40–4Fh |
| Default Value: | 80h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits/register |

This 8-bit register defines the #of banks and DRAM technology of each device group in the Rambus Channel. There are 16 GAR registers (GAR0–GAR15) that are used to define 16 groups for the Rambus Channel.

| Bit | Description |
|---|---|
| 7:6 | **Device Page Size (DPS).** This field defines the page size of the each device in the corresponding group.<br>00 = Reserved<br>01 = Reserved<br>10 = 1 KB<br>11 = 2 KB |
| 5 | **Reserved** |
| 4 | **Device Banks (DB).** This field defines the number of bank architecture in each device in the group.<br>0 = 16 dependent Banks<br>1 = 32 dependent Banks arranged in two groups of 16 dependent banks (i.e., 2x16) |
| 3 | **Reserved** |
| 2:1 | **Device DRAM Technology (DDT).** This field defines the DRAM technology of each device in the group.<br>00 = Reserved<br>01 = 128/144Mbit<br>10 = 256/288Mbit<br>11 = Reserved |
| 0 | **Reserved** |

### 3.4.15 MCHCFG—MCH Configuration Register (Device 0)

| | |
|---|---|
| Offset: | 50–51h |
| Default: | 0000_0000_ 0000_0s00b |
| Access: | Read/Write Once, Read/Write, Read Only |
| Size: | 16 bits |

| Bit | Description |
|---|---|
| 15:13 | **Number of Stop Grant Cycles—R/W.** Number of Stop Grant transactions expected on the host bus before a Stop Grant Acknowledge packet is sent to the ICH2. This field is programmed by the BIOS after it has enumerated the processors and before it has enabled Stop Clock generation in the ICH2. Note that each enabled thread within each processor will generate Stop Grant Acknowledge transactions.<br>**Bits[15:13]** **HL_A Stop Grant Generated After**<br>000 1 System Bus Stop Grant (default)<br>001 2 System Bus Stop Grants<br>010 3 System Bus Stop Grants<br>011 4 System Bus Stop Grants<br>100 5 System Bus Stop Grants<br>101 6 System Bus Stop Grants<br>110 7 System Bus Stop Grants<br>111 8 System Bus Stop Grants |
| 12 | **Reserved** |
| 11 | **Direct RDRAM* Device Frequency—R/W.** These bits are written by the BIOS after polling the Direct RDRAM devices and finding the least common denominator speed.<br>0 = 300 MHz (default)<br>1 = 400 MHz |
| 10 | **Reserved** |
| 9 | **Aperture Access Global Enable—R/W.** This bit is used to prevent access to the graphics aperture from any port (host, Hub Interface_A, Hub Interface_B, Hub Interface_C, or AGP) before configuration software establishes the aperture range and appropriate translation table in the main DRAM has been initialized. It must be set after system is fully configured for aperture accesses.<br>1 = Enable<br>0 = Disable (default) |
| 8:7 | **DRAM Data Integrity Mode (DDIM)—R/W.** These bits select one of two DRAM data integrity modes.<br>00 = Non-ECC (Byte-Wise Writes supported, RDRAM device only) (default)<br>01 = Reserved<br>10 = ECC Mode (Generation and Error Checking/Correction)<br>11 = Reserved |
| 6 | **Reserved** |

| Bit | Description |
|---|---|
| 5 | **MDA Present (MDAP)—R/W.** This bit works with the VGA enable bit in the BCTRL register of Device 1–3 to control the routing of host-initiated transactions targeting MDA compatible I/O and memory address ranges. This bit should not be set when the VGA enable bit is not set in either Device 1–3. If the VGA enable bit is set, accesses to IO address range x3BCh–x3BFh are forwarded to Hub Interface_A. MDA resources are defined as follows:<br>Memory: 0B0000h–0B7FFFh<br>I/O: 3B4h, 3B5h, 3B8h, 3B9h, 3BAh, 3BFh, (including ISA address aliases, A[15:10] are not used in decode)<br>Any I/O reference that includes the I/O locations listed above, or their aliases, will be forwarded to the Hub Interface_A even if the reference includes I/O locations not listed above.<br>Refer to the *System Address Map* section of this document for further information. |
| 4:3 | **Reserved** |
| 2 | **In-Order Queue Depth (IOQD)—RO.** This bit reflects the value sampled on HA7# on the deassertion of the CPURST#. It indicates the depth of the host bus in-order queue (i.e., level of host bus pipelining). If IOQD is set to 1 (i.e., HA7# sampled 1, undriven on the host bus), then the depth of the host bus in-order queue is configured to the maximum allowed by the host bus protocol (i.e., 12). Note that the MCH has an 8-deep IOQ and asserts BNR# on the bus to limit the number of queued bus transactions to 8. If the IOQD bit is set to 0 (HA7# is sampled 0, asserted), then the depth of the host bus in-order queue is set to 1 (i.e., no pipelining support on the host bus).<br>**Note:** HA7# is not driven by the MCH during CPURST#. If an IOQ size of 1 is desired, HA7# must be driven low during CPURST# by an external source. |
| 1 | **APIC Memory Range Disable (APICDIS)—RW.**<br>1 = Disable. The MCH forwards accesses to the IOAPIC regions to the appropriate interface, as specified by the memory and PCI configuration registers.<br>0 = Enable. The MCH sends cycles between 0_FEC0_0000 and 0_FEC7_FFFF to Hub Interface_A; accesses between 0_FEC8_0000 and 0_FEC8_0FFF are sent to Hub Interface_B; accesses between 0_FEC8_1000 and 0_FEC8_1FFF are sent to Hub Interface_C. |
| 0 | **Reserved** |

## 3.4.16 FDHC—Fixed DRAM Hole Control Register (Device 0)

| | |
|---|---|
| Address Offset: | 58h |
| Default Value: | 00h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This 8-bit register controls a fixed DRAM hole.

| Bit | Description |
|---|---|
| 7 | **Hole Enable (HEN).** This field enables a memory hole in DRAM space. Host cycles matching an enabled hole are passed on to ICH2 through the hub interface. The hub interface cycles matching an enabled hole will be ignored by the MCH. Note that a selected hole is not re-mapped.<br>0 = Disable (No hole)<br>1 = Enable. Hole at 15 MB–16 MB (1 MB) |
| 6:0 | **Reserved.** |

## 3.4.17 PAM[0:6]—Programmable Attribute Map Registers (Device 0)

| | |
|---|---|
| Address Offset: | 59–5Fh |
| Default Value: | 00h |
| Attribute: | Read/Write, Read Only |
| Size: | 8 bits |

The MCH allows programmable memory attributes on 13 legacy memory segments of various sizes in the 640-KB to 1-MB address range. Seven Programmable Attribute Map (PAM) Registers are used to support these features. Cacheability of these areas is controlled via the MTRR registers in the processor. Two bits are used to specify memory attributes for each memory segment. These bits apply to host initiator only access to the PAM areas. MCH will forward to main memory for any AGP, PCI or Hub Interface_A–C initiated accesses to the PAM areas. These attributes are:

- **RE** (**Read Enable**). When RE = 1, the host 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 PCI0.
- **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 PCI0.

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.

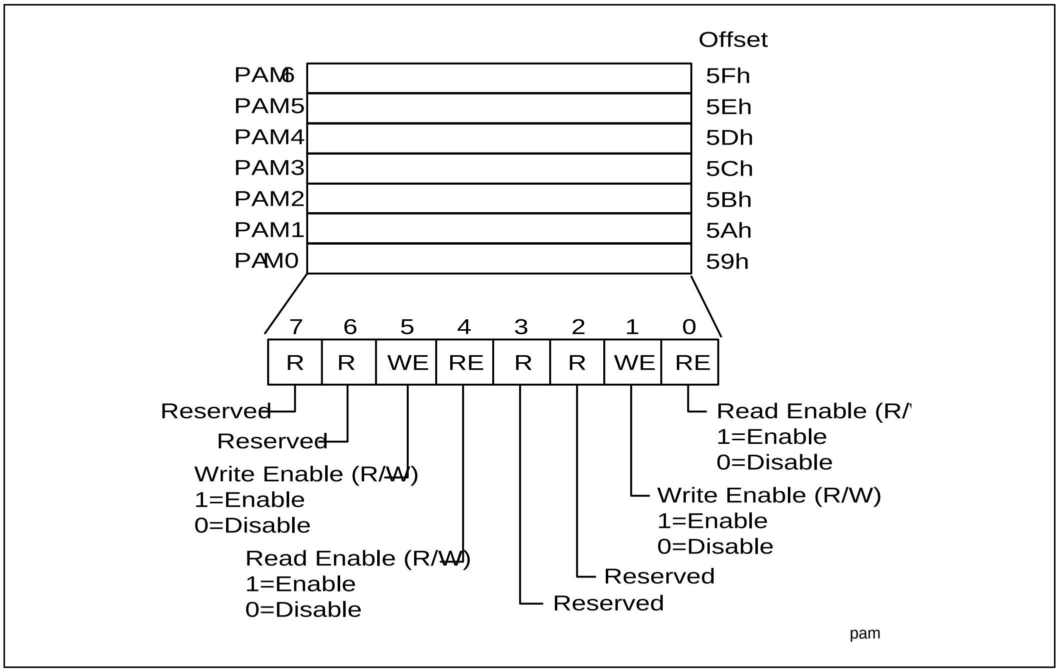
Each PAM register controls two regions, typically 16 KB in size. Each of these regions has a 4-bit field. The four bits that control each region have the same encoding and are defined in the following table.

| Bits [7, 3] Reserved | Bits [6, 2] Reserved | Bits [5, 1] WE | Bits [4, 0] RE | Description |
|---|---|---|---|---|
| X | X | 0 | 0 | **Disabled.** DRAM is disabled and all accesses are directed to the Hub Interface_A. The MCH does not respond as a PCI target for any read or write access to this area. |
| X | X | 0 | 1 | **Read Only**. Reads are forwarded to DRAM and writes are forwarded to the Hub Interface_A for termination. This write protects the corresponding memory segment. The MCH will respond as an AGP or the Hub Interface_A target for read accesses but not for any write accesses. |
| X | X | 1 | 0 | **Write Only**. Writes are forwarded to DRAM and reads are forwarded to the hub interface for termination. The MCH will respond as an AGP or Hub Interface_A target for write accesses but not for any read accesses. |
| X | X | 1 | 1 | **Read/Write**. This is the normal operating mode of main memory. Both read and write cycles from the host are claimed by the MCH and forwarded to DRAM. The MCH will respond as an AGP or the Hub Interface_A target for both read and write accesses. |

At the time that a HI or AGP accesses to the PAM region may occur, the targeted PAM segment must be programmed to be both readable and writeable.

As an example, consider BIOS that is implemented on the expansion bus. During the initialization process, BIOS can be shadowed in main memory to increase the system performance. When BIOS is shadowed in main memory, it should be copied to the same address location. To shadow BIOS, the attributes for that address range should be set to write only. BIOS is shadowed by first doing a read of that address. This read is forwarded to the expansion bus. The host then does a write of the same address, which is directed to main memory. After BIOS is shadowed, the attributes for that memory area are set to read only so that all writes are forwarded to the expansion bus. Table 7 and Figure 3 show the PAM registers and the associated attribute bits.

### Figure 3. PAM Register Attribute Bits



### Table 7. PAM Registers

| PAM Reg | Attribute Bits | | | | Memory Segment | Comments | Offset |
|---|---|---|---|---|---|---|---|
| PAM0[3:0] | Reserved | | | | | | 59h |
| PAM0[7:4] | R | R | WE | RE | 0F0000h–0FFFFFh | BIOS Area | 59h |
| PAM1[3:0] | R | R | WE | RE | 0C0000h–0C3FFFh | ISA Add-on BIOS | 5Ah |
| PAM1[7:4] | R | R | WE | RE | 0C4000h–0C7FFFh | ISA Add-on BIOS | 5Ah |
| PAM2[3:0] | R | R | WE | RE | 0C8000h–0CBFFFh | ISA Add-on BIOS | 5Bh |
| PAM2[7:4] | R | R | WE | RE | 0CC000h–0CFFFFh | ISA Add-on BIOS | 5Bh |
| PAM3[3:0] | R | R | WE | RE | 0D0000h–0D3FFFh | ISA Add-on BIOS | 5Ch |
| PAM3[7:4] | R | R | WE | RE | 0D4000h–0D7FFFh | ISA Add-on BIOS | 5Ch |
| PAM4[3:0] | R | R | WE | RE | 0D8000h–0DBFFFh | ISA Add-on BIOS | 5Dh |
| PAM4[7:4] | R | R | WE | RE | 0DC000h–0DFFFFh | ISA Add-on BIOS | 5Dh |
| PAM5[3:0] | R | R | WE | RE | 0E0000h–0E3FFFh | BIOS Extension | 5Eh |
| PAM5[7:4] | R | R | WE | RE | 0E4000h–0E7FFFh | BIOS Extension | 5Eh |
| PAM6[3:0] | R | R | WE | RE | 0E8000h–0EBFFFh | BIOS Extension | 5Fh |
| PAM6[7:4] | R | R | WE | RE | 0EC000h–0EFFFFh | BIOS Extension | 5Fh |

For details on overall system address mapping scheme see the *System Address Map* section of this document.

- **DOS Application Area (00000h–9FFFh):** The DOS area is 640 KB and is divided into two parts. The 512 KB area at 0 to 7FFFFh is always mapped to the main memory controlled by the MCH. The 128 KB address range from 080000 to 09FFFFh can be mapped to PCI0 or to main memory. By default this range is mapped to main memory and can be declared as a main memory hole (accesses forwarded to PCI0) via MCH FDHC configuration register.
- **Video Buffer Area (A0000h–BFFFFh):** Attribute bits do not control this 128 KB area. The host -initiated cycles in this region are always forwarded to either PCI0 or AGP unless this range is accessed in SMM mode. Routing of accesses is controlled by the Legacy VGA control mechanism of the “virtual” PCI-PCI bridge device embedded within the MCH.

  This area can be programmed as SMM area via the SMRAM register. When used as a SMM space, this range cannot be accessed from the hub interface or AGP.
- **Expansion Area (C0000h–DFFFFh):** This 128 KB area is divided into eight 16 KB segments that can be assigned with different attributes via PAM control register as defined by Table 7.
- **Extended System BIOS Area (E0000h–EFFFFh):** This 64 KB area is divided into four 16 KB segments that can be assigned with different attributes via PAM control register as defined by Table 7.
- **System BIOS Area (F0000h–FFFFFh):** This area is a single 64 KB segment that can be assigned with different attributes via PAM Control register as defined by the table above.

### 3.4.18 GBA[0:15]—RDRAM* Device Group Boundary Address Register (Device 0)

| | |
|---|---|
| Address Offset: | 60–7Fh |
| Default: | 0001h |
| Access: | Read/Write |
| Size | 16 bits/register |

This register is locked and becomes read only when the D_CLK bit in the SMRAM register is set. This is done to improve SMM security.

The Direct RDRAM device-pairs are logically arranged into groups. There are eight groups when the MCH is configured for single channel-pair mode operation (No repeater hubs), and there are four groups for multiple channel-pair mode operation (with repeater hubs). Each group requires a separate GBA register. The GBA registers define group ID and the upper and lower addresses for each group in a channel-pair. Contents of bits 0:10 of this register represent the boundary addresses in 16-MB granularity.

For example, a value of 01h indicates that the programmed group applies to memory below 16 MB. Only the first eight GBA registers are used in single channel-pair mode. All 16 GBA registers are used in multiple channel-pair mode. Note that GBA15 must always contain the group boundary address that points to the TOM, whether the MCH is being used in single channel-pair mode or multiple channel-pair mode.

| | |
|---|---|
| 60–61h | GBA0 = Total memory in group0 (in 16 MB) |
| 62–63h | GBA1 = Total memory in group0 + group1 (in 16 MB) |
| 64–65h | GBA2 = Total memory in group0 + group1 + group2 (in 16 MB) |
| 66–67h | GBA3 = Total memory in group0 + group1 + group2 + group3 (in 16 MB) |
| 7E–7Fh | GBA15 = Total memory in group0 + group1 + group2 + … + group15 (in 16 MB) |

| Bit | Description |
|---|---|
| 15:14 | **Channel ID (CHID).** Reflects the ID of the Rambus* Channel described by this GBA entry. This field is only used when Intel® MRH-R is present. |
| 13:11 | **Group ID (GID).** This 3-bit value is used to identify a logical group of Direct RDRAM* devices. This value and appropriate address bits are used to generate the device RDRAM_D device ID.<br>**Note:** All device-pairs populated in a group must be of the same memory technology. |
| 10:0 | **Group Boundary Address (GBA).** This 11-bit value is compared against address lines A[34:24] to determine the upper address limit of a particular group of devices<br>(i.e., GBA minus previous GBA = group size). |

## 3.4.19 RDPS—RDRAM* Device Pool Sizing Register (Device 0)

| | |
|---|---|
| Address Offset: | 88h |
| Default Value: | 10h |
| Access: | Read Only, Write Once, Lock |
| Size: | 8 bits |

| Bit | Description |
|---|---|
| 7 | **Pool Lock (LOCK).**<br>1 = RDPS register is read only.<br>0 = RDPS register is read/write. |
| 6 | **Reserved** |
| 5 | **Reinitialize Direct RDRAM* Device Pools (POOLINIT).** When the POOLINIT bit is set, the Direct RDRAM device pools are reinitialized to the default value contained in this register. As long as this bit is 0, the other fields in this register may be modified without changing the behavior of the pools. Only when this bit is set or when a thermal over-temperature condition occurs are the values programmed into this register re-examined by the pool logic.<br>Following a write of 1 to this bit, the new pool sizes will take effect and the DRAM interface logic will perform any operations (e.g., NAP'ing devices) necessary to comply with the new pool constraints. When these compliance operations are completed and the MCH is operating with the new pool settings, the POOLINIT bit is cleared to 0. Software can poll the bit to check for completion of the pool mode transition prior to proceeding with test cases.<br>**Note:** While over-temperature conditions (during OVERT# or current calibration) do cause new pool values to be loaded, they do not have any effect on the contents of this bit. |
| 4 | **Pool C Operating Mode (PCS).**<br>1 =All devices found neither in pool A nor in pool B are assumed to be in nap mode.<br>0 =all devices in pool C are assumed to be in standby mode.<br>**Note:** Even though this bit defaults to 1 (which is pool C nap mode), the MCH functionally defaults to pool C standby mode. This bit must be written to a 1 to invoke pool C nap mode. |
| 3:2 | **Pool A Capacity (PAC).** This field defines the maximum number of Direct RDRAM devices that can reside in Pool A at a time.<br>00 = 1<br>01 = 2<br>10 = 4<br>11 = 8 |
| 1:0 | **Pool B Capacity (PBC).** This field defines the maximum number of Direct RDRAM devices that can reside in Pool B at a time.<br>00 = 1<br>01 = 4<br>10 = 8<br>11 = 16 |

## 3.4.20 DRD—RDRAM* Device Register Data Register (Device 0)

| | |
|---|---|
| Address Offset: | 90–93h |
| Default Value: | 0000h |
| Access: | Read/Write |
| Size: | 32 bits |

| Bit | Description |
|---|---|
| 31:0 | **Register Data (RD).** Bits 31:0 contain the 32 bits of data to be written to a Direct RDRAM* device register or the data read from a Direct RDRAM device register as a result of IOP execution. Data will be valid when the IIO bit of the RICM register has transitioned from 1 to 0. Bits 31:16 apply to Rambus* Channel A and bits 15:0 apply to Rambus Channel B. |

## 3.4.21 RICM—RDRAM* Device Initialization Control Management Register (Device 0)

| | |
|---|---|
| Address Offset: | 94–97h |
| Default Value: | 00000000h |
| Access: | Read/Write, Read Only |
| Size: | 32 bits |

| Bit | Description |
|---|---|
| 31:30 | **Reserved** |
| 29:28 | **Time To Powerup (TPU).** This field defines the total powerdown exit time for Direct RDRAM* devices and corresponds to the Direct RDRAM* device (PDNA+tPDNB) timing.<br>00 = 42.0 µs<br>01 = 34.5 µs<br>10 = 27.0 µs<br>11 = 19.5 µs |
| 27 | **Initialization Complete (IC).** This bit is for hardware use.<br>1 = BIOS sets this bit to 1 after the initialization of the Direct RDRAM device memory array is complete. |
| 26:25 | **Reserved** |
| 24 | **MRH-R Present (MRHRP).**<br>1 = This bit is asserted by software when it detects the presence of Memory Repeater Hub for Direct RDRAM device in the system. |
| 23 | **Initiate Initialization Operation (IIO).** The software must check to see if this bit is 0 before writing to it.<br>1 = Execution of the initialization operation specified by IOP starts.<br>0 = After the execution is completed, the MCH clears this bit to 0. The operations that specify register data read from the Direct RDRAM device will have the data valid in DRD register when this is cleared to 0. |
| 22 | **Reserved** |

| Bit | Description |
|---|---|
| 21:20 | **Channel ID (CID).** This field specifies the channel address for which the initialization or the channel reset operation is initialed. |
| 19 | **Broadcast Address (BA).**<br>1 = Initialization operation (IOP) is broadcast to all devices. When this bit is set to 1, the DSA field is don’t care. |
| 18:10 | **Device Register Address (DRA).** This field specifies the register address for the registers read and write operations. |
| 9:5 | **Serial Device/Channel Address (SDCA).** This 5-bit field specifies the serial device ID of the Direct RDRAM* device to which the initialization operation is targeted for the next SIO command to be sent by MCH. |
| 4:0 | **Initialization Opcode (IOP).** This field specifies the initialization operation to be done on Direct RDRAM devices.<br>00000 = RDRAM Register Read<br>00001 = RDRAM Register Write<br>00010 = RDRAM Set Reset<br>00011 = Reserved<br>00100 = RDRAM Set Fast Clock Mode<br>00101 = RDRAM Temperature Calibrate Enable<br>00110 = RDRAM Temperature Calibrate<br>00111 = Reserved<br>01000 = MRH Redirect Next SIO<br>01001 = MRH Stick SIO Reset<br>01010 = Reserved<br>01011 = RDRAM Clear Reset<br>01100 = Reserved<br>01101 = Reserved<br>01110 = Reserved<br>01111 = Reserved<br>10000 = RDRAM Current Calibration<br>10001 = RDRAM SIO Reset<br>10010 = RDRAM Powerdown Exit<br>10011 = RDRAM Powerdown Entry<br>10100 = RDRAM Nap Entry<br>10101 = RDRAM Nap Exit<br>10110 = RDRAM Refresh<br>10111 = RDRAM Precharge<br>11000 = Manual Current Calibr. of MCH RAC<br>11001 = MCH RAC load RAC A config. Reg.<br>11010 = MCH RAC load RAC B config. Reg.<br>11011 = Initialize MCH RAC<br>11100 = MCH RAC Current Calibration<br>11101 = MCH RAC Thermal Calibration<br>11110 = Reserved<br>11111 = PowerUp All Sequence |

## 3.4.22 SMRAM—System Management RAM Control Register (Device 0)

Address Offset: 9Dh
Default Value: 02h
Access: Read/Write, Read Only, Lock
Size: 8 bits

The SMRAMC register controls how accesses to Compatible and Extended SMRAM spaces are treated. The open, close, and lock-bits function only when G_SMRAME bit is set to a 1. Also, the open bit must be reset before the lock-bit is set. A system device must not access the System Management Mode (SMM) space through the graphic aperture GTLB.

| Bit | Description |
|---|---|
| 7 | **Reserved** |
| 6 | **SMM Space Open (D_OPEN).**<br>1 = 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. When D_LCK is set to a 1, D_OPEN is reset to 0 and becomes read only.<br>0 = Not open |
| 5 | **SMM Space Closed (D_CLS).**<br>1 = Closed. 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.<br>0 = Not Closed<br>**Note:** That the D_CLS bit only applies to Compatible SMM space. |
| 4 | **SMM Space Locked (D_LCK).**<br>1 = 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, 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.<br>0 = Can only be cleared by a full reset. |
| 3 | **Global SMRAM Enable (G_SMRAME).**<br>1 = Enable. 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 to be set to 1.<br>0 = Disable<br>**Note:** Once D_LCK is set, this bit becomes read only. |
| 2:0 | **Compatible SMM Space Base Segment (C_BASE_SEG)—RO.** This field indicates the location of 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 the hub interface. C_BASE_SEG is hardwired to 010 to indicate that the MCH supports the SMM space at A0000h–BFFFFh. |

### 3.4.23 ESMRAMC—Extended System Management RAM Control Register (Device 0)

| | |
|---|---|
| Address Offset: | 9Eh |
| Default Value: | 38h |
| Access: | Read Only, Read/Write, Read/Write Clear, Lock |
| Size: | 8 bits |

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 MB.

| Bit | Description |
|---|---|
| 7 | **H_SMRAM_EN (H_SMRAME)—R/W.** This bit controls the SMM memory space location (i.e., above 1 MB or below 1 MB).<br>1 = When G_SMRAME is 1, the high SMRAM memory space is enabled. SMRAM accesses from FEDA0000h to FEDBFFFFh are remapped to DRAM address 000A0000h–000BFFFFh.<br>**Note:** Once D_LCK is set, this bit becomes read only. |
| 6 | **E_SMRAM_ERR (E_SMERR)—R/WC.**<br>1 = The host accesses the defined memory ranges in Extended SMRAM (High Memory and T-segment) while not in SMM space and with the D-OPEN bit = 0.<br>0 = It is software's responsibility to clear this bit. The software must write a 1 to clear this bit. |
| 5 | **SMRAM_Cache (SM_CACHE)—R/O.** Hardwired to 1. |
| 4 | **SMRAM_L1_EN (SM_L1)—RO.** Hardwired to 1. |
| 3 | **SMRAM_L2_EN (SM_L2)—RO.** Hardwired to 1. |
| 2:1 | **TSEG_SZ[1:0] (T_SZ)—R/W.** This field selects the size of the TSEG memory block if enabled. This memory is taken from the top of DRAM space (i.e., TOM - TSEG_SZ), which is no longer claimed by the memory controller (all accesses to this space are sent to the hub interface if TSEG_EN is set). This field decodes as follows:<br>00 = (TOM - 128 KB) to TOM<br>01 = (TOM - 256 KB) to TOM<br>10 = (TOM - 512 KB) to TOM<br>11 = (TOM - 1 MB) to TOM<br>**Note:** Once D_LCK is set, this bit becomes read only. |
| 0 | **TSEG_EN (T_EN)—R/W.** Enabling of SMRAM memory (TSEG, 128 KB, 256 KB, 512 KB or 1 MB of additional SMRAM memory) for Extended SMRAM space only.<br>1 = Enable. When G_SMRAME =1 and TSEG_EN = 1, the TSEG is enabled to appear in the appropriate physical address space.<br>0 = Disable.<br>**Note:** Once D_LCK is set, this bit becomes read only. |

### 3.4.24 ACAPID—AGP Capability Identifier Register (Device 0)

Address Offset: A0–A3h
Default Value: 00200002h
Access: Read Only
Size: 32 bits

This register provides standard identifier for AGP capability.

| Bit | Description |
|---|---|
| 31:24 | **Reserved** |
| 23:20 | **Major AGP Revision Number.** These bits provide a major revision number of AGP specification to which this version of MCH conforms. This field is hardwired to value of “0010b” (i.e., implying Rev 2.x). |
| 19:16 | **Minor AGP Revision Number.** These bits provide a minor revision number of AGP specification to which this version of MCH conforms. This number is hardwired to value of “0000” (i.e., implying Revision x.0)<br>**Note:** Together with major revision number this field identifies MCH as an *AGP Revision 2.0-*compliant device. |
| 15:8 | **Next Capability Pointer.** AGP capability is the first and the last capability described via the capability pointer mechanism and therefore these bits are hardwired to 0 to indicate the end of the capability linked list. |
| 7:0 | **AGP Capability ID.** This field identifies the linked list item as containing AGP registers. This field has a value of 0000_0010b assigned by the PCI SIG. |

### 3.4.25 AGPSTAT—AGP Status Register (Device 0)

| | |
|---|---|
| Address Offset: | A4–A7h |
| Default Value: | 1F000217h |
| Access: | Read Only |
| Size: | 32 bits |

This register reports AGP device capability/status.

| Bit | Description |
|---|---|
| 31:24 | **Requests (RQ).** This field is hardwired to 1Fh to indicate a maximum of 32 outstanding AGP command requests can be handled by the MCH. This field contains the maximum number of AGP command requests the MCH is configured to manage.<br>Default =1Fh to allow a maximum of 32 outstanding AGP command requests. |
| 23:10 | **Reserved.** |
| 9 | **Side Band Addressing (SBA).** Hardwired to 1. This bit indicates that the MCH supports side band addressing. |
| 8:6 | **Reserved** |
| 5 | **Greater Than 4 GB Addressing (4GB).** Hardwired to 0. This bit indicates that the MCH does not support addresses greater than 4 GB. |
| 4 | **Fast Writes (FW).** Hardwired to 1. This bit indicates that the MCH supports Fast Writes from the host to the AGP master. |
| 3 | **Reserved** |
| 2:0 | **RATE.** After reset the MCH reports its data transfer rate capability. Bit 0 identifies if AGP device supports 1x data transfer mode, bit 1 identifies if AGP device supports 2x data transfer mode, bit 2 identifies if AGP device supports 4x data transfer mode. 1x, 2x, and 4x data transfer modes are supported by the MCH and therefore this bit field has a **Default Value = 111.**<br>**Note:** The selected data transfer mode applies to both the AD bus and SBA bus. It also applies to Fast Writes, if they are enabled. |

### 3.4.26 AGPCMD—AGP Command Register (Device 0)

| | |
|---|---|
| Address Offset: | A8–ABh |
| Default Value: | 00000000h |
| Access: | Read/Write, Read Only |
| Size: | 32 bits |

This register provides control of the AGP operational parameters.

| Bit | Description |
|---|---|
| 31:10 | **Reserved** |
| 9 | **Side Band Addressing Enable (SBA_EN).**<br>1 = Enable.<br>0 = Disable. |
| 8 | **AGP Enable.** When this bit is reset to 0, the MCH ignores all AGP operations, including the sync cycle. Any AGP operations received while this bit is set to 1 will be serviced even if this bit is reset to 0. If this bit transitions from a 1 to a 0 on a clock edge in the middle of an SBA command being delivered in 1X mode the command will be issued. When this bit is set to 1, the MCH responds to AGP operations delivered via PIPE#, or to operations delivered via SBA (if SBA_EN=1).<br>1 = Enable.<br>0 = Disable. |
| 7:6 | **Reserved** |
| 5 | **Greater Than 4 GB Addressing Enable (4GB_EN).** Hardwired to 0. The MCH, as an AGP target, does not support addressing greater than 4 GB. |
| 4 | **Fast Write Enable (FW_EN).**<br>1 = MCH uses the Fast Write protocol for memory write transactions from the MCH to the AGP master. Fast Writes will occur at the data transfer rate selected by the data rate bits (2:0) in this register. When this bit is cleared, or when the data rate bits are set to 1x mode, the Memory Write transactions from the MCH to the AGP master use standard PCI protocol.<br>0 = Disable |
| 3 | **Reserved.** |
| 2:0 | **Data Rate.** The settings of these bits determine the AGP data transfer rate. One (*and only one*) bit in this field must be set to indicate the desired data transfer rate. The same bit must be set on both master and target.<br>Bit 0 = 1X<br>Bit 1 = 2X<br>Bit 2 = 4x<br>Configuration software will update this field by setting only one bit that corresponds to the capability of AGP master (after that capability has been verified by accessing the same functional register within the AGP masters' configuration space.)<br>**Note:** This field applies to the AD and SBA buses. It also applies to Fast Writes, if they are enabled. |

## 3.4.27 AGPCTRL – AGP Control Register

| | |
|---|---|
| Address Offset: | B0–B3h |
| Default Value: | 00000000h |
| Access: | Read/Write |
| Size: | 32 bits |

This register provides for additional control of the AGP interface.

| Bit | Description |
|---|---|
| 31:8 | **Reserved** |
| 7 | **GTLB Enable (and GTLB Flush Control).**<br>1 = Enable. Selects normal operations of the Graphics Translation Lookaside Buffer.<br>0 = Disable. GTLB is flushed by clearing the valid bits associated with each entry. (Default) |
| 6:0 | **Reserved** |

## 3.4.28 APSIZE—Aperture Size (Device 0)

| | |
|---|---|
| Address Offset: | B4h |
| Default Value: | 00h |
| Access: | Read/Write |
| Size: | 8 bits |

This register determines the effective size of the graphics aperture used for a particular MCH configuration. This register can be updated by the MCH specific BIOS configuration sequence before the PCI standard bus enumeration sequence takes place. If the register is not updated, a default value will select an aperture of maximum size (i.e., 256 MB). The size of the table that will correspond to a 256 MB aperture is not practical for most applications; therefore, these bits must be programmed to a smaller practical value that will force adequate address range to be requested via APBASE register from the PCI configuration software.

<table>
<tr><th>Bit</th><th>Description</th></tr>
<tr><td>7:6</td><td><b>Reserved</b></td></tr>
<tr><td>5:0</td><td><b>Graphics Aperture Size (APSIZE).</b> Each bit in APSIZE[5:0] operates on similarly ordered bits in APBASE[27:22] of the Aperture Base configuration register. When a particular bit of this field is 0, it forces the similarly ordered bit in APBASE[27:22] to behave as “hardwired” to 0. When a particular bit of this field is set to 1, it allows corresponding bit of the APBASE[27:22] to be read/write accessible. Only the following combinations are allowed:
<table>
<tr><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th><th>Aperture Size</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>4 MB</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>8 MB</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>16 MB</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>32 MB</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>64 MB</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>128 MB</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>256 MB</td></tr>
</table>
Default for APSIZE[5:0]=000000b forces default APBASE[27:22] =000000b (i.e., all bits respond as “hardwired” to 0). This provides maximum aperture size of 256MB. As another example, programming APSIZE[5:0]=111000b hardwires APBASE[24:22]=000b and while enabling APBASE[27:25] as read/write programmable.</td></tr>
</table>

### 3.4.29 ATTBASE—Aperture Translation Table Base Register (Device 0)

| | |
|---|---|
| Address Offset: | B8–BBh |
| Default Value: | 00000000h |
| Access: | Read/Write |
| Size: | 32 bits |

This register provides the starting address of the Graphics Aperture Translation Table Base located in the main memory. This value is used by the MCH graphics aperture address translation logic (including the GTLB logic) to obtain the appropriate address translation entry required during the translation of the aperture address into a corresponding physical DRAM address. The ATTBASE register may be dynamically changed.

***Note:*** The address provided via ATTBASE is 4-KB aligned.

| Bit | Description |
|---|---|
| 31:12 | **Aperture Translation Table Base Address (ATT_BA).** This field contains a pointer to the base of the translation table used to map memory space addresses in the aperture range to addresses in main memory.<br>**Note:** This field should only be modified when the GTLB has been disabled. |
| 11:0 | **Reserved** |

### 3.4.30 AMTT—AGP Interface Multi-Transaction Timer Register (Device 0)

| | |
|---|---|
| Address Offset: | BCh |
| Default Value: | 00h |
| Access: | Read/Write |
| Size: | 8 bits |

AMTT is an 8-bit register that controls the amount of time that the MCH arbiter allows an AGP master to perform multiple back-to-back transactions. The MCH AMTT mechanism is used to optimize the performance of the AGP master (using PCI protocol) that performs multiple back-to-back transactions to fragmented memory ranges (and as a consequence it can not use long burst transfers). The AMTT mechanism applies to the host-AGP transactions as well and it guarantees to the processor a fair share of the AGP interface bandwidth.

The number of clocks programmed in the AMTT represents the guaranteed time slice (measured in 66 MHz clocks) allotted to the current agent (either AGP master or host bridge) after which the AGP arbiter will grant the bus to another agent. The default value of AMTT is 00h and disables this function. The AMTT value can be programmed with 8-clock granularity. For example, if the AMTT is programmed to 18h, the selected value corresponds to the time period of 24 AGP (66 MHz) clocks.

| Bit | Description |
|---|---|
| 7:3 | **Multi-Transaction Timer Count Value.** The number programmed in this field represents the guaranteed time slice (measured in eight 66 MHz clock granularity) allotted to the current agent (either AGP master or MCH) after which the AGP arbiter will grant the bus to another agent. |
| 2:0 | **Reserved.** |

### 3.4.31 LPTT—Low Priority Transaction Timer Register (Device 0)

| | |
|---|---|
| Address Offset: | BDh |
| Default Value: | 00h |
| Access: | Read/Write |
| Size: | 8 bits |

LPTT is an 8-bit register similar in function to AMTT. This register is used to control the minimum tenure on the AGP for low priority data transaction (both reads and writes) issued using PIPE# or SB mechanisms.

The number of clocks programmed in the LPTT represents the guaranteed time slice (measured in 66 MHz clocks) allotted to the current low priority AGP transaction data transfer state. This does not necessarily apply to a single transaction but it can span over multiple low-priority transactions of the same type. After this time expires, the AGP arbiter may grant the bus to another agent if there is a pending request. The LPTT does not apply in the case of high-priority request where ownership is transferred directly to high-priority requesting queue. The default value of LPTT is 00h and disables this function. The LPTT value can be programmed with 8-clock granularity. For example, if the LPTT is programmed to 10h, then the selected value corresponds to the time period of 16 AGP (66 MHz) clocks.

| Bit | Description |
|---|---|
| 7:3 | **Low Priority Transaction Timer Count Value.** The number of clocks programmed in these bits represents the guaranteed time slice (measured in eight 66 MHz clock granularity) allotted to the current low priority AGP transaction data transfer state. |
| 2:0 | **Reserved** |

### 3.4.32 RDTR—RDRAM* Device Timing Register (Device 0)

| | |
|---|---|
| Address Offset: | BEh |
| Default Value: | 00h |
| Access: | Read/Write |
| Size: | 8 bits |

This 8-bit register defines the timing parameters for all devices in the Rambus Channel. The BIOS programs this register with the "least common denominator" values after reading configuration registers of each device in the Rambus Channel. This register applies to the entire DRAM array.

| Bit | Description |
|---|---|
| 7:6 | **Row to Column Delay (tRCD).** This field defines the minimum interval between opening a row and column operation on that row in units of Direct Rambus clocks.<br>00 = Reserved<br>01 = 7 Rclks<br>10 = 9 Rclks<br>11 = Reserved<br>**Note:** When using an Intel® MRH-R, this field should be set to 10 (9 Rclks). |
| 5 | **Reserved** |
| 4:0 | **RDRAM* Total CAS Access Delay (tRDRAM).** This field defines the minimum round trip propagation time of the Rambus* Channel in units of Direct RDRAM device clocks. This value includes the CAS access time, the channel delay time, or any Intel MRH-R delay time.<br>tRDRAM = tCAC + tRDLY<br>• tRDRAM has a minimum value of 8 Rclks since the supported Direct RDRAM device tCAC = 8 Rclks.<br>• tRDLY is the total channel delay time and should include the channel delay time of the Direct RDRAM device in the MCH Direct RDRAM device interface, the Intel MRH-R propagation delay time, and the channel delay time of the Direct RDRAM device in the Intel MRH-R Direct RDRAM device interface.<br>• The minimum tRDRAM value for use with the Intel MRH-R is 14 Direct RDRAM device clks (tCAC of 8 + MRH-R delay of 6). The maximum tRDRAM value for use with Intel MRH-R is 19 Direct RDRAM device clks (tCAC of 8 + MRH-R delay of 6 + total channel delay of 5). |

**Table 8. Valid tRCD and tCAC combinations for 300 MHz and 400 MHz**

| Direct RDRAM Device * Frequency (Rclk) | tRCD in Rclks | tCAC in Rclks |
|---|---|---|
| 300 MHz | 7 | 8 |
| 400 MHz | 9 | 8 |
| 400 MHz | 7 | 8 |

### 3.4.33 TOM—Top of Low Memory Register (Device 0)

| | |
|---|---|
| Address Offset: | C4h |
| Default Value: | 0100h |
| Access: | Read/Write |
| Size: | 16 bits |

A memory hole is present under normal operating conditions from TOM up to the 4-GB address where TOM is the Top of Low Memory register. This hole is used to access devices present behind hub interfaces_A–C, the AGP bus, the memory-mapped APIC register, and the boot BIOS area just below 4 GB. If the total amount of main memory is less than 4 GB, then the addresses (i.e., not their “values”) indicated by the TOM and GBA15 (or TOM and SRBA7) registers will be identical.

***Note:*** That this register must be set to a value of 0100h (16 MB) or greater.

| Bit | Description |
|---|---|
| 15:4 | **Top of Low Memory (TOM).** This register contains the address that corresponds to bits 31 to 20 (1-MB granularity) of the maximum DRAM memory address that lies below 4 GB. Configuration software should set this value to either the maximum amount of memory in the system or to the minimum address allocated for PCI memory or the graphics aperture, whichever is smaller.<br>**Programming example:** C00h = 3 GB (assuming that GAR15 is set > 4 GB):<br>• An access to 0_C000_0000h or above (but < 4 GB) will be considered above the TOM; therefore, the access is not to DRAM. It may go to AGP or one of the hub interfaces and will subtractively decode to Hub Interface_A.<br>• An access to 0_BFFF_FFFFh and below will be considered below the TOM and go to DRAM.<br>**Note:** Locked accesses that cross TOM are illegal and should not be performed. |
| 3:0 | **Reserved** |

### 3.4.34 ERRSTS—Error Status Register (Device 0)

| | |
|---|---|
| Address Offset: | C8–C9h |
| Default Value: | 0000h |
| Access: | Read/Write Clear |
| Size: | 16 bits |

This register is used to report various error conditions via the hub interface messages to the ICH2. An, SERR, SMI, or SCI error message may be generated via Hub Interface_A on a 0-to-1 transition of any of these flags, when enabled, in the PCICMD/ERRCMD, SMICMD, or SCICMD registers respectively. These bits are set, regardless of whether or not the SERR is enabled and generated.

| Bit | Description |
|---|---|
| 15 | **FSB Request Parity Error (FSBRPAR).**<br>1 = MCH detected a parity error on either the address or request signals of the system bus.<br>0 = Software must write a 1 to clear this bit. |
| 14 | **System Bus Data Parity Error (FSBDPAR).**<br>1 = MCH detected a data parity error on the system bus.<br>0 = Software must write a 1 to clear this bit. |
| 13 | **System Bus Address Strobe Glitch Detected (FSBAGL).**<br>1 = MCH detected a glitch on one of the address strobes.<br>0 = Software must write a 1 to clear this bit. |
| 12 | **System Bus Data Strobe Glitch Detected (FSBDGL).**<br>1 = MCH detected a glitch on one of the data strobes.<br>0 = Software must write a 1 to clear this bit. |
| 11 | **Reserved** |
| 10 | **External Thermal Sensor Throttle (ETST).**<br>1 = MCH detected a rising edge on the OVERT# or the Direct RDRAM* devices report an over temperature conditions. The OVERT# should be used to receive an interrupt from an external thermal sensor when the sensor has been tripped.<br>0 = Software must write a 1 to clear this bit. |
| 9 | **LOCK to non-DRAM Memory Flag (LCKF).**<br>1 = A host-initiated LOCK cycle targeting non-DRAM memory space occurred.<br>0 = Software must write a 1 to clear this bit. |
| 8 | **System Bus Address above TOM (FSBATOM).**<br>1 = MCH detected an address above 4 GB and above the Top of Low Memory.<br>0 = Software must write a 1 to clear this bit. |
| 7 | **Reserved** |

| Bit | Description |
|---|---|
| 6 | **SERR on Hub Interface_A Target Abort (TAHLA).**<br>1 = MCH detected that an MCH originated Hub Interface_A cycle was terminated with a Target Abort completion packet or special cycle.<br>0 = Software must write a 1 to clear this bit. |
| 5 | **MCH Detects Unimplemented Hub Interface_A Special Cycle (HIAUSC).**<br>1 = MCH detected an Unimplemented Special Cycle on the Hub Interface_A.<br>0 = Software must write a 1 to clear this bit. |
| 4 | **AGP Access Outside of Graphics Aperture Flag (OOGF).**<br>1 = An AGP access occurred to an address that is outside of the graphics aperture range.<br>0 = Software must write a 1 to clear this bit. |
| 3 | **Invalid AGP Access Flag (IAAF).**<br>1 = An AGP access was attempted outside of the graphics aperture and either to the 640 KB–1 MB range or above the TOM.<br>0 = Software must write a 1 to clear this bit. |
| 2 | **Invalid Graphics Aperture Translation Table Entry (ITTEF).**<br>1 = An invalid translation table entry was returned in response to an AGP access to the graphics aperture.<br>0 = Software must write a 1 to clear this bit. |
| 1 | **Multiple-bit DRAM ECC Error Flag (DMERR).**<br>1 = A memory read data transfer had an uncorrectable multiple-bit error. When this bit is set, the address, channel number, and device number that caused the error are logged in the EAP register. Once this bit is set, the EAP, CN, DN, and ES fields are locked until the processor clears this bit by writing a 1. Software uses bits [1:0] to detect whether the logged error address is for single- or multiple-bit error.<br>0 = Once software completes the error processing, a value of 1 is written to this bit field to clear the value (back to 0) and unlock the error logging mechanism. |
| 0 | **Single-bit DRAM ECC Error Flag (DSERR).**<br>1 = A memory read data transfer had a single-bit correctable error and the corrected data was sent for the access. When this bit is set, the address, channel number, and device number that caused the error are logged in the EAP register. Once this bit is set, the EAP, CN, DN, and ES fields are locked to further single-bit error updates until the processor clears this bit by writing a 1. A multiple-bit error that occurs after this bit is set will overwrite the EAP, CN, and DN fields with the multiple-bit error signature and the MEF bit will also be set.<br>0 = Software must write a 1 to clear this bit and unlock the error logging mechanism. |

### 3.4.35 ERRCMD—Error Command Register (Device 0)

| | |
|---|---|
| Address Offset: | CA–CBh |
| Default Value: | 0000h |
| Access: | Read/Write |
| Size: | 16 bits |

This register enables various errors to generate a SERR message via the Hub Interface_A. Since the MCH does not have an SERR# signal, SERR messages are passed from the MCH to the ICH2 over the hub interface. When a bit in this register is set, a SERR message will be generated on Hub Interface_A when the corresponding flag is set in the ERRSTS register. The actual generation of the SERR message is globally enabled for Device 0 via the PCI Command register.

***Note:*** An error can generate one and only one error message via the Hub Interface_A. It is software's responsibility to make sure that when an SERR error message is enabled for an error condition, SMI and SCI error messages are disabled for that same error condition.

| Bit | Description |
|---|---|
| 15 | **SERR on System Bus Request Parity Error (HBRERR).**<br>1 = Enable. Generation of the Hub Interface_A SERR message is enabled for the parity errors on the address or request signals of the system bus.<br>0 = Disable. |
| 14 | **SERR on System Bus Data Parity Error (HBDERR).**<br>1 = Enable. Data parity errors on the System Bus will cause the MCH to send an SERR message over Hub Interface_A to the ICH2.<br>0 = Disable. |
| 13 | **SERR on System Bus Address Strobe Glitch (AGLERR).**<br>1 = Enable. MCH will generate a Hub Interface_A SERR message when a glitch is detected on one of the system bus address strobes.<br>0 = Disable. |
| 12 | **SERR on System Bus Data Strobe Glitch (DGLERR).**<br>1 = Enable. MCH will generate a Hub Interface_A SERR message when a glitch is detected on one of the system bus data strobes.<br>0 = Disable. |
| 11 | **Reserved** |
| 10 | **SERR on External Thermal Sensor Trip (THERM_SERR).**<br>1 = Enable. Generation of the Hub Interface_A SERR message is enabled when the MCH has detected a rising edge on the OVERT# or the Direct RDRAM* devices report an over-temperature conditions.<br>0 = Disable. |
| 9 | **SERR on Non-DRAM Lock (LCKERR).**<br>1 = Enable. MCH will generate a Hub Interface_A SERR special cycle when a processor lock cycle is detected that does not hit DRAM.<br>0 = Disable. |

| Bit | Description |
|---|---|
| 8 | **SERR on Host Bus Access above TOM (HBATOMERR).**<br>1 = Enable. MCH will generate Hub Interface_A SERR special cycle when the processor generates an access above 4 GB and above the TOM.<br>0 = Disable. |
| 7 | **Reserved** |
| 6 | **SERR on Target Abort on Hub Interface_A Exception (TAHLA_SERR).**<br>1 = Enable. Generation of the Hub Interface_A SERR message is enabled when an MCH-originated Hub Interface_A cycle is completed with “Target Abort” completion packet or special cycle status.<br>0 = Disable. |
| 5 | **SERR on Detecting Hub Interface_A Unimplemented Special Cycle (HIAUSCERR).** SERR messaging for Device 0 is globally enabled in the PCICMD register.<br>1 = Enable. MCH generates an SERR message over Hub Interface_A when an Unimplemented Special Cycle is received on the hub interface.<br>0 = Disable. The MCH does not generate an SERR message for this event. |
| 4 | **SERR on AGP Access Outside of Graphics Aperture (OOGF_SERR).**<br>1 = Enable. Generation of the Hub Interface_A SERR message is enabled when an AGP access occurs to an address outside of the graphics aperture.<br>0 = Disable. |
| 3 | **SERR on Invalid AGP Access (IAAF_SERR).**<br>1 = Enable. Generation of the Hub Interface_A SERR message is enabled when an AGP access occurs to an address outside of the graphics aperture and either to the 640 KB – 1 MB range or above the TOM.<br>0 = Disable. |
| 2 | **SERR on Invalid Translation Table Entry (ITTEF_SERR).**<br>1 = Enable. Generation of the Hub Interface_A SERR message is enabled when an invalid translation table entry was returned in response to an AGP access to the graphics aperture.<br>0 = Disable. |
| 1 | **SERR Multiple-Bit DRAM ECC Error (DMERR_SERR).**<br>1 = Enable. Generation of the Hub Interface_A SERR message is enabled when the MCHDRAM controller detects a multiple-bit error.<br>0 = Disable. For systems not supporting ECC, this bit must be disabled. |
| 0 | **SERR on Single-bit ECC Error (DSERR).**<br>1 = Enable. Generation of the Hub Interface_A SERR message is enabled when the MCH DRAM controller detects a single-bit error.<br>0 = Disable. For systems that do not support ECC, this bit must be disabled. |

### 3.4.36 SMICMD—SMI Command Register (Device 0)

| | |
|---|---|
| Address Offset: | CC–CDh |
| Default Value: | 0000h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register enables various errors to generate a SMI message via the Hub Interface_A.

***Note:*** An error can generate one and only one error message via the Hub Interface_A. It is software's responsibility to make sure that when an SMI error message is enabled for an error condition, SERR and SCI error messages are disabled for that same error condition.

| Bit | Description |
|---|---|
| 15:2 | **Reserved** |
| 1 | **SMI on Multiple-Bit DRAM ECC Error (DMERR_SMI).**<br>1 = Enable. Generation of the Hub Interface_A SMI message is enabled when the MCH DRAM controller detects a multiple-bit error.<br>0 = Disable. For systems not supporting ECC, this bit must be disabled. |
| 0 | **SMI on Single-bit ECC Error (DSERR_SMI).**<br>1 = Enable. Generation of the Hub Interface_A SMI message is enabled when the MCH DRAM controller detects a single-bit error.<br>0 = Disable. For systems that do not support ECC, this bit must be disabled. |

### 3.4.37 SCICMD—SCI Command Register (Device 0)

| | |
|---|---|
| Address Offset: | CE–CFh |
| Default Value: | 0000h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register enables various errors to generate a SCI message via the Hub Interface_A.

***Note:*** An error can generate one and only one error message via the Hub Interface_A. It is software's responsibility to make sure that when an SCI error message is enabled for an error condition, SERR and SMI error messages are disabled for that same error condition.

| Bit | Description |
|---|---|
| 15:2 | **Reserved.** |
| 1 | **SCI on Multiple-Bit DRAM ECC Error (DMERR_SCI).**<br>1 = Enable. Generation of the Hub Interface_A SCI message is enabled when the MCH DRAM controller detects a multiple-bit error.<br>0 = Disable. For systems not supporting ECC, this bit must be disabled. |
| 0 | **SCI on Single-bit ECC Error (DSERR_SCI).**<br>1 = Enable. Generation of the Hub Interface_A SCI message is enabled when the MCH DRAM controller detects a single-bit error.<br>0 = Disable. For systems that do not support ECC, this bit must be disabled. |

### 3.4.38 DRAMRC—RDRAM* Device Refresh Control Register (Device 0)

| | |
|---|---|
| Address Offset: | DC–DDh |
| Default Value: | 0000h |
| Access: | Read Only, Read/Write |
| Size: | 16 bits |

This register is loaded by configuration software with the refresh timings for all Rambus Channels present in the system. The value placed into this register should represent the least common denominator of all of the devices on the specified channel pair.

***Note:*** The refresh rate for a channel is programmed to that of the device with the fastest refresh rate on that channel. That is, if a channel has a mix of 128/144 Mbit (3.9 µs) and 256/288 Mbit (1.95 µs) technology devices, the refresh rate for the channel will be programmed to 1.95 µs.

| Bit | Description |
|---|---|
| 15:12 | **Reserved.** |
| 11:9 | **DRAM Refresh Rate for Rambus* Channel pair #3 (DRR3).** The DRAM refresh rate is adjusted according to the frequency selected by this field. Note that refresh is also disabled via this field, and that disabling refresh results in the eventual loss of DRAM data. This field is programmed by the BIOS after collecting configuration information from all Direct RDRAM* devices in the channel and determining the least common denominator value for refresh.<br>000 = Refresh Disabled<br>001 = 1.95 µs<br>010 = 3.9 µs<br>011 – 111 = Reserved |
| 8:6 | **DRAM Refresh Rate Rambus Channel Pair #2 (DRR2).** The DRAM refresh rate is adjusted according to the frequency selected by this field. Note that refresh is also disabled via this field, and that disabling refresh results in the eventual loss of DRAM data. This field is programmed by BIOS after collecting configuration information from all Direct RDRAM devices in the channel and determining the least common denominator value for refresh.<br>000 = Refresh Disabled<br>001 = 1.95 µs<br>010 = 3.9 µs<br>011 – 111 = Reserved |
| 5:3 | **DRAM Refresh Rate Rambus Channel Pair #1 (DRR1).** The DRAM refresh rate is adjusted according to the frequency selected by this field. Note that refresh is also disabled via this field, and that disabling refresh results in the eventual loss of DRAM data. This field is programmed by BIOS after collecting config information from all Direct RDRAM devices in the channel and determining the least common denominator value for refresh.<br>000 = Refresh Disabled<br>001 = 1.95 µs<br>010 = 3.9 µs<br>011 – 111 = Reserved |
| 2:0 | **DRAM Refresh Rate Rambus Channel Pair #0 (DRR0).** The DRAM refresh rate is adjusted according to the frequency selected by this field. Note that refresh is also disabled via this field, and that disabling refresh results in the eventual loss of DRAM data. This field is programmed by BIOS after collecting configuration information from all Direct RDRAM devices in the channel and determining the least common denominator value for refresh.<br>000 = Refresh Disabled<br>001 = 1.95 µs<br>010 = 3.9 µs<br>011 – 111 = Reserved |

### 3.4.39 SKPD—Scratchpad Data (Device 0)

| | |
|---|---|
| Address Offset: | DE–DFh |
| Default Value: | 0000h |
| Access: | Read/Write |
| Size: | 16 bits |

| Bit | Description |
|---|---|
| 15:0 | **Scratchpad [15:0].** These bits are R/W storage bits that have no effect on the MCH functionality. |

### 3.4.40 DERRCTL_STS—DRAM Error Control/Status Register (Device 0)

| | |
|---|---|
| Address Offset: | E2–E3h |
| Default Value: | 0000h |
| Access: | Read Only |
| Size: | 16 bits |

This register enables and reflects the status of various errors checking functions that the MCH supports on the DRAM interface.

| Bit | Description |
|---|---|
| 15:8 | **Reserved** |
| 7:0 | **DRAM ECC Syndrome (DECCSYN).** After a DRAM ECC error, hardware loads this field with a syndrome that describes the set of bits found to be in error. Note that this field is locked from the time that it is loaded up to the time when the error flag is cleared by software. If the first error was a single-bit, correctable error, then a subsequent multiple-bit error will overwrite this field. In all other cases, an error that occurs after the first error and before the error flag has been cleared by software will escape recording. |

### 3.4.41 EAP—Error Address Pointer Register (Device 0)

| | |
|---|---|
| Address Offset: | E4–E7h |
| Default Value: | 0000h |
| Access: | Read Only |
| Size: | 32 bits |

This register stores the DRAM address when an ECC error occurs.

| Bit | Description |
|---|---|
| 31:9 | **Error Address Pointer (EAP).** This field is used to store address bits A[33:11] of the 4-KB block of main memory of which an error (single-bit or multiple-bit error) has occurred. Note that the value of this bit field represents the address of the first single- or the first multiple-bit error occurrence after the error flag bits in the ERRSTS register have been cleared by software. A multiple-bit error will overwrite a single-bit error.<br>Once the error flag bits are set as a result of an error, this bit field is locked and does not change as a result of a new error until the error flag is cleared by software. |
| 8:1 | **Reserved.** |
| 0 | **Error Address Segment (EAS).** This bit indicates whether the reported error was found on Rambus* Channel A or on Rambus Channel B. Once the error flag bits are set as a result of an error, this bit is locked and does not change as a result of a new error until the error flag is cleared by software.<br>1 = Rambus Channel B<br>0 = Rambus Channel A |

### 3.4.42 MISC_CNTL—Miscellaneous Control Register (Device 0)

| | |
|---|---|
| Address Offset: | F4–F7h |
| Default: | 0000F874h |
| Default: | R/W |
| Size | 32 bits |

| Bit | Description |
|---|---|
| 31:22 | Reserved. |
| 21 | **Write Combining Disable (PCIBWCD).**<br>1 = Disable. Write combining is disabled for host bus writes targeting AGP (depends on configuration).<br>0 = Enable (default).<br>**Note:** This bit must be set to 1 (disable) for normal operations. |
| 20:0 | **Reserved.** |

## 3.5 AGP Bridge Registers (Device 1)

Table 9 describes the access attributes for the Device 1 configuration space.

### Table 9. MCH Configuration Space (Device 1)

| Address Offset | Symbol | Register Name | Default Value | Access |
|---|---|---|---|---|
| 00–01h | VID1 | Vendor Identification | 8086h | RO |
| 02–03h | DID1 | Device Identification | 2532h | RO |
| 04–05h | PCICMD1 | PCI Command Register | 0000h | RO, R/W |
| 06–07h | PCISTS1 | PCI Status Register | 00A0h | RO, R/WC |
| 08h | RID1 | Revision Identification | 04h | RO |
| 09h | — | Reserved | — | — |
| 0Ah | SUBC1 | Sub-Class Code | 04h | RO |
| 0Bh | BCC1 | Base Class Code | 06h | RO |
| 0Ch | — | Reserved | — | — |
| 0Dh | MLT1 | Master Latency Timer | 00h | RO, R/W |
| 0Eh | HDR1 | Header Type | 01h | RO |
| 0F–17h | — | Reserved | — | — |
| 18h | PBUSN1 | Primary Bus Number | 00h | RO |
| 19h | SBUSN1 | Secondary Bus Number | 00h | R/W |
| 1Ah | SUBUSN1 | Subordinate Bus Number | 00h | R/W |
| 1Bh | SMLT1 | Secondary Bus Master Latency Timer | 00h | RO, R/W |
| 1Ch | IOBASE1 | I/O Base Address Register | F0h | RO, R/W |
| 1Dh | IOLIMIT1 | I/O Limit Address Register | 00h | RO, R/W |
| 1E–1Fh | SSTS1 | Secondary Status Register | 02A0h | RO, R/WC |
| 20–21h | MBASE1 | Memory Base Address Register | FFF0h | RO, R/W |
| 22–23h | MLIMIT1 | Memory Limit Address Register | 0000h | RO, R/W |
| 24–25h | PMBASE1 | Prefetchable Memory Base Address Register | FFF0h | RO, R/W |
| 26–27h | PMLIMIT1 | Prefetchable Memory Limit Address Register | 0000h | RO, R/W |
| 28–3Dh | — | Reserved | — | — |
| 3Eh | BCTRL1 | Bridge Control Register | 00h | RO, R/W |
| 3Fh | — | Reserved | — | — |
| 40h | ERRCMD1 | Error Command | 00h | RO, R/W |
| 41–FFh | — | Reserved | — | — |

### 3.5.1 VID1—Vendor Identification Register (Device 1)

| | |
|---|---|
| Address Offset: | 00–01h |
| Default Value: | 8086h |
| Attribute: | Read Only |
| Size: | 16 bits |

The VID register contains the vendor identification number. This 16-bit register combined with the Device Identification register uniquely identifies any PCI device. Writes to this register have no effect.

| Bit | Description |
|---|---|
| 15:0 | **Vendor Identification Number**. This is a 16-bit value assigned to Intel. Intel VID = 8086h. |

### 3.5.2 DID1—Device Identification Register (Device 1)

| | |
|---|---|
| Address Offset: | 02–03h |
| Default Value: | 2532h |
| Attribute: | Read Only |
| Size: | 16 bits |

This 16-bit register combined with the Vendor Identification register uniquely identifies any PCI device. Writes to this register have no effect.

| Bit | Description |
|---|---|
| 15:0 | **Device Identification Number**. This is a 16-bit value assigned to the MCH Device 1.<br>MCH Device 1 DID =2532h. |

### 3.5.3 PCICMD1—PCI-PCI Command Register (Device 1)

| | |
|---|---|
| Address Offset: | 04–05h |
| Default: | 0000h |
| Access: | Read Only, Read/Write |
| Size | 16 bits |

| Bit | Descriptions |
|---|---|
| 15:10 | **Reserved** |
| 9 | **Fast Back-to-Back—RO.** Not implemented; hardwired to 0. |
| 8 | **SERR Message Enable (SERRE1)—R/W.** This bit is a global enable bit for Device 1 SERR messaging. The MCH communicates the SERR# condition by sending an SERR message to the ICH2.<br>1 = Enable. MCH is enabled to generate SERR messages over the hub interface for specific Device 1 error conditions that are individually enabled in the BCTRL register. The error status is reported in the PCISTS1 register.<br>0 = Disable. SERR message is not generated by the MCH for Device 1.<br>**Note:** This bit only controls SERR messaging for Device 1. Device 0 has its own SERRE bit to control error reporting for error conditions occurring on Device 0. |
| 7 | **Address/Data Stepping—RO.** Not implemented; hardwired to 0. |
| 6 | **Parity Error Enable (PERRE1)—RO.** Not implemented; hardwired to 0. Parity checking is not supported on the primary side of this device. |
| 5 | **Reserved** |
| 4 | **Memory Write and Invalidate Enable—RO.** Not implemented; hardwired to 0. |
| 3 | **Special Cycle Enable—RO.** Not implemented; hardwired to 0. |
| 2 | **Bus Master Enable (BME1)—R/W.** This bit is not functional. It is a R/W bit for compatibility with compliance testing software. |
| 1 | **Memory Access Enable (MAE1)—R/W.**<br>1 = Enable. This bit must be set to 1 to enable the Memory and Prefetchable memory address ranges defined in the MBASE1, MLIMIT1, PMBASE1, and PMLIMIT1 registers.<br>0 = Disable. All of Device 1’s memory space is disabled. |
| 0 | **I/O Access Enable (IOAE1)—R/W.**<br>1 = Enable. This bit must be set to1 to enable the I/O address range defined in the IOBASE1, and IOLIMIT1 registers.<br>0 = Disable. All of Device 1’s I/O space is disabled. |

## 3.5.4 PCISTS1—PCI-PCI Status Register (Device 1)

| | |
|---|---|
| Address Offset: | 06–07h |
| Default Value: | 00A0h |
| Access: | Read Only, Read/Write Clear |
| Size: | 16 bits |

PCISTS1 is a 16-bit status register that reports the occurrence of error conditions associated with primary side of the “virtual” PCI-PCI bridge in the MCH. Since this device does not physically reside on PCI_A, it reports the optimum operating conditions so that it does not restrict the capability of PCI_A.

| Bit | Descriptions |
|---|---|
| 15 | **Detected Parity Error (DPE1)—RO.** Not implemented; hardwired to 0. |
| 14 | **Signaled System Error (SSE1)—R/WC**<br>1 = MCH Device 1 generates an SERR message over the Hub Interface_A for any enabled Device 1 error condition. Device 1 error conditions are enabled in the ERRCMD, PCICMD1 and BCTRL1 registers. Device 1 error flags are read/reset from the ERRSTS and SSTS1 register.<br>0 = Software clears this bit by writing a 1 to it. |
| 13 | **Received Master Abort Status (RMAS1)—RO.** Not implemented; hardwired to 0. |
| 12 | **Received Target Abort Status (RTAS1)—RO.** Not implemented; hardwired to 0. |
| 11 | **Signaled Target Abort Status (STAS1)—RO.** Not implemented; hardwired to 0. |
| 10:9 | **DEVSEL# Timing (DEVT1):** This bit field is hardwired to “00b” to indicate that the Device 1 uses the fastest possible decode. |
| 8 | **Data Parity Detected (DPD1)—RO.** Not implemented; hardwired to 0. |
| 7 | **Fast Back-to-Back (FB2B1)—RO.** This bit is hardwired to 1 to indicate that the AGP port always supports fast back-to-back transactions. |
| 6 | **Reserved** |
| 5 | **66 MHz Capability—RO.** This bit is hardwired to 1 to indicate that the AGP port is 66 MHz capable. |
| 4:0 | **Reserved** |

### 3.5.5 RID1—Revision Identification Register (Device 1)

| | |
|---|---|
| Address Offset: | 08h |
| Default Value: | 04h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the revision number of the MCH Device 1. These bits are read only and writes to this register have no effect.

| Bit | Description |
|---|---|
| 7:0 | **Revision Identification Number**. This is an 8-bit value that indicates the revision identification number for the MCH Device 1.<br>A-3 Stepping = 04h |

### 3.5.6 SUBC1—Sub-Class Code Register (Device 1)

| | |
|---|---|
| Address Offset: | 0Ah |
| Default Value: | 04h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the Sub-Class Code for the MCH Device 1.

| Bit | Description |
|---|---|
| 7:0 | **Sub-Class Code (SUBC1)**. This is an 8-bit value that indicates the category of bridge for the MCH.<br>04h = Host Bridge. |

### 3.5.7 BCC1—Base Class Code Register (Device 1)

| | |
|---|---|
| Address Offset: | 0Bh |
| Default Value: | 06h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the Base Class Code of the MCH Device 1.

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BASEC)**. This is an 8-bit value that indicates the Base Class Code for the MCH Device 1.<br>06h = Bridge device. |

## 3.5.8 MLT1—Master Latency Timer Register (Device 1)

| | |
|---|---|
| Address Offset: | 0Dh |
| Default Value: | 00h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This functionality is not applicable. It is described here since these bits should be implemented as read/write to prevent standard PCI-PCI bridge configuration software from getting “confused”.

| Bit | Description |
|---|---|
| 7:3 | **Not applicable but support read/write operations.** (Reads return previously written data.) |
| 2:0 | **Reserved** |

## 3.5.9 HDR1—Header Type Register (Device 1)

| | |
|---|---|
| Offset: | 0Eh |
| Default: | 01h |
| Access: | Read Only |
| Size: | 8 bits |

This register identifies the header layout of the configuration space. No physical register exists at this location.

| Bit | Descriptions |
|---|---|
| 7:0 | This read only field always returns 01h when read. Writes have no effect. |

## 3.5.10 PBUSN1—Primary Bus Number Register (Device 1)

| | |
|---|---|
| Offset: | 18h |
| Default: | 00h |
| Access: | Read Only |
| Size: | 8 bits |

This register identifies that “virtual” PCI-PCI Bridge is connected to bus #0.

| Bit | Descriptions |
|---|---|
| 7:0 | **Bus Number.** Hardwired to 0. |

### 3.5.11 SBUSN1—Secondary Bus Number Register (Device 1)

| | |
|---|---|
| Offset: | 19h |
| Default: | 00h |
| Access: | Read /Write |
| Size: | 8 bits |

This register identifies the bus number assigned to the second bus side of the “virtual” PCI-PCI bridge (i.e., to AGP). This number is programmed by the PCI configuration software to allow mapping of configuration cycles to AGP.

| Bit | Descriptions |
|---|---|
| 7:0 | **Bus Number.** Programmable<br>Default = 00h. |

### 3.5.12 SUBUSN1—Subordinate Bus Number Register (Device 1)

| | |
|---|---|
| Offset: | 1Ah |
| Default: | 00h |
| Access: | Read /Write |
| Size: | 8 bits |

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

| Bit | Descriptions |
|---|---|
| 7:0 | **Bus Number.** Programmable<br>Default = 00h. |

### 3.5.13 SMLT1—Secondary Master Latency Timer Register (Device 1)

| | |
|---|---|
| Address Offset: | 1Bh |
| Default Value: | 00h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This register controls the bus tenure of the MCH on AGP. MLT is an 8-bit register that controls the amount of time the MCH as an AGP/PCI bus master, can burst data on the AGP Bus. The count value is an 8-bit quantity; however, MLT[2:0] are reserved and assumed to be 0 when determining the count value. The MCH’s MLT is used to guarantee to the AGP master a minimum amount of the system resources. When the MCH begins the first AGP FRAME# cycle after being granted the bus, the counter is loaded and enabled to count from the assertion of FRAME#. If the count expires while the MCH’s grant is removed (due to AGP master request), then the MCH will lose the use of the bus and the AGP master agent may be granted the bus. If the MCH’s bus grant is not removed, the MCH will continue to own the AGP bus, regardless of the MLT expiration or idle condition. Note that the MCH must always properly terminate an AGP transaction, with FRAME# negation prior to the final data transfer.

The number of clocks programmed in the MLT represents the guaranteed time slice (measured in 66 MHz AGP clocks) allotted to the MCH, after which it must complete the current data transfer phase and then surrender the bus as soon as its bus grant is removed. For example, if the MLT is programmed to 18h, the value is 24 AGP clocks. The default value of MLT is 00h and disables this function. When the MLT is disabled, the burst time for the MCH is unlimited (i.e., the MCH can burst forever).

| Bit | Description |
|---|---|
| 7:3 | **Secondary MLT Counter Value.** Default=0 (i.e., SMLT disabled) |
| 2:0 | **Reserved** |

### 3.5.14 IOBASE1—I/O Base Address Register (Device 1)

| | |
|---|---|
| Address Offset: | 1Ch |
| Default Value: | F0h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This register controls the host-to-AGP I/O access routing based on the following formula:

$$\text{IO\_BASE} \le \text{address} \le \text{IO\_LIMIT}$$

Only upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] are treated as 0. Thus, the bottom of the defined I/O address range will be aligned to a 4-KB boundary.

***Note:*** BIOS must not set this register to 00h; otherwise, 0CF8h/0CFCh accesses will be forwarded to AGP.

| Bit | Description |
|---|---|
| 7:4 | **I/O Address Base.** Corresponds to A[15:12] of the I/O address. Default=F0h |
| 3:0 | **Reserved** |

### 3.5.15 IOLIMIT1—I/O Limit Address Register (Device 1)

| | |
|---|---|
| Address Offset: | 1Dh |
| Default Value: | 00h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This register controls the host-to-AGP I/O access routing based on the following formula:

$$\text{IO\_BASE} \le \text{address} \le \text{IO\_LIMIT}$$

Only the upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] are assumed to be FFFh. Thus, the top of the defined I/O address range will be at the top of a 4-KB aligned address block.

| Bit | Description |
|---|---|
| 7:4 | **I/O Address Limit.** Corresponds to A[15:12] of the I/O address. Default=0 |
| 3:0 | **Reserved.** Only 16-bit addressing is supported. |

### 3.5.16 SSTS1—Secondary PCI-PCI Status Register (Device 1)

| | |
|---|---|
| Address Offset: | 1E–1Fh |
| Default Value: | 02A0h |
| Access: | Read Only, Read/Write Clear |
| Size: | 16 bits |

SSTS1 is a 16-bit status register that reports the occurrence of error conditions associated with secondary side (i.e., AGP side) of the “virtual” PCI-PCI bridge in the MCH.

| Bit | Descriptions |
|---|---|
| 15 | **Detected Parity Error (DPE1)—R/WC.**<br>1 = MCH detected a parity error in the address or data phase of AGP bus transactions.<br>0 = Software clears this bit by writing a 1 to it. |
| 14 | **Received System Error (SSE1)—R/WC.**<br>1 = MCH detects G_SERR# assertion on the secondary side of this device.<br>0 = Software clears this bit by writing a 1 to it. |
| 13 | **Received Master Abort Status (RMAS1)—R/WC.**<br>1 = MCH terminates a Host-to-AGP with an unexpected master abort.<br>0 = Software clears this bit by writing a 1 to it. |
| 12 | **Received Target Abort Status (RTAS1)—R/WC.**<br>1 = MCH-initiated transaction on AGP is terminated with a target abort.<br>0 = Software clears this bit by writing a 1 to it. |
| 11 | **Signaled Target Abort Status (STAS1)—RO.** Hardwired to a 0; the MCH does not generate target abort on AGP. |
| 10:9 | **DEVSEL# Timing (DEVT1)—RO.** This 2-bit field indicates the timing of the G_DEVSEL# signal when the MCH responds as a target on AGP, and is hardwired to the value 01b (medium) to indicate the time when a valid G_DEVSEL# can be sampled by the initiator of the PCI cycle. |
| 8 | **Master Data Parity Error Detected (DPD1)—RO.** Hardwired to 0. MCH does not implement the G_PERR# signal. |
| 7 | **Fast Back-to-Back (FB2B1)—RO.** Hardwired to 1; MCH, as a target, supports fast back-to-back transactions on AGP. |
| 6 | **Reserved** |
| 5 | **66 MHz Capable (CAP66)—RO.** Hardwired to 1; AGP bus is capable of 66 MHz operation. |
| 4:0 | **Reserved**. |

## 3.5.17 MBASE1—Memory Base Address Register (Device 1)

| | |
|---|---|
| Address Offset: | 20–21h |
| Default Value: | FFF0h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host to AGP non-prefetchable memory accesses routing based on the following formula:

MEMORY_BASE1≤ address ≤ MEMORY_LIMIT1

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 0s when read. Configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be 0. Thus, the bottom of the defined memory address range will be aligned to a 1-MB boundary.

| Bit | Description |
|---|---|
| 15: 4 | **Memory Address Base 1 (MEM_BASE1).** Corresponds to A[31:20] of the memory address. |
| 3:0 | **Reserved** |

## 3.5.18 MLIMIT1—Memory Limit Address Register (Device 1)

| | |
|---|---|
| Address Offset: | 22–23h |
| Default Value: | 0000h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-AGP non-prefetchable memory accesses routing based on the following formula:

MEMORY_BASE1≤ address ≤ MEMORY_LIMIT1

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. The configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be FFFFFh. Thus, the top of the defined memory address range will be at the top of a 1-MB aligned memory block.

| Bit | Description |
|---|---|
| 15: 4 | **Memory Address Limit 1(MEM_LIMIT1).** Corresponds to A[31:20] of the memory address.<br>Default=0 |
| 3:0 | **Reserved** |

***Note:*** Memory range covered by the MBASE1 and MLIMIT1 registers are used to map non-prefetchable AGP address ranges (typically, where control/status memory-mapped I/O data structures of the graphics controller will reside) and PMBASE 1and PMLIMIT1 are used to map prefetchable address ranges (typically, graphics local memory). 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 host-AGP memory access performance.

## 3.5.19 PMBASE1—Prefetchable Memory Base Address Register (Device 1)

| | |
|---|---|
| Address Offset: | 24–25h |
| Default Value: | FFF0h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-AGP prefetchable memory access routing based on the following formula:

PREFETCHABLE_MEMORY_BASE1≤ address ≤ PREFETCHABLE_MEMORY_LIMIT1

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. The configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be 0. Thus, the bottom of the defined memory address range will be aligned to a 1-MB boundary.

| Bit | Description |
|---|---|
| 15:4 | **Prefetchable Memory Address Base 1(PMEM_BASE1).** Corresponds to A[31:20] of the memory address. |
| 3:0 | **Reserved** |

## 3.5.20 PMLIMIT1—Prefetchable Memory Limit Address Register (Device 1)

| | |
|---|---|
| Address Offset: | 26–27h |
| Default Value: | 0000h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-AGP prefetchable memory access routing based on the following formula:

PREFETCHABLE_MEMORY_BASE1≤ address ≤ PREFETCHABLE_MEMORY_LIMIT1

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. The configuration software must initialize this register. For the purpose of address decode, address, bits A[19:0] are assumed to be FFFFFh. Thus, the top of the defined memory address range will be at the top of a 1-MB aligned memory block.

| Bit | Description |
|---|---|
| 15:4 | **Prefetchable Memory Address Limit 1(PMEM_LIMIT1).** Corresponds to A[31:20] of the memory address. Default=0 |
| 3:0 | **Reserved** |

***Note:*** Prefetchable memory range is supported to allow segregation by the configuration software between the memory ranges that must be defined as UC and the ones that can be designated as a USWC (i.e., prefetchable) from the processor perspective.

### 3.5.21 BCTRL1—PCI-PCI Bridge Control Register (Device 1)

| | |
|---|---|
| Address Offset: | 3Eh |
| Default: | 00h |
| Access: | Read Only, Read/Write |
| Size | 8 bits |

This register provides extensions to the PCICMD1 register that are specific to PCI-PCI bridges. The BCTRL provides additional control for the secondary interface (i.e., AGP) as well as some bits that affect the overall behavior of the "virtual" PCI-PCI bridge in the MCH (e.g., VGA-compatible address ranges mapping).

| Bit | Descriptions |
|---|---|
| 7 | **Fast Back to Back Enable—RO.** Hardwired to 0. Since there is only one target allowed on AGP, this bit is meaningless. The MCH will not generate FB2B cycles in 1x mode, but will generate FB2B cycles in 2x and 4x fast write modes. |
| 6 | **Secondary Bus Reset—RO.** Hardwired to 0. MCH does not support generation of reset via this bit on the AGP.<br>**Note:** That the only way to perform a hard reset of the AGP is via the system reset either initiated by software or hardware via ICH2. |
| 5 | **Master Abort Mode—RO.** This bit is hardwired to 0. This means that when acting as a master on AGP, the MCH will discard data on writes and return all 1s during reads when a Master Abort occurs. |
| 4 | **Reserved** |
| 3 | **VGA Enable (VGAEN1)—R/W.** This bit controls the routing of host-initiated transactions targeting VGA compatible I/O and memory address ranges.<br>1 = The MCH will forward the following host accesses to the AGP:<br>- memory accesses in the range 0A0000h to 0BFFFFh<br>- I/O addresses where A[9:0] are in the ranges 3B0h to 3BBh and 3C0h to 3DFh (inclusive of ISA address aliases - A[15:10] are not decoded)<br>When this bit is set, forwarding of these accesses issued by the host is independent of the I/O address and memory address ranges defined by the previously defined base and limit registers. Forwarding of these accesses is also independent of the settings of bit 2 (ISA Enable) of this register if this bit is 1.<br>0 = VGA compatible memory and I/O range accesses are not forwarded to AGP; rather, they are mapped to primary PCI, unless they are mapped to AGP via I/O and memory range registers defined above (IOBASE1, IOLIMIT1, MBASE1, MLIMIT1, PMBASE1, PMLIMIT1). (default)<br>Refer to the *System Address Map* Chapter of this document for further information.<br>**Note:** This bit must be set to 1 if a video device sits behind this bridge (i.e., video device is on AGP). If there is no video device behind this bridge, then this bit must be set to 0. One of the MCH devices must set this bit. This must be enforced via software. |

| Bit | Descriptions |
|---|---|
| 2 | **ISA Enable—R/W.** This bit modifies the response by the MCH to an I/O access issued by the host that targets ISA I/O addresses. This applies only to I/O addresses that are enabled by the IOBASE and IOLIMIT Registers.<br>1 = Enable. MCH does not forward to AGP 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 IOLIMIT Registers. Instead of going to AGP, these cycles are forwarded to PCI0 where they can be subtractively or positively claimed by the ISA bridge.<br>0 = Disable. All addresses defined by the IOBASE and IOLIMIT for host I/O transactions are mapped to AGP. (default)<br>**Note:** This bit must be set to 1. |
| 1 | **SERR# Enable—R/W.** This bit controls the forwarding of SERR# on the secondary interface to the primary interface.<br>1 = Enable. MCH generates SERR messages to Hub Interface_A when the SERR# pin on AGP bus is asserted and when the messages are enabled by the SERRE bit in the PCICMD1 register.<br>0 = Disable |
| 0 | **Parity Error Response Enable—R/W.** This bit controls the MCH's response to data phase parity errors on AGP.<br>1 = G_PERR# is not implemented by the MCH. However, when this bit is set to 1, address and data parity errors detected on AGP are reported via Hub Interface_A SERR# messaging mechanism, if further enabled by SERRE1.<br>0 = Address and data parity errors on AGP are not reported via the MCH Hub Interface_A SERR# messaging mechanism. Other types of error conditions can still be signaled via SERR# messaging independent of this bit's state. |

### 3.5.22 ERRCMD1—Error Command Register (Device 1)

| | |
|---|---|
| Address Offset: | 40h |
| Default Value: | 00h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:1 | **Reserved** |
| 0 | **SERR on Receiving Target Abort (SERTA).**<br>1 = MCH generates an SERR message over Hub Interface_A upon receiving a target abort on AGP. SERR messaging for Device 1 is globally enabled in the PCICMD1 register.<br>0 = MCH does not assert an SERR message upon receipt of a target abort on AGP. |

## 3.6 Hub Interface_B Bridge Registers (Device 2)

Table 10 provides an address map and describes the access attributes for the Device 2 configuration space.

### Table 10. MCH Configuration Space (Device 2)

| Address Offset | Symbol | Register Name | Default Value | Access |
|---|---|---|---|---|
| 00–01h | VID2 | Vendor Identification | 8086h | RO |
| 02–03h | DID2 | Device Identification | 2533h | RO |
| 04–05h | PCICMD2 | PCI Command Register | 0000h | RO, R/W |
| 06–07h | PCISTS2 | PCI Status Register | 00A0h | RO, R/WC |
| 08h | RID2 | Revision Identification | 03h | RO |
| 09h | — | Reserved | — | — |
| 0Ah | SUBC2 | Sub-Class Code | 04h | RO |
| 0Bh | BCC2 | Base Class Code | 06h | RO |
| 0Ch | — | Reserved | — | — |
| 0Dh | MLT2 | Master Latency Timer | 00h | RO, R/W |
| 0Eh | HDR2 | Header Type | 01h | RO |
| 0F–17h | — | Reserved | — | — |
| 18h | PBUSN2 | Primary Bus Number | 00h | RO |
| 19h | SBUSN2 | Secondary Bus Number | 00h | R/W |
| 1Ah | SUBUSN2 | Subordinate Bus Number | 00h | R/W |
| 1Bh | SMLT2 | Secondary Bus Master Latency Timer | 00h | R/W |
| 1Ch | IOBASE2 | I/O Base Address Register | F0h | RO, R/W |
| 1Dh | IOLIMIT2 | I/O Limit Address Register | 00h | RO, R/W |
| 1E–1Fh | SSTS2 | Secondary Status Register | 02A0h | RO, R/WC |
| 20–21h | MBASE2 | Memory Base Address Register | FFF0h | RO, R/W |
| 22–23h | MLIMIT2 | Memory Limit Address Register | 0000h | RO, R/W |
| 24–25h | PMBASE2 | Prefetchable Memory Base Address Register | FFF0h | RO, R/W |
| 26–27h | PMLIMIT2 | Prefetchable Memory Limit Address Register | 0000h | RO, R/W |
| 28–3Dh | — | Reserved | — | — |
| 3Eh | BCTRL2 | Bridge Control Register | 00h | RO, R/W |
| 3Fh | — | Reserved | — | — |
| 40h | ERRCMD2 | Error Command | 00h | RO, R/W |
| 41–FFh | — | Reserved | — | — |

### 3.6.1 VID2—Vendor Identification Register (Device 2)

| | |
|---|---|
| Address Offset: | 00–01h |
| Default Value: | 8086h |
| Attribute: | Read Only |
| Size: | 16 bits |

The VID register contains the vendor identification number. This 16-bit register combined with the Device Identification register uniquely identifies any PCI device. Writes to this register have no effect.

| Bit | Description |
|---|---|
| 15:0 | **Vendor Identification Number.** This is a 16-bit value assigned to Intel. Intel VID = 8086h. |

### 3.6.2 DID2—Device Identification Register (Device 2)

| | |
|---|---|
| Address Offset: | 02–03h |
| Default Value: | 2533h |
| Attribute: | Read Only |
| Size: | 16 bits |

This 16-bit register, combined with the Vendor Identification register, uniquely identifies any PCI device. Writes to this register have no effect.

| Bit | Description |
|---|---|
| 15:0 | **Device Identification Number**. This is a 16-bit value assigned to the MCH Device 2.<br>MCH Device 2 DID =2533h. |

## 3.6.3 PCICMD2—PCI-PCI Command Register (Device 2)

| | |
|---|---|
| Address Offset: | 04–05h |
| Default: | 0000h |
| Access: | Read Only, Read/Write |
| Size | 16 bits |

| Bit | Descriptions |
|---|---|
| 15:10 | **Reserved.** |
| 9 | **Fast Back-to-Back—RO.** Not implemented; Hardwired to 0. |
| 8 | **SERR Message Enable (SERRE2)—R/W.** This bit is a global enable bit for Device 2 SERR messaging. The MCH does not have an SERR# signal. The MCH communicates the SERR# condition by sending an SERR message to the ICH2.<br>1 = Enable. MCH is enabled to generate SERR messages over the Hub Interface_A for specific Device 2 error conditions.<br>0 = Disable. SERR message is not generated by the MCH for Device 2.<br>**Note:** This bit only controls SERR messaging for the Device 2. Device 0–5 have their own SERRE bit to control error reporting for error conditions occurring on their device. |
| 7 | **Address/Data Stepping—RO.** Not implemented; Hardwired to 0. |
| 6 | **Parity Error Enable (PERRE2)—RO.** Hardwired to 0. Parity checking is not supported on the primary side of this device. |
| 5 | **Reserved.** |
| 4 | **Memory Write and Invalidate Enable—RO.** Not implemented; Hardwired to 0. |
| 3 | **Special Cycle Enable—RO.** Not implemented; Hardwired to 0. |
| 2 | **Bus Master Enable (BME2)—R/W.** Not applicable. However, supported as a read/write bit to avoid the problems with standard PCI-PCI Bridge configuration software. |
| 1 | **Memory Access Enable (MAE2)—R/W.**<br>1 = Enable. Must be set to 1 to enable the Memory and Prefetchable memory address ranges defined in the MBASE2, MLIMIT2, PMBASE2, and PMLIMIT2 Registers.<br>0 = Disable. All of Device 2’s memory space is disabled. |
| 0 | **I/O Access Enable (IOAE2)—R/W.**<br>1 = Enable. Must be set to 1 to enable the I/O address range defined in the IOBASE2 and IOLIMIT2 Registers.<br>0 = Disable. All of Device 2’s I/O space is disabled. |

### 3.6.4 PCISTS2—PCI-PCI Status Register (Device 2)

| | |
|---|---|
| Address Offset: | 06–07h |
| Default Value: | 00A0h |
| Access: | Read Only, Read/Write Clear |
| Size: | 16 bits |

PCISTS2 is a 16-bit status register that reports the occurrence of error conditions associated with the primary side of the “virtual” PCI-PCI bridge in the MCH. Since this device does not physically reside on PCI_A, it reports the optimum operating conditions so that it does not restrict the capability of PCI_A.

| Bit | Descriptions |
|---|---|
| 15 | **Detected Parity Error (DPE2)—RO.** Not implemented; Hardwired to 0. |
| 14 | **Signaled System Error (SSE2)—R/WC.**<br>1 = MCH Device 2 generates an SERR message over the Hub Interface_A for any enabled Device 2 error condition.<br>0 = Software clears this bit by writing a 1 to it. |
| 13 | **Received Master Abort Status (RMAS2)—RO.** Hardwired to 0. The concept of master abort does not exist on primary side of this device. |
| 12 | **Received Target Abort Status (RTAS2)—RO.** Hardwired to 0. The concept of target abort does not exist on primary side of this device. |
| 11 | **Signaled Target Abort Status (STAS2)—RO.** Hardwired to 0. The concept of target abort does not exist on primary side of this device. |
| 10:9 | **DEVSEL# Timing (DEVT2)—RO.** Hardwired to 00. Device 2 uses the fastest possible decode. |
| 8 | **Master Data Parity Error Detected (DPD2)—RO.** Hardwired to 0. Parity is not supported on the primary side of this device. |
| 7 | **Fast Back-to-Back (FB2B2)—RO.** Hardwired to 1. Fast back-to-back writes are always supported on this interface. |
| 6 | **Reserved**. |
| 5 | **66 MHz Capability—RO.** Hardwired to 1. Device is capable of 66 MHz operation. |
| 4:0 | **Reserved**. |

## 3.6.5 RID2—Revision Identification Register (Device 2)

| | |
|---|---|
| Address Offset: | 08h |
| Default Value: | 03h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the revision number of the MCH Device 2. These bits are read only and writes to this register have no effect.

| Bit | Description |
|---|---|
| 7:0 | **Revision Identification Number**. This is an 8-bit value that indicates the revision identification number for the MCH Device 2.<br>A-3 Stepping = 03h. |

## 3.6.6 SUBC2—Sub-Class Code Register (Device 2)

| | |
|---|---|
| Address Offset: | 0Ah |
| Default Value: | 04h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the Sub-Class Code for the MCH Device 2.

| Bit | Description |
|---|---|
| 7:0 | **Sub-Class Code (SUBC2)**. This is an 8-bit value that indicates the category of bridge for the MCH.<br>04h = Host Bridge. |

## 3.6.7 BCC2—Base Class Code Register (Device 2)

| | |
|---|---|
| Address Offset: | 0Bh |
| Default Value: | 06h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the Base Class Code of the MCH Device 2.

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BASEC2)**. This is an 8-bit value that indicates the Base Class Code for the MCH Device 2.<br>06h = Bridge device. |

### 3.6.8 MLT2—Master Latency Timer Register (Device 2)

| | |
|---|---|
| Address Offset: | 0Dh |
| Default Value: | 00h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This functionality is not applicable. It is described here since these bits should be implemented as a read/write to prevent standard PCI-PCI bridge configuration software from getting “confused”.

| Bit | Description |
|---|---|
| 7:3 | **Not applicable but support read/write operations.** Reads return previously written data. |
| 2:0 | **Reserved** |

### 3.6.9 HDR2—Header Type Register (Device 2)

| | |
|---|---|
| Offset: | 0Eh |
| Default: | 01h |
| Access: | Read Only |
| Size: | 8 bits |

This register identifies the header layout of the configuration space. No physical register exists at this location.

| Bit | Descriptions |
|---|---|
| 7:0 | **This read only field always returns 01h when read.** Writes have no effect. |

### 3.6.10 PBUSN2—Primary Bus Number Register (Device 2)

| | |
|---|---|
| Offset: | 18h |
| Default: | 00h |
| Access: | Read Only |
| Size: | 8 bits |

This register identifies that “virtual” PCI-PCI bridge is connected to bus #0.

| Bit | Descriptions |
|---|---|
| 7:0 | **Bus Number.** Hardwired to 0. |

## 3.6.11 SBUSN2—Secondary Bus Number Register (Device 2)

| | |
|---|---|
| Offset: | 19h |
| Default: | 00h |
| Access: | Read /Write |
| Size: | 8 bits |

This register identifies the bus number assigned to the second bus side of the “virtual” PCI-PCI bridge (the Hub Interface_B connection). This number is programmed by the PCI configuration software to allow mapping of configuration cycles to a second bridge device connected to Hub Interface_B.

| Bit | Descriptions |
|---|---|
| 7:0 | **Bus Number.** Programmable. Default = 00h. |

## 3.6.12 SUBUSN2—Subordinate Bus Number Register (Device 2)

| | |
|---|---|
| Offset: | 1Ah |
| Default: | 00h |
| Access: | Read /Write |
| Size: | 8 bits |

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

| Bit | Descriptions |
|---|---|
| 7:0 | **Bus Number.** Programmable. Default = 00. |

## 3.6.13 SMLT2—Secondary Master Latency Timer Register (Device 2)

| | |
|---|---|
| Address Offset: | 1Bh |
| Default Value: | 00h |
| Access: | Read Only |
| Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | **Reserved** |

## 3.6.14 IOBASE2—I/O Base Address Register (Device 2)

| | |
|---|---|
| Address Offset: | 1Ch |
| Default Value: | F0h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This register control the host-to-Hub Interface_B I/O accesses routing based on the following formula:

$$\text{IO\_BASE2} \le \text{address} \le \text{IO\_LIMIT2}$$

Only upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] are treated as 0. Thus, the bottom of the defined I/O address range will be aligned to a 4-KB boundary.

| Bit | Description |
|---|---|
| 7:4 | **I/O Address Base 2.** Corresponds to A[15:12] of the I/O addresses passed by the Device 2 bridge-to-Hub Interface_B. Default=F0h |
| 3:0 | **Reserved** |

## 3.6.15 IOLIMIT2—I/O Limit Address Register (Device 2)

| | |
|---|---|
| Address Offset: | 1Dh |
| Default Value: | 00h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This register control the host-to-Hub Interface_B I/O accesses routing based on the following formula:

$$\text{IO\_BASE2} \le \text{address} \le \text{IO\_LIMIT2}$$

Only upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] are assumed to be FFFh. Thus, the top of the defined I/O address range will be at the top of a 4-KB aligned address block.

| Bit | Description |
|---|---|
| 7:4 | **I/O Address Limit.** Corresponds to A[15:12] of the I/O address limit of Device 2. Default = 0 |
| 3:0 | **Reserved.** Only 16-bit addressing supported. |

### 3.6.16 SSTS2—Secondary PCI-PCI Status Register (Device 2)

| | |
|---|---|
| Address Offset: | 1E–1Fh |
| Default Value: | 02A0h |
| Access: | Read Only, Read/Write Clear |
| Size: | 16 bits |

SSTS2 is a 16-bit status register that reports the occurrence of error conditions associated with the secondary side (i.e., Hub Interface_B side) of the “virtual” PCI-PCI bridge in the MCH.

| Bit | Descriptions |
|---|---|
| 15 | **Detected Parity Error (DPE2)—R/WC.**<br>1 = MCH detected a parity error in the address or data phase of Hub Interface_B bus transactions.<br>0 = Software clears this bit by writing a 1 to it. |
| 14 | **Received System Error (SSE2)—R/WC.**<br>1 = MCH receives an SERR message across the Hub Interface_B.<br>0 = Software clears this bit by writing a 1 to it. |
| 13 | **Received Master Abort Status (RMAS2)—R/WC.**<br>1 = MCH receives a Master Abort completion packet or Master Abort special cycle on Hub Interface_B.<br>0 = Software clears this bit by writing a 1 to it. |
| 12 | **Received Target Abort Status (RTAS2)—R/WC.**<br>1 = MCH receives a Target Abort completion packet or Target Abort special cycle on Hub Interface_B this bit is set.<br>0 = Software clears this bit by writing a 1 to it. |
| 11 | **Signaled Target Abort Status (STAS2)—RO.** Not Implemented; Hardwired to 1. |
| 10:9 | **DEVSEL# Timing (DEVT2)—RO.** Not Applicable. Hardwired to 01b |
| 8 | **Master Data Parity Error Detected (DPD2)—RO.** Not Implemented; Hardwired to 1. |
| 7 | **Fast Back-to-Back (FB2B2)—RO.** Not Implemented; Hardwired to 1. |
| 6 | **Reserved** |
| 5 | **66 MHz Capable (CAP66)—RO.** Hardwired to 1. Hub Interface_B is capable of 66 MHz operation. |
| 4:0 | **Reserved** |

### 3.6.17 MBASE2—Memory Base Address Register (Device 2)

| | |
|---|---|
| Address Offset: | 20–21h |
| Default Value: | FFF0h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-Hub Interface_B non-prefetchable memory access routing based on the following formula:

$$\text{MEMORY\_BASE2} \le \text{address} \le \text{MEMORY\_LIMIT2}$$

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 four bits of this register are read-only and return 0s when read. The configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be 0. Thus, the bottom of the defined memory address range will be aligned to a 1-MB boundary.

| Bit | Description |
|---|---|
| 15: 4 | **Memory Address Base 2 (MEM_BASE2).** Corresponds to A[31:20] of the lower limit memory address that will be passed by the Device 2 to Hub Interface_B. |
| 3:0 | **Reserved** |

## 3.6.18 MLIMIT2—Memory Limit Address Register (Device 2)

| | |
|---|---|
| Address Offset: | 22–23h |
| Default Value: | 0000h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-Hub Interface_B non-prefetchable memory access routing based on the following formula:

MEMORY_BASE2 ≤ address ≤ MEMORY_LIMIT2

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 four bits of this register are read-only and return 0s when read. The configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be FFFFFh. Thus, the top of the defined memory address range will be at the top of a 1-MB aligned memory block.

| Bit | Description |
|---|---|
| 15:4 | **Memory Address Limit 2(MEM_LIMIT2).** Corresponds to A[31:20] of the upper limit memory address that will be passed by the Device 2 to Hub Interface_B.<br>Default=0 |
| 3:0 | **Reserved** |

***Note:*** The memory range covered by MBASE2 and MLIMIT2 Registers are used to map non-prefetchable Hub Interface_B address ranges (typically, where control/status memory-mapped I/O data structures of the graphics controller will reside) and PMBASE2 and PMLIMIT2 are used to map prefetchable address ranges (typically, graphics local memory). 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 host-hub interface memory access performance.

## 3.6.19 PMBASE2—Prefetchable Memory Base Address Register (Device 2)

| | |
|---|---|
| Address Offset: | 24–25h |
| Default Value: | FFF0h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-Hub Interface_B prefetchable memory access routing based on the following formula:

PREFETCHABLE_MEMORY_BASE2 ≤ address ≤ PREFETCHABLE_MEMORY_LIMIT2

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 four bits of this register are read-only and return 0s when read. The configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be 0. Thus, the bottom of the defined memory address range will be aligned to a 1-MB boundary.

| Bit | Description |
|---|---|
| 15: 4 | **Prefetchable Memory Address Base 2 (PMEM_BASE2).** Corresponds to A[31:20] of the memory address. |
| 3:0 | **Reserved** |

### 3.6.20 PMLIMIT2—Prefetchable Memory Limit Address Register (Device 2)

| | |
|---|---|
| Address Offset: | 26–27h |
| Default Value: | 0000h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-Hub Interface_B prefetchable memory access routing based on the following formula:

PREFETCHABLE_MEMORY_BASE2 ≤ address ≤ PREFETCHABLE_MEMORY_LIMIT2

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 four bits of this register are read-only and return zeroes when read. The configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be FFFFFh. Thus, the top of the defined memory address range will be at the top of a 1-MB aligned memory block.

| Bit | Description |
|---|---|
| 15: 4 | **Prefetchable Memory Address Limit 2(PMEM_LIMIT2).** Corresponds to A[31:20] of the memory address.<br>Default=0 |
| 3:0 | **Reserved** |

***Note:*** The prefetchable memory range is supported to allow segregation by the configuration software between the memory ranges that must be defined as UC and the ones that can be designated as a USWC (i.e., prefetchable) from the processor perspective.

### 3.6.21 BCTRL2—PCI-PCI Bridge Control Register (Device 2)

| | |
|---|---|
| Address Offset: | 3Eh |
| Default: | 00h |
| Access: | Read/Write, Read Only |
| Size | 8 bits |

This register provides extensions to the PCICMD2 register that are specific to PCI-PCI bridges. The BCTRL2 provides additional control for the secondary interface (i.e., Hub Interface_B) as well as some bits that affect the overall behavior of the "virtual" PCI-PCI bridge in the MCH (e.g., VGA-compatible address ranges mapping).

| Bit | Descriptions |
|---|---|
| 7 | **Fast Back to Back Enable—RO.** Hardwired to 0. The MCH does not generate fast back-to-back cycles as a master on Hub Interface_B. |
| 6 | **Secondary Bus Reset—RO.** Hardwired to 0. MCH does not support generation of reset via this bit on the Hub Interface_B. |
| 5 | **Master Abort Mode—RO.** Hardwired to 0. As a master on Hub Interface_B, the MCH discards data on writes and returns all 1s during reads when a Master Abort occurs. |
| 4 | **Reserved** |
| 3 | **VGA2 Enable (VGAEN2)—R/W.** This bit controls the routing of host-initiated transactions targeting VGA compatible I/O and memory address ranges.<br>1 = MCH forwards the following host accesses to the Hub Interface_B:<br>• memory accesses in the range 0A0000h–0BFFFFh<br>• I/O addresses where A[9:0] are in the ranges 3B0h to 3BBh and 3C0h to 3DFh (inclusive of ISA address aliases - A[15:10] are not decoded)<br>When this bit is set, forwarding of these accesses issued by the processor is independent of the I/O address and memory address ranges defined by the previously defined base and limit registers. Forwarding of these accesses is also independent of the settings of the bit 2 (ISA Enable) of this register if this bit is 1.<br>0 = VGA compatible memory and I/O range accesses are not forwarded to Hub Interface_B; rather, they are subtractively mapped to primary PCI unless they are mapped to Hub Interface_B via I/O and memory range registers defined above (IOBASE2, IOLIMIT2, MBASE2, MLIMIT2, PMBASE2, PMLIMIT2). (default)<br>Refer to the *System Address Map* chapter of this document for further information.<br>**Note:** This bit must be set to 1 if a video device sits behind this bridge (i.e., video device is on AGP). If there is no video device behind this bridge, this bit must be set to 0. One of the MCH devices must set this bit. This must be enforced via software. |

| Bit | Descriptions |
|---|---|
| 2 | **ISA Enable—R/W.** Modifies the response by the MCH to an I/O access issued by the processor that targets ISA I/O addresses. This applies only to I/O addresses that are enabled by the IOBASE and IOLIMIT registers.<br>1 = Enable. The MCH does not forward to Hub Interface_B 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 IOLIMIT registers. Instead of going to Hub Interface_B, these cycles are forwarded to Hub Interface_A where they can be subtractively or positively claimed by the ISA bridge.<br>0 = All addresses defined by the IOBASE and IOLIMIT for processor I/O transactions will be mapped to Hub Interface_B. (default)<br>**Note:** This bit must be set to 1. |
| 1 | **SERR# Enable—R/W.** This bit enables or disables forwarding of SERR messages from Hub Interface_B-to-Hub Interface_A, where they can be converted into interrupts that are eventually delivered to the processor.<br>1 = Enable<br>0 = Disable |
| 0 | **Parity Error Response Enable—R/W.** This bit controls the MCH's response to data phase parity errors on Hub Interface_B.<br>1 = Address and data parity errors on Hub Interface_B are reported via the Hub Interface_A SERR# messaging mechanism, if further enabled by SERRE2.<br>0 = Address and data parity errors on Hub Interface_B are not reported via the MCH Hub Interface_A SERR# messaging mechanism. Other types of error conditions can still be signaled via SERR# messaging independent of this bit's state. |

### 3.6.22 ERRCMD2—Error Command Register (Device 2)

Address Offset: 40h
Default Value: 00h
Access: Read/Write, Read Only
Size: 8 bits

| Bit | Description |
|---|---|
| 7:4 | **Reserved** |
| 3 | **SERR on Detecting Hub Interface_B Unimplemented Special Cycle (HIDUSCERR).**<br>1 = MCH generates an SERR message over Hub Interface_A when an Unimplemented Special Cycle is received on the hub interface.<br>0 = MCH does not generate an SERR message for this event. SERR messaging for Device 2 is globally enabled in the PCICMD2 register. |
| 2 | **SERR on Generating Hub Interface_B Master Abort (HIDMAERR).**<br>1 = MCH generates an SERR message over Hub Interface_A when an invalid address is received on the hub interface.<br>0 = MCH does not generate an SERR message for this event. SERR messaging for Device 2 is globally enabled in the PCICMD2 register. |
| 1 | **Reserved.** |
| 0 | **SERR on Receiving Target Abort (HISERTA).**<br>1 = MCH generates a SERR message over Hub Interface_A upon receiving a target abort on Hub Interface_B.<br>0 = MCH does not assert an SERR message upon receipt of a target abort on Hub Interface_B. SERR messaging for Device 2 is globally enabled in the PCICMD2 register. |

## 3.7 Hub Interface_C Bridge Registers (Device 3)

Table 11 provides the address map and describes the access attributes for Device 3 configuration space.

### Table 11. MCH Configuration Space (Device 3)

| Address Offset | Symbol | Register Name | Default Value | Access |
|---|---|---|---|---|
| 00–01h | VID3 | Vendor Identification | 8086h | RO |
| 02–03h | DID3 | Device Identification | 2534h | RO |
| 04–05h | PCICMD3 | PCI Command Register | 0000h | RO, R/W |
| 06–07h | PCISTS3 | PCI Status Register | 00A0h | RO, R/WC |
| 08 | RID3 | Revision Identification | 03h | RO |
| 09 | — | Reserved | — | — |
| 0Ah | SUBC3 | Sub-Class Code | 04h | RO |
| 0Bh | BCC3 | Base Class Code | 06h | RO |
| 0Ch | — | Reserved | — | — |
| 0Dh | MLT3 | Master Latency Timer | 00h | RO, R/W |
| 0Eh | HDR3 | Header Type | 01h | RO |
| 0F–17h | — | Reserved | — | — |
| 18h | PBUSN3 | Primary Bus Number | 00h | RO |
| 19h | SBUSN3 | Secondary Bus Number | 00h | R/W |
| 1Ah | SUBUSN3 | Subordinate Bus Number | 00h | R/W |
| 1Bh | SMLT3 | Secondary Bus Master Latency Timer | 00h | RO, R/W |
| 1Ch | IOBASE3 | I/O Base Address Register | F0h | RO, R/W |
| 1Dh | IOLIMIT3 | I/O Limit Address Register | 00h | RO, R/W |
| 1E–1Fh | SSTS3 | Secondary Status Register | 02A0h | RO, R/WC |
| 20–21h | MBASE3 | Memory Base Address Register | FFF0h | RO, R/W |
| 22–23h | MLIMIT3 | Memory Limit Address Register | 0000h | RO, R/W |
| 24–25h | PMBASE3 | Prefetchable Memory Base Address Register | FFF0h | RO, R/W |
| 26–27h | PMLIMIT3 | Prefetchable Memory Limit Address Register | 0000h | RO, R/W |
| 28–3Dh | — | Reserved | — | — |
| 3Eh | BCTRL3 | Bridge Control Register | 00h | RO, R/W |
| 3Fh | — | Reserved | — | — |
| 40h | ERRCMD3 | Error Command | 00h | RO, R/W |
| 41–FFh | — | Reserved | — | — |

### 3.7.1 VID3—Vendor Identification Register (Device 3)

| | |
|---|---|
| Address Offset: | 00–01h |
| Default Value: | 8086h |
| Attribute: | Read Only |
| Size: | 16 bits |

The VID register contains the vendor identification number. This 16-bit register combined with the Device Identification register uniquely identifies any PCI device. Writes to this register have no effect.

| Bit | Description |
|---|---|
| 15:0 | **Vendor Identification Number**. This is a 16-bit value assigned to Intel. Intel VID = 8086h. |

### 3.7.2 DID3—Device Identification Register (Device 3)

| | |
|---|---|
| Address Offset: | 02–03h |
| Default Value: | 2534h |
| Attribute: | Read Only |
| Size: | 16 bits |

This 16-bit register, combined with the Vendor Identification register, uniquely identifies any PCI device. Writes to this register have no effect.

| Bit | Description |
|---|---|
| 15:0 | **Device Identification Number**. This is a 16-bit value assigned to the MCH Device 3.<br>MCH1 Device 3 DID =2534h. |

### 3.7.3 PCICMD3—PCI-PCI Command Register (Device 3)

| | |
|---|---|
| Address Offset: | 04–05h |
| Default: | 0000h |
| Access: | Read Only, Read/Write |
| Size | 16 bits |

| Bit | Descriptions |
|---|---|
| 15:10 | **Reserved** |
| 9 | **Fast Back-to-Back—RO.** Not implemented; Hardwired to 0. |
| 8 | **SERR Message Enable (SERRE3)—RW.** This bit is a global enable bit for Device 3 SERR messaging. The MCH does not have an SERR# signal. The MCH communicates the SERR# condition by sending an SERR message to the ICH2.<br>1 = Enable. MCH is enabled to generate SERR messages over the Hub Interface_A for specific Device 3 error conditions.<br>0 = Disable. SERR message is not generated by the MCH for Device 3.<br>**Note:** This bit only controls SERR messaging for the Device 3. Device 0–5 have their own SERRE bit to control error reporting for error conditions occurring on their device. |
| 7 | **Address/Data Stepping—RO.** Not implemented; Hardwired to 0. |
| 6 | **Parity Error Enable (PERRE3)—RO.** Hardwired to 0. Parity checking is not supported on the primary side of this device. |
| 5 | **Reserved** |
| 4 | **Memory Write and Invalidate Enable—RO.** Not implemented; Hardwired to 0. |
| 3 | **Special Cycle Enable—RO.** Not implemented; Hardwired to 0. |
| 2 | **Bus Master Enable (BME3)—R/W.** Not applicable. However, supported as a read/write bit to avoid the problems with standard PCI-PCI Bridge configuration software. |
| 1 | **Memory Access Enable (MAE3)—R/W.**<br>1 = Enable. Must be set to 1 to enable the memory and prefetchable memory address ranges defined in the MBASE3, MLIMIT3, PMBASE3, and PMLIMIT3 registers.<br>0 = Disable. All of Device 3's memory space is disabled. |
| 0 | **I/O Access Enable (IOAE3)—R/W.**<br>1 = Enable. Must be set to 1 to enable the I/O address range defined in the IOBASE3 and IOLIMIT3 registers.<br>0 = Disable. All of Device 3's I/O space is disabled. |

### 3.7.4 PCISTS3—PCI-PCI Status Register (Device 3)

| | |
|---|---|
| Address Offset: | 06–07h |
| Default Value: | 00A0h |
| Access: | Read Only, Read/Write Clear |
| Size: | 16 bits |

PCISTS3 is a 16-bit status register that reports the occurrence of error conditions associated with the primary side of the “virtual” PCI-PCI bridge in the MCH. Since this device does not physically reside on PCI_A, it reports the optimum operating conditions so that it does not restrict the capability of PCI_A.

| Bit | Descriptions |
|---|---|
| 15 | **Detected Parity Error (DPE3)—RO.** Not Applicable. Hardwired to 0. |
| 14 | **Signaled System Error (SSE3)—R/WC.**<br>1 = MCH Device 3 generates an SERR message over the Hub Interface_A for any enabled Device 3 error condition.<br>0 = Software clears this bit by writing a 1 to it. |
| 13 | **Received Master Abort Status (RMAS3)—RO.** Hardwired to 0. The concept of master abort does not exist on primary side of this device. |
| 12 | **Received Target Abort Status (RTAS3)—RO.** Hardwired to 0. The concept of target abort does not exist on primary side of this device. |
| 11 | **Signaled Target Abort Status (STAS3)—RO.** Hardwired to 0. The concept of target abort does not exist on primary side of this device. |
| 10:9 | **DEVSEL# Timing (DEVT3)—RO.** Hardwired to 00; Device 3 uses the fastest possible decode. |
| 8 | **Master Data Parity Error Detected (DPD3)—RO.** Hardwired to 0. Parity is not supported on the primary side of this device. |
| 7 | **Fast Back-to-Back (FB2B3)—RO.** Hardwired to 1; fast back-to-back writes are always supported on this interface. |
| 6 | **Reserved** |
| 5 | **66 MHz Capability—RO.** Hardwired to a 1; device is capable of 66 MHz operation. |
| 4:0 | **Reserved** |

### 3.7.5 RID3—Revision Identification Register (Device 3)

| | |
|---|---|
| Address Offset: | 08h |
| Default Value: | 03h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the revision number of the MCH Device 3. These bits are read only and writes to this register have no effect.

| Bit | Description |
|---|---|
| 7:0 | **Revision Identification Number**. This is an 8-bit value that indicates the revision identification number for the MCH Device 3.<br>A-3 Stepping = 03h. |

### 3.7.6 SUBC3—Sub-Class Code Register (Device 3)

| | |
|---|---|
| Address Offset: | 0Ah |
| Default Value: | 04h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the Sub-Class Code for the MCH Device 3.

| Bit | Description |
|---|---|
| 7:0 | **Sub-Class Code (SUBC3)**. This is an 8-bit value that indicates the category of bridge for the MCH.<br>04h = Host Bridge |

### 3.7.7 BCC3—Base Class Code Register (Device 3)

| | |
|---|---|
| Address Offset: | 0Bh |
| Default Value: | 06h |
| Access: | Read Only |
| Size: | 8 bits |

This register contains the Base Class Code of the MCH Device 3.

| Bit | Description |
|---|---|
| 7:0 | **Base Class Code (BASEC3)**. This is an 8-bit value that indicates the Base Class Code for the MCH Device 3.<br>06h = Bridge device |

## 3.7.8 MLT3—Master Latency Timer Register (Device 3)

| | |
|---|---|
| Address Offset: | 0Dh |
| Default Value: | 00h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This functionality is not applicable. It is described here since these bits should be implemented as a read/write to prevent standard PCI-PCI bridge configuration software from getting “confused.”

| Bit | Description |
|---|---|
| 7:3 | Not applicable but supports read/write operations. Reads return previously written data. |
| 2:0 | **Reserved** |

## 3.7.9 HDR3—Header Type Register (Device 3)

| | |
|---|---|
| Offset: | 0Eh |
| Default: | 01h |
| Access: | Read Only |
| Size: | 8 bits |

This register identifies the header layout of the configuration space. No physical register exists at this location.

| Bit | Descriptions |
|---|---|
| 7:0 | This read only field always returns 01h when read. Writes have no effect. |

## 3.7.10 PBUSN3—Primary Bus Number Register (Device 3)

| | |
|---|---|
| Offset: | 18h |
| Default: | 00h |
| Access: | Read Only |
| Size: | 8 bits |

This register identifies that “virtual” PCI-PCI bridge is connected to bus #0.

| Bit | Descriptions |
|---|---|
| 7:0 | **Bus Number.** Hardwired to 0. |

## 3.7.11 SBUSN3—Secondary Bus Number Register (Device 3)

| | |
|---|---|
| Offset: | 19h |
| Default: | 00h |
| Access: | Read /Write |
| Size: | 8 bits |

This register identifies the bus number assigned to the second bus side of the “virtual” PCI-PCI bridge (the Hub Interface_C connection). This number is programmed by the PCI configuration software to allow mapping of configuration cycles to a second bridge device connected to Hub Interface_C.

| Bit | Descriptions |
|---|---|
| 7:0 | **Bus Number.** Programmable. Default = 00h. |

## 3.7.12 SUBUSN3—Subordinate Bus Number Register (Device 3)

| | |
|---|---|
| Offset: | 1Ah |
| Default: | 00h |
| Access: | Read /Write |
| Size: | 8 bits |

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

| Bit | Descriptions |
|---|---|
| 7:0 | **Bus Number.** Programmable. Default = 00. |

## 3.7.13 SMLT3—Secondary Master Latency Timer Register (Device 3)

| | |
|---|---|
| Address Offset: | 1Bh |
| Default Value: | 00h |
| Access: | Read Only |
| Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:0 | **Reserved** |

### 3.7.14 IOBASE3—I/O Base Address Register (Device 3)

| | |
|---|---|
| Address Offset: | 1Ch |
| Default Value: | F0h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This register control the host-to-Hub Interface_C I/O access routing based on the following formula:

IO_BASE3 ≤ address ≤ IO_LIMIT3

Only upper 4 bits are programmable. For the purpose of address decode, address bits A[11:0] are treated as 0. Thus, the bottom of the defined I/O address range will be aligned to a 4-KB boundary.

| Bit | Description |
|---|---|
| 7:4 | **I/O Address Base 3.** Corresponds to A[15:12] of the I/O addresses passed by the Device 3 bridge to Hub Interface_C. Default=F0h |
| 3:0 | **Reserved** |

### 3.7.15 IOLIMIT3—I/O Limit Address Register (Device 3)

| | |
|---|---|
| Address Offset: | 1Dh |
| Default Value: | 00h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

This register controls the host-to-Hub Interface_C I/O access routing based on the following formula:

IO_BASE3 ≤ address ≤ IO_LIMIT3

Only upper four bits are programmable. For the purpose of address decode, address bits A[11:0] are assumed to be FFFh. Thus, the top of the defined I/O address range will be at the top of a 4-KB aligned address block.

| Bit | Description |
|---|---|
| 7:4 | **I/O Address Limit.** Corresponds to A[15:12] of the I/O address limit of Device 3. Default = 0 |
| 3:0 | **Reserved.** Only 16-bit addressing supported. |

### 3.7.16 SSTS3—Secondary PCI-PCI Status Register (Device 3)

| | |
|---|---|
| Address Offset: | 1E–1Fh |
| Default Value: | 02A0h |
| Access: | Read Only, Read/Write Clear |
| Size: | 16 bits |

SSTS3 is a 16-bit status register that reports the occurrence of error conditions associated with secondary side (i.e., Hub Interface_C side) of the “virtual” PCI-PCI bridge in the MCH.

| Bit | Descriptions |
|---|---|
| 15 | **Detected Parity Error (DPE3)—R/WC.**<br>1 = MCH detected of a parity error in the address or data phase of Hub Interface_C bus transactions.<br>0 = Software clears this bit by writing a 1 to this bit. |
| 14 | **Received System Error (SSE3)—R/WC.**<br>1 = MCH receives a SERR message across the Hub Interface_C.<br>0 = Software clears this bit by writing a 1 to this bit. |
| 13 | **Received Master Abort Status (RMAS3)—R/WC.**<br>1 = MCH receives a Master Abort completion packet or Master Abort special cycle on Hub Interface_C this bit is set.<br>0 = Software clears this bit by writing a 1 to this bit. |
| 12 | **Received Target Abort Status (RTAS3)—R/WC.**<br>1 = MCH receives a Target Abort completion packet or Target Abort special cycle on Hub Interface_C.<br>0 = Software clears this bit by writing a 1 to this bit. |
| 11 | **Signaled Target Abort Status (STAS3)—RO.** Not Applicable. Hardwired to 0. |
| 10:9 | **DEVSEL# Timing (DEVT3)—RO.** Not Applicable. Hardwired to 01b |
| 8 | **Master Data Parity Error Detected (DPD3)—RO.** Not Applicable. Hardwired to 0. |
| 7 | **Fast Back-to-Back (FB2B3)—RO.** Not Applicable. Hardwired to 1. |
| 6 | **Reserved** |
| 5 | **66 MHz Capable (CAP66)—RO.** This bit is hardwired to 1 to indicate that Hub Interface_C is capable of 66 MHz operation. |
| 4:0 | **Reserved** |

## 3.7.17 MBASE3—Memory Base Address Register (Device 3)

| | |
|---|---|
| Address Offset: | 20–21h |
| Default Value: | FFF0h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-Hub Interface_C non-prefetchable memory access routing based on the following formula:

MEMORY_BASE3 ≤ address ≤ MEMORY_LIMIT3

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 four bits of this register are read-only and return 0s when read. The configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be 0. Thus, the bottom of the defined memory address range will be aligned to a 1-MB boundary.

| Bit | Description |
|---|---|
| 15: 4 | **Memory Address Base 3 (MEM_BASE3).** Corresponds to A[31:20] of the lower limit memory address that will be passed by the Device 3 to Hub Interface_C. |
| 3:0 | **Reserved** |

## 3.7.18 MLIMIT3—Memory Limit Address Register (Device 3)

| | |
|---|---|
| Address Offset: | 22–23h |
| Default Value: | 0000h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-Hub Interface_C non-prefetchable memory access routing based on the following formula:

MEMORY_BASE3 ≤ address ≤ MEMORY_LIMIT3

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 four bits of this register are read-only and return 0s when read. The configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be FFFFFh. Thus, the top of the defined memory address range will be at the top of a 1-MB aligned memory block.

| Bit | Description |
|---|---|
| 15:4 | **Memory Address Limit 3(MEM_LIMIT3).** Corresponds to A[31:20] of the upper limit memory address that will be passed by the Device 3 to Hub Interface_C. Default = 0 |
| 3:0 | **Reserved** |

***Note:*** Memory range covered by MBASE3 and MLIMIT3 registers are used to map non-prefetchable Hub Interface_C address ranges (typically, where control/status memory-mapped I/O data structures of the graphics controller will reside) and PMBASE3 and PMLIMIT3 are used to map prefetchable address ranges (typically, graphics local memory). 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 host-hub interface memory access performance.

## 3.7.19 PMBASE3—Prefetchable Memory Base Address Register (Device 3)

| | |
|---|---|
| Address Offset: | 24–25h |
| Default Value: | FFF0h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-Hub Interface_C prefetchable memory access routing based on the following formula:

PREFETCHABLE_MEMORY_BASE3 ≤ address ≤ PREFETCHABLE_MEMORY_LIMIT3

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 four bits of this register are read-only and return 0s when read. The configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be 0. Thus, the bottom of the defined memory address range will be aligned to a 1-MB boundary.

| Bit | Description |
|---|---|
| 15: 4 | **Prefetchable Memory Address Base 3(PMEM_BASE3).** Corresponds to A[31:20] of the memory address. |
| 3:0 | **Reserved** |

## 3.7.20 PMLIMIT3—Prefetchable Memory Limit Address Register (Device 3)

| | |
|---|---|
| Address Offset: | 26–27h |
| Default Value: | 0000h |
| Access: | Read/Write, Read Only |
| Size: | 16 bits |

This register controls the host-to-Hub Interface_C prefetchable memory access routing based on the following formula:

PREFETCHABLE_MEMORY_BASE3 ≤ address ≤ PREFETCHABLE_MEMORY_LIMIT3

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 four bits of this register are read-only and return 0s when read. The configuration software must initialize this register. For the purpose of address decode, address bits A[19:0] are assumed to be FFFFFh. Thus, the top of the defined memory address range will be at the top of a 1-MB aligned memory block.

| Bit | Description |
|---|---|
| 15: 4 | **Prefetchable Memory Address Limit 3(PMEM_LIMIT3).** Corresponds to A[31:20] of the memory address. Default=0 |
| 3:0 | **Reserved** |

**NOTE:** The prefetchable memory range is supported to allow segregation by the configuration software between the memory ranges that must be defined as UC and the ones that can be designated as a USWC (i.e., prefetchable) from the processor perspective.

### 3.7.21 BCTRL3—PCI-PCI Bridge Control Register (Device 3)

| | |
|---|---|
| Address Offset: | 3Eh |
| Default: | 00h |
| Access: | Read/Write, Read Only |
| Size | 8 bits |

This register provides extensions to the PCICMD3 register that are specific to PCI-PCI bridges. The BCTRL3 provides additional control for the secondary interface (i.e., Hub Interface_C) as well as some bits that affect the overall behavior of the “virtual” PCI-PCI bridge in the MCH (e.g., VGA-compatible address ranges mapping).

| Bit | Descriptions |
|---|---|
| 7 | **Fast Back to Back Enable—RO.** Hardwired to 0. The MCH does not generate fast back-to-back cycles as a master on Hub Interface_C. |
| 6 | **Secondary Bus Reset—RO.** Hardwired to 0. MCH does not support generation of reset via this bit on Hub Interface_C. |
| 5 | **Master Abort Mode—RO.** Hardwired to 0. As a master on Hub Interface_C the MCH will discard data on writes and return all 1s during reads when a Master Abort occurs. |
| 4 | **Reserved** |
| 3 | **VGA3 Enable (VGAEN3)—R/W.** This bit controls the routing of host-initiated transactions targeting VGA compatible I/O and memory address ranges.<br>1 = The MCH forwards the following host accesses to Hub Interface_C:<br>• memory accesses in the range 0A0000h–0BFFFFh<br>• I/O addresses where A[9:0] are in the ranges 3B0h to 3BBh and 3C0h to 3DFh (inclusive of ISA address aliases - A[15:10] are not decoded)<br>When this bit is set, forwarding of these accesses issued by the processor is independent of the I/O address and memory address ranges defined by the previously defined base and limit registers. Forwarding of these accesses is also independent of the settings of the bit 2 (ISA Enable) of this register, if this bit is 1.<br>0 = VGA compatible memory and I/O range accesses are not forwarded to Hub Interface_C; rather, they are subtractively mapped to primary PCI unless they are mapped to Hub Interface_C via I/O and memory range registers defined above (IOBASE3, IOLIMIT3, MBASE3, MLIMIT3, PMBASE3, PMLIMIT3). (default)<br>Refer to the *System Address Map* chapter of this document for further information.<br>**Note:** This bit must be set to 1 if a video device sits behind this bridge (i.e., video device is on AGP). If there is no video device behind this bridge, this bit must be set to 0. One of the MCH devices must set this bit. This must be enforced via software. |

| Bit | Descriptions |
|---|---|
| 2 | **ISA Enable—R/W.** This bit modifies the response by the MCH to an I/O access issued by the processor that targets ISA I/O addresses. This applies only to I/O addresses that are enabled by the IOBASE and IOLIMIT registers.<br>1 = Enable. MCH does not forward to Hub Interface_C 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 IOLIMIT registers. Instead of going to Hub Interface_C these cycles will be forwarded to Hub Interface_A where they can be subtractively or positively claimed by the ISA bridge.<br>0 = All addresses defined by the IOBASE and IOLIMIT for processor I/O transactions will be mapped to Hub Interface_C. (default)<br>**Note:** This bit must be set to 1. |
| 1 | **SERR# Enable—R/W.** This bit enables or disables forwarding of SERR messages from Hub Interface_C-to-Hub Interface_A, where they can be converted into interrupts that are eventually delivered to the processor.<br>1 = Enable<br>0 = Disable |
| 0 | **Parity Error Response Enable—R/W.** This bit controls the MCH's response to data phase parity errors on Hub Interface_C.<br>1 = Address and data parity errors on Hub Interface_C are reported via the Hub Interface_A SERR# messaging mechanism, if further enabled by SERRE3.<br>0 = Address and data parity errors on Hub Interface_C are not reported via the MCH Hub Interface_A SERR# messaging mechanism. Other types of error conditions can still be signaled via SERR# messaging independent of this bit's state. |

### 3.7.22 ERRCMD3—Error Command Register (Device 3)

| | |
|---|---|
| Address Offset: | 40h |
| Default Value: | 00h |
| Access: | Read/Write, Read Only |
| Size: | 8 bits |

| Bit | Description |
|---|---|
| 7:4 | **Reserved** |
| 3 | **SERR on Detecting Hub Interface_C Unimplemented Special Cycle (HIDUSCERR).**<br>1 = MCH generates an SERR message over Hub Interface_A when an Unimplemented Special Cycle is received on the hub interface.<br>0 = MCH does not generate an SERR message for this event. SERR messaging for Device 3 is globally enabled in the PCICMD3 register. |
| 2 | **SERR on Generating Hub Interface_C Master Abort (HIDMAERR).**<br>1 = MCH generates an SERR message over Hub Interface_A when an invalid address is received on the hub interface.<br>0 = MCH does not generate an SERR message for this event. SERR messaging for Device 3 is globally enabled in the PCICMD3 register. |
| 1 | **Reserved.** |
| 0 | **SERR on Receiving Target Abort (HISERTA).**<br>1 = MCH generates an SERR message over Hub Interface_A upon receiving a target abort on Hub Interface_C.<br>0 = MCH does not assert an SERR message upon receipt of a target abort on Hub Interface_C. SERR messaging for Device 3 is globally enabled in the PCICMD3 register. |

This page is intentionally left blank.

# 4 System Address Map

A system based on the Intel 82860 MCH supports 16 GB of addressable memory space and 64 KB+3 of addressable I/O space. The I/O and memory spaces are divided by system configuration software into regions. The memory ranges are useful either as system memory or as specialized memory, while the I/O regions are used solely to control the operation of devices in the system.

When the MCH receives a write request whose address targets an invalid space, the data is ignored. For reads, the MCH responds by returning all zeros on the requesting interface.

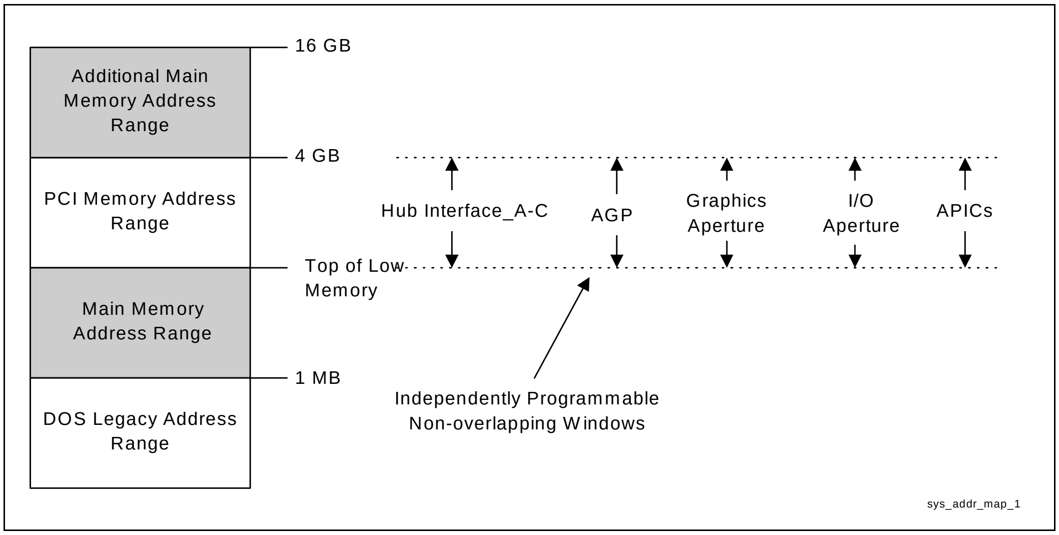
## 4.1 Memory Address Ranges

The system memory map is divided into three categories:

- **High Memory Range (above 4 GB) –** The first is DRAM only, and exists between 4 GB and 16 GB (bit 32 of the address is active)
- **Extended Memory Range (1 MB to 4 GB)** - The second is extended memory, existing between 1 MB and 4 GB. It contains a 32-bit memory space, which is used for mapping PCI, AGP, APIC, SMRAM, and BIOS memory spaces.
- **DOS Compatible Area (below 1 MB)** - The final range is a DOS legacy space, which is used for BIOS and legacy devices on the LPC interface.

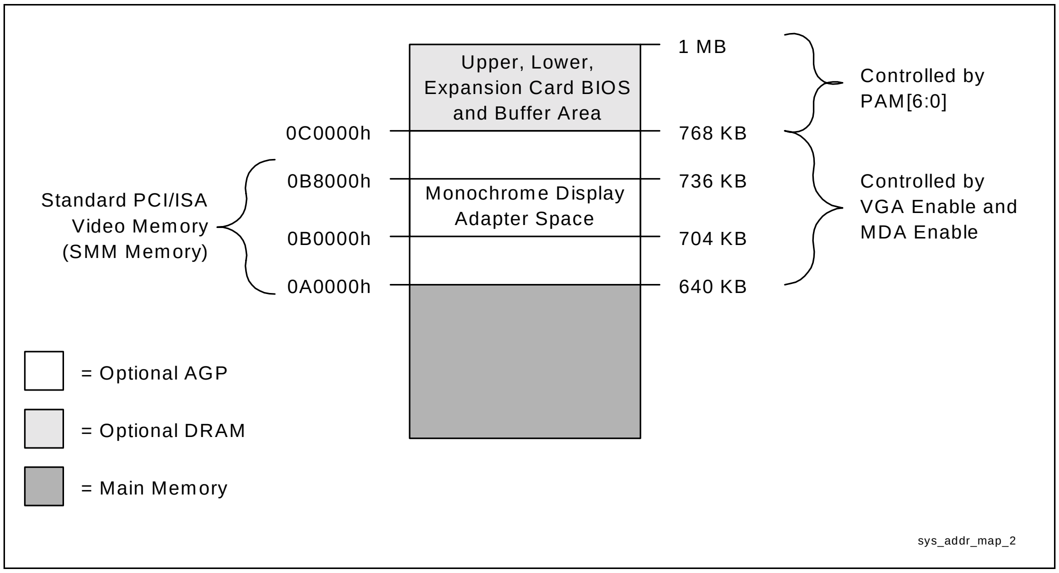
Figure 4 shows the major portions of the system address Map. Figure 5 and Figure 6 provide detailed address maps for the DOS memory address range and extended memory address range.

Figure 4. System Address Map

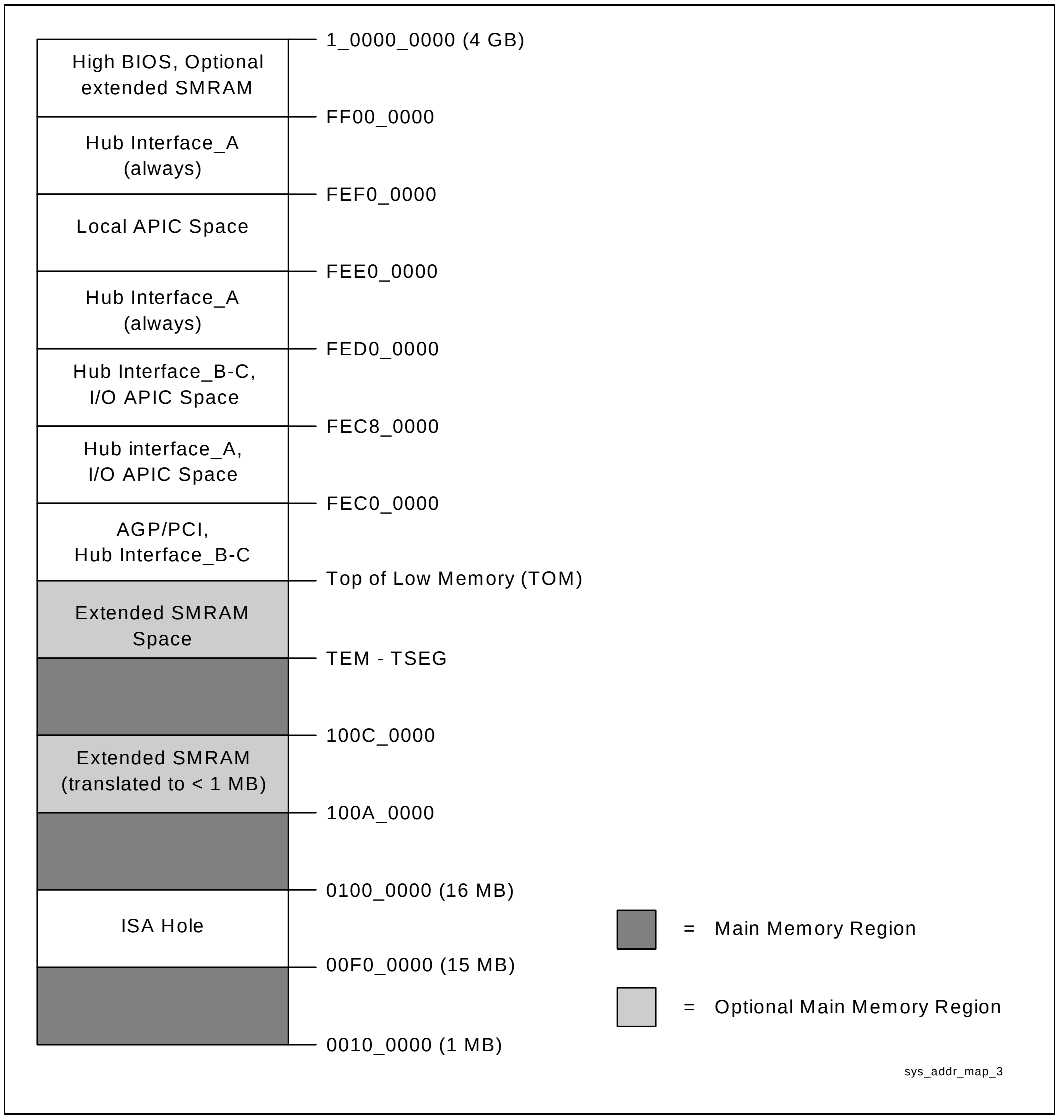


These address ranges are always mapped to system memory, regardless of the system configuration. Memory may be taken out of the main memory segment for use by System Management Mode (SMM) hardware and software. The Top of Low Memory (TOM) register defines the top of Main Memory. Note that the address of the highest 16 MB quantity of valid memory in the system is placed into the GBA15 register. For memory populations < 3 GB, this value will be the same as the one programmed into the TOM register. For other memory configurations, the two are unlikely to be the same, since the PCI configuration portion of the BIOS software will program the TOM register to the maximum value that is less than the amount of memory in the system and that allows enough room for all populated PCI devices.

### Figure 5. Detailed DOS Compatible Area Address Map


Upper, Lower, Expansion Card BIOS and Buffer Area
1 MB
768 KB
736 KB
704 KB
640 KB
0C0000h
0B8000h
0B0000h
0A0000h
Monochrome Display Adapter Space
Standard PCI/ISA Video Memory (SMM Memory)
Controlled by PAM[6:0]
Controlled by VGA Enable and MDA Enable
= Optional AGP
= Optional DRAM
= Main Memory
sys_addr_map_2

### Figure 6. Detailed Extended Memory Range Address Map



## 4.1.1 VGA and MDA Memory Space

Video cards use these legacy address ranges to map a frame buffer or a character-based video buffer. The address ranges in this memory space are:

| | |
|---|---|
| VGAA | 0_000A_0000h to 0_000A_FFFFh |
| MDA | 0_000B_0000h to 0_000B_7FFFh |
| VGAB | 0_000B_8000h to 0_000B_FFFFh |

By default, accesses to these ranges are forwarded to Hub Interface_A. However, if the VGA_EN bit is set in the BCTRL1–3 configuration registers, then transactions within the VGA and MDA spaces are sent to AGP or Hub Interface_B–C, respectively. **Note that the VGA_EN bit may be set in one and only one of the BCTRL registers. Software must not set more than one of the VGA_EN bits.** If the configuration bit MCHCFG.MDAP is set, accesses that fall within the MDA range will be sent to Hub Interface_A without regard for the VGAEN bits.

If the configuration bit MCHCFG.MDAP is set, then accesses that fall within the MDA range will be sent to Hub Interface_A without regard for the VGAEN bits. Legacy support requires the ability to have a second graphics controller (monochrome) in the system. In an Intel 82860 MCH system, accesses in the standard VGA range are forwarded to the AGP or Hub Interface_B–C (depending on configuration bits). Since the monochrome adapter may be on Hub Interface_A (or ISA bus) the MCH must decode cycles in the MDA range and forward them to Hub Interface_A. This capability is controlled by a configuration bit (MDAP bit). In addition to the memory range B0000h to B7FFFh, the MCH decodes I/O cycles at 3B4h, 3B5h, 3B8h, 3B9h, 3Bah, and 3BFh and forwards them to Hub Interface_A.

An optimization allows the system to reclaim the memory displaced by these regions. If SMM memory space is enabled by SMRAM.G_SMRARE and either the SMRAM.D_OPEN bit is set or the processor bus receives an SMM-encoded request for code (not data), then the transaction is steered to system memory rather than Hub Interface_A. Under these conditions, both of the VGAEN bits and the MDAP bit are ignored.

## 4.1.2 PAM Memory Spaces

The address ranges in this memory space are:

| | |
|---|---|
| PAMC0 | 0_000C_0000h to 0_000C_3FFFh |
| PAMC4 | 0_000C_4000h to 0_000C_7FFFh |
| PAMC8 | 0_000C_8000h to 0_000C_BFFFh |
| PAMCC | 0_000C_C000h to 0_000C_FFFFh |
| PAMD0 | 0_000D_0000h to 0_000D_3FFFh |
| PAMD4 | 0_000D_4000h to 0_000D_7FFFh |
| PAMD8 | 0_000D_8000h to 0_000D_BFFFh |
| PAMDC | 0_000D_C000h to 0_000D_FFFFh |
| PAME0 | 0_000E_0000h to 0_000E_3FFFh |
| PAME4 | 0_000E_4000h to 0_000E_7FFFh |
| PAME8 | 0_000E_8000h to 0_000E_BFFFh |
| PAMEC | 0_000E_C000h to 0_000E_FFFFh |
| PAMF0 | 0_000F_0000h to 0_000F_FFFFh |

The 256-KB PAM region is divided into three parts:

- **ISA expansion region**, a 128-KB area between 0_000C_0000h – 0_000D_FFFFh
- **Extended BIOS region**, a 64-KB area between 0_000E_0000h – 0_000E_FFFFh
- **System BIOS region**, a 64-KB area between 0_000F_0000h – 0_000F_FFFFh.

The ISA expansion region is 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 MCH and are subtractively decoded to ISA space.

The extended System BIOS region is divided into four 16 KB segments. Each segment can be assigned independent read and write attributes so it can be mapped either to main DRAM or to Hub Interface_A. Typically, this area is used for RAM or ROM.

The system BIOS region is a single 64-KB segment. This segment can be assigned read and write attributes. It is by default (after reset) read/write disabled and cycles are forwarded to Hub Interface_A. By manipulating the read/write attributes, the MCH can “shadow” BIOS into the main DRAM.

## 4.1.3 ISA Hole Memory Space

BIOS software may optionally open a “window” between 15 MB and 16 MB (0_00F0_0000h to 0_00FF_FFFFh) that relays transactions to Hub Interface_A instead of completing them with a system memory access. This window is opened with the FDHC.HEN configuration field.

### 4.1.4 TSEG SMM Memory Space

The TSEG SMM space (TOM – TSEG to TOM) allows system management software to partition a region of main memory just below the top of low memory (TOM) that is accessible only by system management software. This region may be 128 KB, 256 KB, 512 KB, or 1 MB, depending on the ESMRAMC.TSEG_SZ field. SMM memory is globally enabled by SMRAM.G_SMRARE. Requests may access SMM system memory when either SMM space is open (SMRAM.D_OPEN) or the MCH receives an SMM code request on its processor bus. In order to access the TSEG SMM space, the TSEG must be enabled by ESMRAMC.T_EN. When all of these conditions are met, a processor bus access to the TSEG space (between TOM-TSEG and TOM) is sent to system memory. If the high SMRAM is not enabled or if the TSEG is not enabled, all memory requests from all interfaces are forwarded to system memory. If the TSEG SMM space is enabled, and an agent attempts a non-SMM access to TSEG space, the transaction is specially terminated.

Hub interface and AGP originated accesses are not allowed to SMM space.

### 4.1.5 I/O APIC Memory Space

The I/OAPIC spaces are used to communicate with I/O APIC interrupt controllers that may be populated on Hub Interface_A–C. Since it is difficult to relocate an interrupt controller using plug-and-play software, fixed address decode regions have been allocated for them. The address ranges are:

| | |
|---|---|
| • IOAPIC0 (Hub Interface_A) | 0_FEC0_0000h to 0_FEC7_FFFFh |
| • IOAPIC1 (Hub Interface_B) | 0_FEC8_0000h to 0_FEC8_0FFFh |
| • IOAPIC2 (Hub Interface_C) | 0_FEC8_1000h to 0_FEC8_1FFFh |

Processor accesses to the IOAPIC0 region are always sent to Hub Interface_A. Processor accesses to the IOAPIC1 region are always sent to Hub Interface_B and so on.

### 4.1.6 System Bus Interrupt Memory Space

The system bus interrupt space (0_FEE0_0000h to 0_FEEF_FFFFh) is the address used to deliver interrupts to the system bus. Any device on AGP or Hub Interface_A–C may issue a memory write to 0FEEx_xxxxh. The MCH will forward this memory write, along with the data, to the system bus as an Interrupt Message Transaction. The MCH terminates the system bus transaction by providing the response and asserting TRDY#. This memory write cycle does not go to DRAM.

### 4.1.7 High SMM Memory Space

The HIGHSMM space (0_FEDA_0000h to 0_FEDB_FFFFh) allows cacheable access to the compatible SMM space by re-mapping valid SMM accesses between 0_FEDA_0000 and 0_FEDB_FFFF to accesses between 0_000A_0000 and 0_000B_FFFF. The accesses are remapped when SMRAM space is enabled, an appropriate access is detected on the processor bus, and when ESMRAMC.H_SMRAME allows access to high SMRAM space. SMM memory accesses from any hub interface or AGP are specially terminated: reads are provided with the value from address 0 while writes are ignored entirely.

### 4.1.8 AGP Aperture Space (Device 0 BAR)

Processors and AGP devices communicate through a special buffer called the “graphics aperture” (located at APBASE to APBASE + APSIZE). This aperture acts as a window into main memory and is defined by the APBASE and APSIZE configuration registers of the MCH. Note that the AGP aperture must be above the TOM and must not intersect with any other address space.

### 4.1.9 AGP Memory and Prefetchable Memory

Plug-and-play software configures the AGP memory window to provide enough memory space for the devices behind this PCI-to-PCI bridge. Accesses whose addresses fall within this window are decoded and forwarded to AGP for completion. The address ranges are:

- M1 MBASE1 to MLIMIT1
- PM1 PMBASE1 to PMLIMIT1

Note that these registers must be programmed with values that place the AGP memory space window between the value in the TOM register and 4 GB. In addition, neither region should overlap with any other fixed or relocatable area of memory.

### 4.1.10 Hub Interface_B Memory and Prefetchable Memory

Plug-and-play software configures the Hub Interface_B memory window to provide enough memory space for the devices behind this PCI-to-PCI bridge. Accesses whose addresses fall within this window are decoded and forwarded to Hub Interface_B for completion. The address ranges are:

- M2 MBASE2 to MLIMIT2
- PM2 PMBASE2 to PMLIMIT2

Note that these registers must be programmed with values that place the Hub Interface_B memory space window between the value in the TOM register and 4 GB. In addition, neither region should overlap with any other fixed or relocatable area of memory.

### 4.1.11 Hub Interface_C Memory and Prefetchable Memory

Plug-and-play software configures the Hub Interface_C memory window to provide enough memory space for the devices behind this PCI-to-PCI bridge. Accesses whose addresses fall within this window are decoded and forwarded to Hub Interface_C for completion. The address ranges are:

- M3 MBASE3 to MLIMIT3
- PM3 PMBASE3 to PMLIMIT3

Note that these registers must be programmed with values that place the Hub Interface_C memory space window between the value in the TOM register and 4 GB. In addition, neither region should overlap with any other fixed or relocatable area of memory.

### 4.1.12 Hub Interface_A Subtractive Decode

All accesses that fall between the values programmed into the TOM register and 4 GB are subtractively decoded and forwarded to Hub Interface_A, if they do not decode to a space that corresponds to another device.

## 4.2 AGP Memory Address Ranges

The MCH can be programmed to direct memory accesses to the AGP bus interface when addresses are within either of two ranges specified via registers in MCH Device 1 configuration space. The first range is controlled via the Memory Base (MBASE1) register and Memory Limit (MLIMIT1) register. The second range is controlled via the Prefetchable Memory Base (PMBASE1) register and Prefetchable Memory Limit (PMLIMIT1) register.

The MCH positively decodes memory accesses to AGP memory address space as defined by the following equations:

- Memory_Base_Address ≤ Address ≤ Memory_Limit_Address
- Prefetchable_Memory_Base_Address ≤ Address ≤ Prefetchable_Memory_Limit_Address

The plug-and-play configuration software programs the effective size of the range and it depends on the size of memory claimed by the AGP device.

***Note:*** That the MCH Device 1 memory range registers described above are used to allocate memory address space for any devices sitting on AGP bus that require such a window.

### 4.2.1 AGP DRAM Graphics Aperture

Memory-mapped, graphics data structures can reside in a *Graphics Aperture* to main DRAM memory. This aperture is an address range defined by the APBASE and APSIZE configuration registers of the MCH Device 0. The APBASE register follows the standard Base Address register template as defined by the PCI 2.1 specification. The size of the range claimed by the APBASE is programmed via “back-end” register APSIZE (programmed by the chipset specific BIOS before plug-and-play session is performed). APSIZE allows the BIOS software to pre-configure the aperture size to be 4 MB, 8 MB, 16 MB, 32 MB, 64 MB, 128 MB or 256 MB. By programming APSIZE to specific size, the corresponding lower bits of APBASE are forced to 0 (behave as hardwired). Default value of APSIZE forces an aperture size of 256 MB. The aperture address range is naturally aligned.

Accesses within the aperture range are forwarded to the main DRAM subsystem. The MCH will translate the originally issued addresses via a translation table maintained in main memory. The range should be programmed as non-cacheable in the processor caches.

***Note:*** Plug-and-play software configuration model does not allow overlap of different address ranges. Therefore the AGP Graphics Aperture and AGP Memory Address Range are independent address ranges that may abut, but cannot overlap one another.

## 4.3 System Management Mode (SMM) Memory Range

The MCH supports the use of main memory as System Management RAM (SMRAM) enabling the use of SMM. The MCH supports two SMRAM options: Compatible SMRAM (C_SMRAM) and Extended SMRAM (E_SMRAM). System Management RAM (SMRAM) space provides a memory area that is available for the SMI handler's and code and data storage. This memory resource is normally hidden from the system OS so that the processor has immediate access to this memory space upon entry to SMM. The MCH provides three SMRAM options:

- Below 1 MB option that supports compatible SMI handlers.
- Above 1 MB option that allows new SMI handlers to execute with write-back cacheable SMRAM.
- Optional larger write-back cacheable T_SEG area from 128 KB to 1 MB in size above 1 MB is reserved from the highest area in system DRAM memory. The above 1-MB solutions require changes to compatible SMRAM handlers' code to properly execute above 1 MB.

***Note:*** Masters from the hub interface and AGP are not allowed to access the SMM space.

### 4.3.1 SMM Space Definition

The addressed SMM space is defined as the range of bus addresses used by the processor to access SMM space. DRAM SMM space is defined as the range of physical DRAM memory locations containing the SMM code. SMM space can be accessed at one of three transaction address ranges: Compatible, High and TSEG. The Compatible and TSEG SMM space is not remapped; therefore, the addressed and DRAM SMM space is the same address range. Since the High SMM space is remapped, the addressed and DRAM SMM space is a different address range. Note that the High DRAM space is the same as the Compatible Transaction Address space. Therefore, Table 12 describes three unique address ranges:

- Compatible Transaction Address
- High Transaction Address
- TSEG Transaction Address

**Table 12. SMM Space Address Ranges**

| SMM Space Enabled | Transaction Address Space | DRAM Space (DRAM) |
|---|---|---|
| Compatible | A0000h to BFFFFh | A0000h to BFFFFh |
| High | 0FEDA0000h to 0FEDBFFFFh | A0000h to BFFFFh |
| TSEG | (TOM-TSEG_SZ) to TOM | (TOM-TSEG_SZ) to TOM |

**NOTES:**

1. High SMM: This is different than in previous chip sets. In previous chip sets the High segment was the 384-KB region from A0000h to FFFFFh. However, C0000h to FFFFFh was not practically useful so it is deleted in MCH.
2. TSEG SMM: This is different than in previous chipsets. In previous chipsets the TSEG address space was offset by 256 MB to allow for simpler decoding and the TSEG was remapped to just under the TOM. In the MCH 256 MB do not offset the TSEG region and it is not remapped.

### 4.3.2 SMM Space Restrictions

If any of the following conditions are violated, the results of SMM accesses are unpredictable and may cause the system to hang:

- The Compatible SMM space **must not** be setup as cacheable.
- High or TSEG SMM transaction address space **must not** overlap address space assigned to system DRAM, the AGP aperture range, or to any "PCI" devices (including hub interface and AGP devices). This is a BIOS responsibility.
- Both D_OPEN and D_CLOSE **must not** be set to 1 at the same time.
- When TSEG SMM space is enabled, the TSEG space **must not** be reported to the OS as available main memory. This is a BIOS responsibility.
- Any address translated through the AGP Aperture GTLB **must not** target main memory from 000A0000h to 000FFFFFh.

## 4.4 I/O Address Space

The MCH does not support the existence of any other I/O devices beside itself on the system bus. The MCH generates either Hub Interface_A–C or AGP bus cycles for all processor I/O accesses. The MCH contains two internal registers in the processor I/O space, Configuration Address (CONF_ADDR) register and the Configuration Data (CONF_DATA) register. These locations are used to the implement configuration space access mechanism as described in the *Register Description* chapter.

The processor allows 64K+3 bytes to be addressed within the I/O space. The MCH propagates the processor I/O address without any translation on to the destination bus and therefore provides addressability for 64K+3 byte locations. Note that the upper 3 locations can be accessed only during I/O address wrap-around when system bus A16# address signal is asserted. A16# is asserted on the system bus whenever an I/O access is made to 4 bytes from address 0FFFDh, 0FFFEh, or 0FFFFh. A16# is also asserted when an I/O access is made to 2 bytes from address 0FFFFh.

The I/O accesses (other than ones used for configuration space access) are forwarded normally to either the Hub Interface_A, Hub Interface_B, or Hub Interface_C unless they fall within the AGP I/O address range as defined by the mechanisms explained below. The MCH will not post I/O write cycles to IDE.

The MCH never responds to I/O or configuration cycles initiated on AGP or any of the hub interfaces. Hub interface transactions requiring completion are terminated with "master abort" completion packets on the hub interfaces. Hub interface write transactions not requiring completion are dropped. AGP/PCI I/O reads are never acknowledged by the MCH.

## 4.5 MCH Decode Rules and Cross-Bridge Address Mapping

The address map described above applies globally to accesses arriving on any of the five interfaces (i.e., Host bus, Hub Interface_A, Hub Interface_B, Hub Interface_C, or AGP).

### 4.5.1 Hub Interface_A Decode Rules

The MCH accepts accesses from the Hub Interface_A with the following address ranges:

- All memory read and write accesses to main memory (except SMM space).
- All memory write accesses from the Hub Interface_A-to-AGP memory range defined by MBASE1, MLIMIT1, PMBASE1, and PMLIMIT1.
- All memory read/write accesses to the Graphics Aperture defined by APBASE and APSIZE.
- Memory writes to VGA range on AGP if enabled.

All memory reads from the Hub Interface_A that are targeted > 4 GB memory range are terminated with Master Abort completion, and all memory writes (> 4 GB) from the Hub Interface_A are ignored.

### 4.5.2 Hub Interface_B Decode Rules

The MCH accepts accesses from the Hub Interface_B from the following address ranges:

- All memory read and write accesses to Main DRAM (except SMM space).
- All memory write accesses from the hub interface-to-AGP memory range defined by MBASE2, MLIMIT2, PMBASE2, and PMLIMIT2.
- All memory read/write accesses to the Graphics Aperture defined by APBASE and APSIZE.
- Memory writes to VGA range on AGP if enabled.

Memory accesses from the Hub Interface_B that fall elsewhere within the memory range and I/O cycles will not be accepted. They are terminated with Master Abort completion.

### 4.5.3 Hub Interface_C Decode Rules

The MCH accepts accesses from the Hub Interface_C from the following address ranges:

- All memory read and write accesses to main memory (except SMM space).
- All memory write accesses from the hub interface-to-AGP memory range defined by MBASE3, MLIMIT3, PMBASE3, and PMLIMIT3.
- All memory read/write accesses to the graphics aperture defined by APBASE and APSIZE.
- Memory writes to VGA range on AGP if enabled.

Memory accesses from the Hub Interface_C that fall elsewhere within the memory range and I/O cycles will not be accepted. They are terminated with Master Abort completion.

## 4.5.4 AGP Interface Decode Rules

### Cycles Initiated Using AGP FRAME# Protocol

The MCH does not support any AGP FRAME# access targeting the Hub Interface_A. The MCH will claim AGP-initiated memory read and write transactions decoded to the main memory range or the graphics aperture range. All other memory read and write requests are master-aborted by the AGP initiator as a consequence of MCH not responding to a transaction.

Under certain conditions, the MCH restricts access to the DOS Compatibility ranges governed by the PAM registers by distinguishing access type and destination bus. The MCH does NOT accept AGP FRAME# write transactions to the compatibility ranges if the PAM designates main memory as writeable. If accesses to a range are not write-enabled by the PAM, the MCH does not respond and the cycle results in a master-abort. The MCH accepts AGP FRAME# read transactions to the compatibility ranges if the PAM designates main memory as readable. If accesses to a range are not read-enabled by the PAM, the MCH does not respond and the cycle results in a master-abort.

If an agent on AGP issues an I/O, PCI Configuration or PCI Special Cycle transaction, the MCH does not respond and the cycle results in a master-abort.

### Cycles Initiated Using AGP PIPE# or SB Protocol

All cycles must reference main memory; that is, main memory address range (including PAM) or Graphics Aperture range (also physically mapped within main memory but using a different address range). AGP accesses to SMM space are not allowed. AGP-initiated cycles that target main memory are not snooped on the host bus, even if they fall outside of the AGP aperture range.

If a cycle is outside of the main memory range, it terminates as follows:

- **Reads:** remap to memory address 0h, return data from address 0h, and set the IAAF error bit in ERRSTS register in Device 0
- **Writes:** dropped “on the floor” (i.e., terminated internally without affecting any buffers or main memory)

### AGP Accesses to MCH that Cross Device Boundaries

For AGP FRAME# accesses, when an AGP master gets disconnected, it resumes at the new address which allows the cycle to be routed to or claimed by the new target. Therefore, the target on potential device boundaries should disconnect accesses. The MCH disconnects AGP FRAME# transactions on 4 KB boundaries.

AGP PIPE# and SBA accesses are limited to 256 bytes and must hit main memory. Read accesses crossing a device boundary returns invalid data when the access crosses out of main memory. Write accesses crossing out of main memory are discarded. IAAF Error bit will be set.

# 5 Memory Interface

The MCH directly supports dual channels (interfaces) of Rambus Direct RDRAM devices operating in lock-step using RSL technology. The MCH supports two different operation modes:

- **Single Channel-Pair Mode.** The MCH is configured to directly support Direct RDRAM devices on its dual Rambus interfaces. There is no Intel MRH-R used on the memory subsystem. A maximum of 64 Direct RDRAM devices are supported on the paired channels without external logic.
- **Multiple Channel-Pair Mode.** The MCH is configured to use the Intel MRH-R on the memory subsystem. Each Rambus Channel of the Intel MRH-R on the MCH Direct Rambus Channel A is paired with one Rambus Channel of the Intel MRH-R on the Direct Rambus Channel B. The MCH supports one Intel MRH-R per interface, and each Intel MRH-R can support up to two Rambus Channels. Therefore, up to four Rambus Channels are supported by the MCH.

The interface between the MCH and Direct RDRAM devices is referred to either as a “channel” or as an “expansion channel.” The channel interface consists of 33 signals including clocks (30 signals are RSL and three signals are CMOS). There are two additional RSL signals per channel when the Intel MRH-R is used for channel expansion.

Figure 7 shows the interconnections between the MCH and its dual Rambus Channels configured in single channel-pair mode.

### Figure 7. Single Channel-Pair Mode

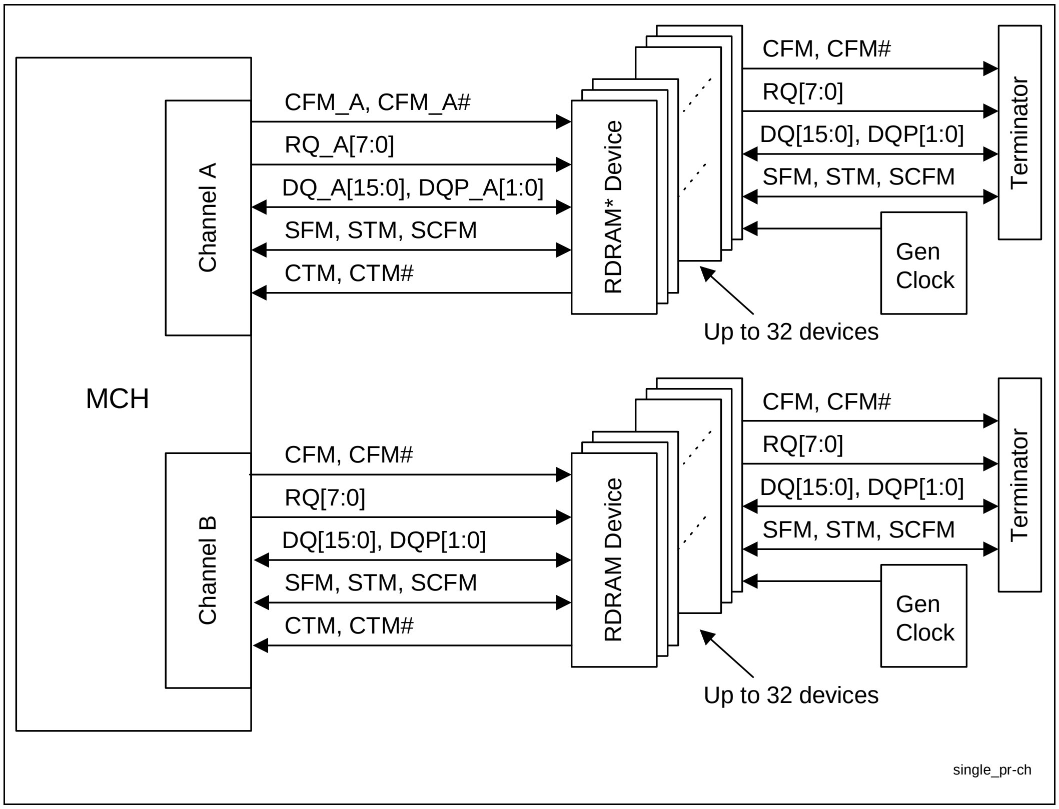
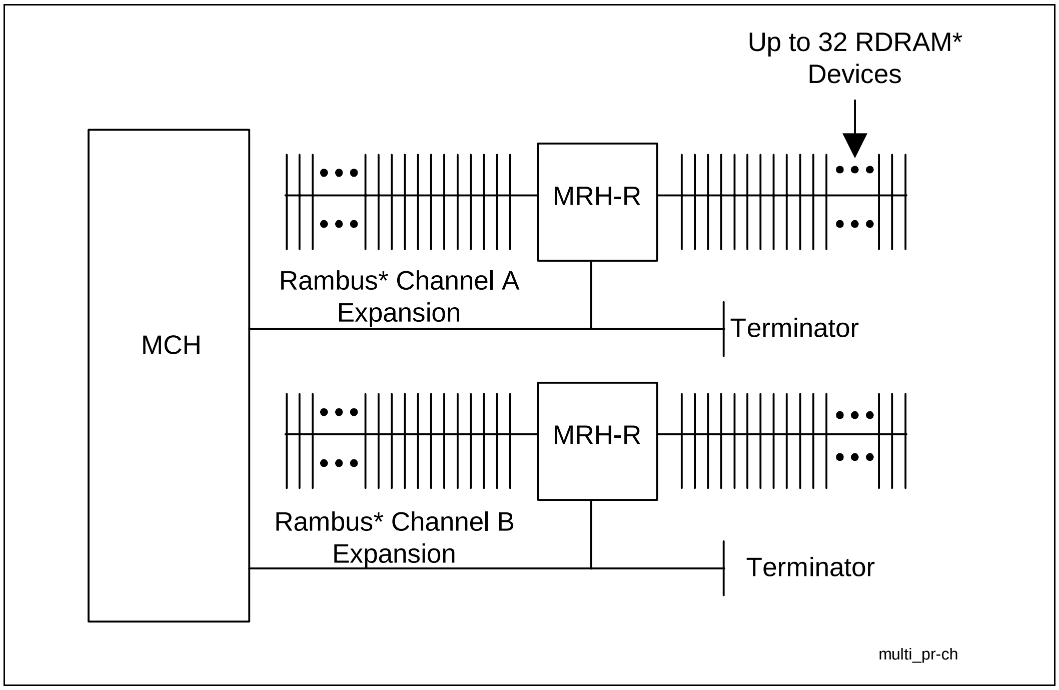


Figure 8 shows the interconnections between MCH and its dual Rambus Channels configured at multiple channel-pair mode.

**Figure 8. Multiple Channel-Pair Mode**



The maximum system memory supported by the MCH depends on the Direct RDRAM device technology (Section 1.4.2, *Memory Interface* lists the maximum memory supported). The row, column, and bank address bits required for the Direct RDRAM device depends on the number of banks and page size of the device (see Section 1.4.2, *Memory Interface* for further information).

A brief overview of the registers that configure the Direct RDRAM device interface is provided below:

- **Group Boundary Address Register (GBA).** GBA registers define the upper and lower addresses for a group of Direct RDRAM device pairs in a channel-pair. Each group requires a separate GBA register. Each group consists of four device-pairs in single-channel mode and eight device-pairs in multiple-channel mode. The MCH contains 16 GBA registers.
- **Group Architecture Register (GAR).** GAR registers specify the architecture features of each group of device pairs in a channel pair. The architecture features specified are bank-type and device-core technology. Each GAR represents a group consisting of four device-pairs in single-channel mode and eight device-pairs in multiple-channel mode. There is a 1:1 correspondence between GBA and GAR registers.
- **Direct RDRAM Device Timing Register (RDTR).** The DTR defines the timing parameters for all devices in all channels. BIOS programs this register with “least common denominator” values after reading the configuration registers of each device in the channels.
- **Direct RDRAM Device Pool Sizing Register (RPMR).** This register provides bits to program the number of RDRAM device-pair in one of three RDRAM power management states.
- **Direct RDRAM Device Initialization Control Register (RICM).** This register provides bits to program the MCH to do initialization activities on Direct RDRAM devices.

## 5.1 Direct RDRAM* Device Organization and Configuration

The MCH supports 16-/18-bit Direct RDRAM device configurations. The MCH supports a maximum of 64 Direct RDRAM devices (32 devices per channel) on its dual Rambus Channels. The Rambus Channel can be populated with a mix of 128-/144-Mbit and 256-/288-Mbit Direct RDRAM devices.

### 5.1.1 Rules for Populating Direct RDRAM* Devices

MCH Rambus Channels can be fully or partially loaded with Direct RDRAM devices; however, they must be populated in either single-device pair or multiple-device pair.

- **Single-Device-pair.** The MCH is configured to directly support Direct RDRAM devices on its dual Rambus Channel. Each Direct RDRAM device of the MCH Rambus Channel A is paired with one Direct RDRAM device of the Rambus Channel B. There is no Intel MRH-R used on the memory subsystem.
- **Multiple Device-pair.** The MCH is configured to use Intel MRH-R on the memory subsystem. Each Direct RDRAM device on Rambus Channel A is paired with one Direct RDRAM device on the Rambus Channel B.

***Note:*** The MCH supports a maximum of two RIMMs per channel.

From the MCH point of view, all device-pairs in the channels are grouped into logical groups. System initialization software partitions the Direct RDRAM devices into groups of four device-pairs in single-channel mode operation and into groups of eight device-pairs in multiple channel mode operation. As a result, there can be a maximum of eight groups per channel-pair in single-channel pair operation and a maximum of four groups per channel pair in multiple channel-pair mode. All device-pairs populated in a group must be of the same architecture. In other words all device-pairs in a group must be the same core technology and have the same number of banks. Following are the rules for populating the groups:

- A group can be partially populated.
- There is no requirement that group members have to be populated in contiguous physical slots.
- There can be a maximum of eight groups in single-channel pair mode or four groups per channel in multiple channel-pair mode. A member that does not belong to any of the groups in the channel will not be recognized.

Table 13 provides the device IDs for members in all groups.

### Table 13. Direct RDRAM* Device Grouping

| Single-Channel Mode | | Multiple Channel Mode | |
|---|---|---|---|
| Device-Pair IDs for Group Members | Group Name | Device Pair IDs for Group Members | Group Name |
| 0, 1, 2, 3 | Group#0 | 0, 1, 2, 3, 4, 5, 6, 7, | Ch#0 Pair, Group#0 |
| 4, 5, 6, 7 | Group#1 | 8, 9, 10, 11, 12, 13, 14, 15 | Ch#0 Pair, Group#1 |
| 8, 9, 10, 11 | Group#2 | 16, 17, 18, 19, 20, 21, 22, 23 | Ch#0 Pair, Group#2 |
| 12, 13, 14, 15 | Group#3 | 24, 25, 26, 27, 28, 28, 30, 31 | Ch#0 Pair, Group#3 |
| 16, 17, 18, 19 | Group#4 | 0, 1, 2, 3, 4, 5, 6, 7, | Ch#1 Pair, Group#0 |
| 20, 21, 22, 23 | Group#5 | 8, 9, 10, 11, 12, 13, 14, 15 | Ch#1 Pair, Group#1 |
| 24, 25, 26, 27 | Group#6 | 16, 17, 18, 19, 20, 21, 22, 23 | Ch#1 Pair, Group#2 |
| 28, 29, 30, 31 | Group#7 | 24, 25, 26, 27, 28, 28, 30, 31 | Ch#1 Pair, Group#3 |
| | | 0, 1, 2, 3, 4, 5, 6, 7, | Ch#2 Pair, Group#0 |
| | | 8, 9, 10, 11, 12, 13, 14, 15 | Ch#2 Pair, Group#1 |
| | | 16, 17, 18, 19, 20, 21, 22, 23 | Ch#2 Pair, Group#2 |
| | | 24, 25, 26, 27, 28, 28, 30, 31 | Ch#2 Pair, Group#3 |
| | | 0, 1, 2, 3, 4, 5, 6, 7, | Ch#3 Pair, Group#0 |
| | | 8, 9, 10, 11, 12, 13, 14, 15 | Ch#3 Pair, Group#1 |
| | | 16, 17, 18, 19, 20, 21, 22, 23 | Ch#3 Pair, Group#2 |
| | | 24, 25, 26, 27, 28, 28, 30, 31 | Ch#3 Pair, Group#3 |

All RSL signals must be terminated at the far end from the MCH.

The default device ID for a Direct RDRAM device after power up is 1Fh.

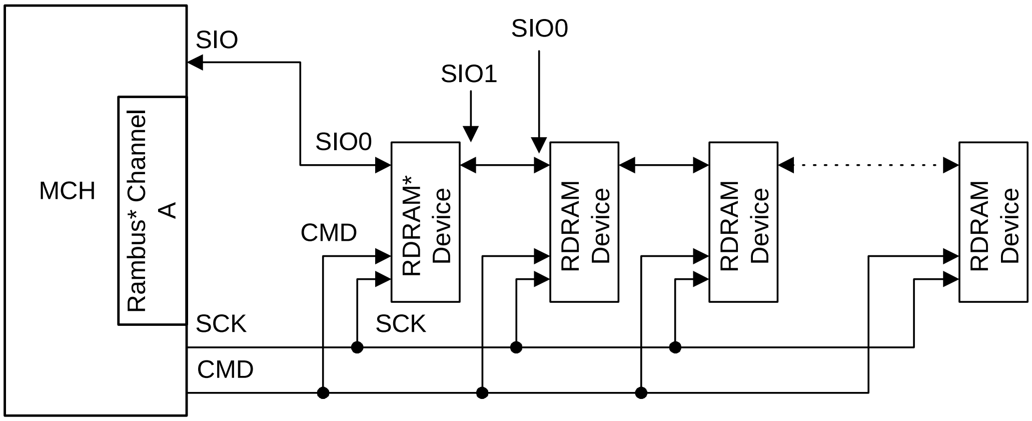
### 5.1.2 Direct RDRAM* Device CMOS Signals

There are three CMOS signal pins per channel on the MCH to support Direct RDRAM device configuration, SIO reset, register accesses, and Nap and Powerdown exits. These signals are SCK, CMD and SIO. These signals are used to perform the following operations:

- SIO pin initialization
- SIO operations (includes register accesses and device reset)
- Device selection for Nap and Powerdown exits

***Note:*** The MCH supports dual Rambus Channels

**Figure 9. Direct RDRAM* Devices Sideband CMOS Signal Configuration on Rambus* Channel A**



Note: MCH supports Dual Rambus Channels.

rdram_cmos

**Table 14. Sideband CMOS Signal Description**

| Signal | Description |
|---|---|
| SCK | **Serial Clock:** This signal serves as the clock for SIO and CMD signals. SCK is a clock source used for reading from and writing to control register.<br>• For SIO operations and pin initialization, SCK ≤ 1 MHz<br>• For power mode operations, SCK ≤ 100 MHz |
| CMD | **Command:** CMD is a control signal used for power mode transitions, SIO pin configuration during initialization, and framing of SIO operations. CMD is active high. CMD is sampled at both edges of SCK. CMD is a level sensitive signal. |
| SIO | **Serial In Out:** This bi-directional signal is daisy chained through all Direct RDRAM* devices (SIO0 to SIO1) in a channel. This pin carries data used for SIO operations, which include register accesses, device reset, and device ID initialization. It is also used for power mode control. SIO is an active low signal and is sampled on the falling edge of SCK. |

### Table 15. CMD Signal Value Decode

| SIO = 0, CMD Sample Value on 4 SCK Edges | | | | Command | SIO = 1, CMD Sample Value on 4 SCK Edges | | | | Command |
|---|---|---|---|---|---|---|---|---|---|
| Cycle 0 | | Cycle 1 | | | Cycle 0 | | Cycle 1 | | |
| 0 | 1 | x | X | Nap Exit | 0 | 1 | x | x | Power-down Exit |
| 1 | 0 | x | X | Reserved | 1 | 0 | x | x | Reserved |
| 0 | 0 | x | X | No-op | 0 | 0 | x | x | No-op |
| 1 | 1 | 1 | 1 | SIO Request Frame | 1 | 1 | 1 | 1 | SIO Request Frame |
| 1 | 1 | 0 | 0 | SIO Reset | 1 | 1 | 0 | 0 | SIO Reset |
| 1 | 1 | 1 | 0 | Reserved | 1 | 1 | 1 | 0 | Reserved |
| 1 | 1 | 0 | 1 | Reserved | 1 | 1 | 0 | 1 | Reserved |

## SIO Pin Initialization

The SIO0 and SIO1 pins on the Direct RDRAM devices are bi-directional and their direction needs to be initialized. The “SIO Reset” initializes the SIO0 and SIO1 pins on all Direct RDRAM devices as daisy chain configuration and is performed with the SCK and CMD. Once the SIO daisy chain is fully configured, SIO operations can occur.

***Note:*** “SIO Reset” does NOT reset the entire device. For a complete description of the operation and associated timing diagrams, refer to the *Direct RDRAM Component* data sheet from Rambus.

## SIO Operations

SIO operations are also known as Direct RDRAM device initialization operations. These operations include Direct RDRAM device register accesses and device reset, and is performed using the CMOS pins: SCK, CMD, SIO0, and SIO1. For a complete description of operation and associated timing diagram, refer to *Direct RDRAM Component* data sheet from Rambus.

## Nap and Powerdown Exits

The Nap and Powerdown exits are performed using CMD, SIO and SCK signals. For complete description and timing diagrams associated with Nap and Powerdown exits, refer to *Direct RDRAM Component* data sheet from Rambus*.

### 5.1.3 Direct RDRAM* Device Core Refresh

All rows in a Direct RDRAM device must be refreshed within 32 ms. The refresh rate depends on the device size and page size of a device.

The MCH supports two core refresh mechanisms: Active refresh and self refresh

- **Active Refresh:** Refresh and precharge after Refresh commands are issued from the primary control packet. These commands provide refresh support in standby/active modes.
- **Self Refresh:** Internal time-base and row/bank address counters in the core allow for a self refresh in powerdown modes without controller support.

#### Direct RDRAM* Device Current Calibration

All Direct RDRAM devices must be current calibrated once every 100 ms. There are RSL commands to perform this function. The MCH schedules periodic current calibration activity such that every device in the channel is current calibrated at least once every 100 ms.

## 5.2 Direct RDRAM* Device Command Encoding

The operations on a Rambus Channel are performed using control packets. There are two types of command packets: row (ROWA/ROWR) packet and column (COLC/COLM/COLX) packet. Each command packet requires four Direct RDRAM device clock durations and packet data is transferred on both (leading and falling) edges of the clock. The row packet contains 24 bits and the column packet contains 40 bits.

### 5.2.1 Row Packet (ROWA/ROWR)

The row packet is defined using three RSL signals RQ[7:5]/ROW[2:0]. It will generally be the first control packet issued to a device. Major characteristics of row packet are:

- The only way to activate (sense) a row within a bank
- Independent of Direct RDRAM device Active/Standby state
- A non-broadcast row package causes an addressed Direct RDRAM device to move to Active state

The packet definition of row packet is given below.

### Table 16. ROWA Packet for Activating (sensing) a Row (i.e., AV = 1)

| Row | Cycle 0 | | Cycle 1 | | Cycle 2 | | Cycle 3 | |
|---|---|---|---|---|---|---|---|---|
| ROW 2 | DR4T | DR[2] | BR[0] | BR[3] | R[10] | R[8] | R[5] | R[2] |
| ROW 1 | DR4F | D[R1] | BR[1] | BR[4] | R[9] | R[7] | R[4] | R[1] |
| ROW 0 | DR[3] | DR[0] | BR[2] | REV | AV = 1 | R[6] | R[3] | R[0] |

### Table 17. ROWR Packet for Other Operations (i.e., AV = 0)

| Row | Cycle 0 | | Cycle 1 | | Cycle 2 | | Cycle 3 | |
|---|---|---|---|---|---|---|---|---|
| ROW 2 | DR4T | DR[2] | BR[0] | BR[3] | ROP[10] | ROP[8] | ROP[5] | ROP[2] |
| ROW 1 | DR4F | DR[1] | BR[1] | BR[4] | ROP[9] | ROP[7] | ROP[4] | ROP[1] |
| ROW 0 | DR[3] | DR[0] | BR[2] | REV | AV = 0 | ROP[6] | ROP[3] | ROP[0] |

| DR4T | DR4F | Device ID |
|---|---|---|
| 0 | 0 | No row packet |
| 0 | 1 | DR[3:0], DR[4] = 0 |
| 1 | 0 | DR[3:0], DR[4] = 1 |
| 1 | 1 | Broadcast |

**NOTES:**

| | | |
|---|---|---|
| 1. | DR[4]–DR[0] | Device Address |
| 2. | BR[5]–BR[0] | Bank Address |
| 3. | R[10]–R[0] | Row Address |
| 4. | AV | Select between ROWA and ROWR, Active Row |
| 5. | ROP[10]–ROP[0] | Opcode for Primary Control Packet |
| 6. | REV | Reserved |

**Table 18. Row Packet Encodings**

| AV | Opcode bits | | | | | | | | | Operation Description |
|---|---|---|---|---|---|---|---|---|---|---|
| | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2: 0 | |
| 1 | x | x | x | X | x | x | x | x | xxx | Activate Row |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 000 | Precharge |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 1 | 000 | Precharge and Relax |
| 0 | 0 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 000 | Refresh |
| 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 000 | Precharge Postrefresh |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 000 | Nap |
| 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 0 | 000 | Conditional Nap |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 000 | Power Down |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 000 | Relax |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 010 | Temp Calibration Enable |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 001 | Temp Calibration |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 000 | No-op |

**NOTES:**

1. x = Controller Drives 0 or 1
2. 0 = Controller Drives 0
3. 1 = Controller Drives 1

## 5.2.2 Column Packet (COLC/COLX)

The column packet is defined using five of the RSL signals RQ[4:0]/COL[4:0]. Major characteristics of column are:

- The only way to dispatch column operation for read or write
- Requires the target Direct RDRAM device to be in Active state

***Note:*** When a Direct RDRAM device is in the Active state, it can receive both row and column packets. When a Direct RDRAM device is in the Standby state, it can only receive a row packet. Thus, before sending a column packet, make sure the addressed Direct RDRAM device is in the Active state.

The packet definition of column packet is given below.

## Table 19. COLC Packet

| Column | Cycle 0 | | Cycle 1 | | Cycle 2 | | Cycle 3 | |
|---|---|---|---|---|---|---|---|---|
| COL4 | DC[4] | S = 1 | | | | | C[6] | C[4] |
| COL3 | DC[3] | | | | | | C[5] | C[3] |
| COL2 | DC[2] | COP[1] | | | | REV | BC[2] | C[2] |
| COL1 | DC[1] | COP[0] | | | | BC[4] | BC[1] | C[1] |
| COL0 | DC[0] | COP[2] | | | COP[3] | BC[3] | BC[0] | C[0] |

**NOTES:**

1. DC[4:0] Device ID for Column Operation
2. S Start Bit; for framing
3. M Mask Bit; asserted indicates mask format for packet
4. COP[3:0] Column Operation Code
5. C[6:0] Address for Column Operation
6. BC[4:0] Bank Address for Column Operation
7. REV Reserved

## Table 20. COLC Packet Field Encodings

| S | COP[3] | COP[2] | COP[1] | COP[0] | Command Operation |
|---|---|---|---|---|---|
| 0 | x | x | x | x | No operation |
| 1 | x | 0 | 0 | 0 | NOCOP. Retire write buffer of this device |
| 1 | x | 0 | 0 | 1 | Write |
| 1 | x | 0 | 1 | 1 | Read |

**NOTE:** All other combination are reserved

## COLX Packet (M = 0)

| Column | Cycle 0 | | Cycle 1 | | Cycle 2 | | Cycle 3 | |
|---|---|---|---|---|---|---|---|---|
| COL4 | | | DX[4] | XOP[4] | REV | BX[1] | | |
| COL3 | | M = 0 | DX[3] | XOP[3] | BX[4] | BX[0] | | |
| COL2 | | | DX[2] | XOP[2] | BX[3] | | | |
| COL1 | | | DX[1] | XOP[1] | BX[2] | | | |
| COL0 | | | DX[0] | XOP[0] | | | | |

**NOTES:**

1. DX[4:0] Device ID for Extra Operation
2. BX[4:0] Bank Address for Extra Operation
3. MA[7:0] Byte Mask (low order)
4. MB[7:0] Byte Mask (high order)
5. XOP[4:0] Opcode for Extra Operation
6. REV Reserved

**Table 21. COLM Packet and COLX Packet Field Encodings**

| M | XOP Bits | | | | | Operation Description |
|---|---|---|---|---|---|---|
| | 4 | 3 | 2 | 1 | 0 | |
| 1 | x | x | x | x | x | Non existent Xop |
| 0 | 0 | 0 | 0 | 0 | 0 | NoXop |
| 0 | 1 | 0 | 0 | 0 | 0 | Reserved |
| 0 | 0 | 1 | 0 | 0 | 0 | Calibrate Current |
| 0 | 0 | 1 | 1 | 0 | 0 | Calibrate Current and Sample |
| 0 | 0 | 0 | 0 | 0 | 1 | Reserved |

**NOTES:**

1. x = Controller Drives 0 or 1
2. 0 = Controller Drives 0
3. 1 = Controller Drives 1

### 5.2.3 Data Packet

**Table 22. Data Packet**

| Data Signals | Cycle 0 | | Cycle 1 | | Cycle 2 | | Cycle 3 | |
|---|---|---|---|---|---|---|---|---|
| DQA[8:0] | DA0[8:0] | DA1[8:0] | DA2[8:0] | DA3[8:0] | DA4[8:0] | DA5[8:0] | DA6[8:0] | DA7[8:0] |
| DQB[8:0] | DB0[8:0] | DB1[8:0] | DB2[8:0] | DB3[8:0] | DB4[8:0] | DB5[8:0] | DB6[8:0] | DB7[8:0] |

## 5.3 Direct RDRAM* Device Register Programming

Software can read and write Direct RDRAM device registers by programming the Direct RDRAM Device Initialization Control Management register (RICM) in the MCH. The register data returned by the device is available in the Device Register Data register (DRD).

## 5.4 Direct RDRAM* Device Operating States

The Direct RDRAM devices support different operating and idle states to minimize the power consumption and thermal overload. Table 23 provides an overview of the different operating/power states supported by Direct RDRAM devices.

### Table 23. DRAM Operating States

| Direct RDRAM* Device State | Functionality | Refresh Scheme | Direct RDRAM Device Clock State |
|---|---|---|---|
| **Inactive States** | | | |
| Powerdown | No operation allowed except refresh. Direct RDRAM* device awaits CMOS signals to exit Powerdown state | Self Refresh | stopped |
| Nap | No operation allowed except refresh. Direct RDRAM device awaits Nap exit command to exit Nap | Active Refresh | stopped |
| **Active States** | | | |
| Standby | Device Ready to receive row packet with fast clock | Active Refresh | full speed |
| Active | Device ready to receive any control packet | Active Refresh | full speed |
| Active-Read | Device ready to receive any control packet. Transmitting data on channel | Active Refresh | full speed |
| Active-Write | Device ready to receive any control packet. Receiving data from channel | Active Refresh | full speed |

## Active-Read/Write State

A Direct RDRAM device is in Active-Read/Write state when it is transferring data. This state lasts as long as data transfer is occurring. Once the data transfer is complete, the Direct RDRAM device transitions into Active or Standby state based on the column command last executed.

## Active State

A Direct RDRAM device enters into the Active state immediately after the data transfer from/to that device is complete and the last COLC command that caused the data transfer does not have its RC bit set to 1. When a device is in Active state, it can accept both row and column packets.

## Standby State

A Direct RDRAM device enters into Standby state either from Active-Read/Write or Active state. Transition from Active-Read/Write to Standby happens if the last column executed has its RC bit set to 1. Transition from Active-to-Standby happens if COLC or row specifies an operation with the Relax command. When a device is in standby mode it can accept only row packets. Once a device receives any row packet, it transitions into Active state and then only it can accept a column packet.

## Nap State

A Direct RDRAM device enters into Nap state when it receives a row packet that specified an operation with Nap. No operation except refresh is allowed during Nap state.

## Powerdown State

A Direct RDRAM device enters into Powerdown state when it receives a row packet that specified an operation with powerdown. No operations except self-refresh is allowed during Powerdown state.

## 5.5 Direct RDRAM* Device Operating Pools

To minimize the operating power, the Direct RDRAM devices are grouped into three operating pools called Pool “A”, Pool “B”, and Pool “C”.

### Pool “A”, Pool “B”, and Pool “C” Operation

In the pool mode, three queues are used inside the MCH. The “A” pool contains references to device pairs that are currently in the active mode; the “B” pool contains references to device pairs that are in the standby mode. All devices that are **not** found in pool “A” or “B” are said to be in Pool “C” and can be configured for either napping or standby. The “A” pool may hold between 1 and 8 device pairs, while the “B” pool may be configured to contain between 1 and 16 device pairs.

## 5.6 Direct RDRAM* Device Power Management

Systems based on the Intel 82860 MCH support ACPI-based power management. The MCH puts all Direct RDRAM devices into Powerdown (PD) state during S3 power management states. To enter the Powerdown state all Direct RDRAM devices in the channel must be in active or standby state. The MCH then sends a broadcast Powerdown command to that channel.

During the Powerdown state, Direct RDRAM devices are put into self-refresh mode so that external (active) refreshes are not required. During the Powerdown state, the clocks to Direct RDRAM devices are shut off. Exiting the power-down and Nap states are done through CMOS signals.

Table 24 shows the actions taken by the MCH during different processor and system power states.

### Table 24. Direct RDRAM* Device Power Management States

| Processor State | System State | State of Direct RDRAM* Devices in Pool “A” | State of Direct RDRAM Devices in Pool “B” | State of Direct RDRAM Devices in Pool “C” | Refresh Scheme | Direct RDRAM Device Clock State |
|---|---|---|---|---|---|---|
| C0, C1, C2 (processor in working state) | S0 | Active-Read/Write, Active | Standby | Nap or Standby | Active | Running |
| (processor in inactive state) | S1, S3( STR) | No devices in Pool “A” | No devices in Pool “B” | Power-down | Self | Stopped |

## 5.7 Data Integrity

The MCH supports an error correcting code (or error checking and correcting) on the main memory interface. The MCH can optionally be configured to generate the ECC code for writes to memory and check the code for reads from memory. The MCH generates an 8-bit code word for each 64-bit QWord of memory. Since the code word covers a full QWord, writes of less than a QWord require a read-merge-write operation. Consider a DWord write to memory. In this case, when in ECC mode, the MCH will read the QWord where the addressed DWord will be written, merge in the new DWord, generate a code covering the new QWord and finally write the entire QWord and code back to memory. Any correctable (single) errors detected during the initial QWord read are corrected before merging the new DWord.

Single-bit and multiple-bit errors set separate flags in the ERRSTS register. Single-bit errors and multiple-bit errors can be independently enabled to generate hub interface SERR, SMI, or SCI special cycles to the ICH2. The address and syndrome of the first single-bit error are latched in the EAP and DERRCTL Registers. Subsequent single-bit errors will not overwrite the EAP and DERRCTL Registers unless the single-bit error status bit is cleared. A multiple-bit error will overwrite the EAP and DERRCTL Registers. Subsequent multiple-bit errors will not overwrite the EAP and DERRCTL Registers unless the multiple-bit error status bit is cleared.

***Note:*** During a write to memory over hub interface_B/C, if a parity error occurs, the byte enables will be turned off. Parity Errors will be logged and can be recovered by system or device handlers. Data with parity errors will not be merged into memory. Some PCI cards do not implement Parity correctly. If a false parity error occurs, that transaction will not be placed in memory.

***Note:*** When an Intel 860 chipset platform is configured for ECC support, if a multi-bit uncorrectable memory error is detected during a memory read by a system device, an SERR, SCI, or SMI will be generated. This typically results in an NMI; however, bad data can still reach the intended target before the NMI can be generated or before the NMI interrupt handler can service the problem. This can result in bad data being returned to the target and may be permanently stored, resulting in system data corruption. This chipset was not designed to ensure that targets are protected from this corrupted data in these situations.

## 5.8 Direct RDRAM* Device Array Thermal Management

The Direct RDRAM device thermal and power management of the MCH has been optimized for workstation system designs. It is assumed that proper system design will always provide and ensure adequate cooling in a system based on the Intel 860 chipset. The failsafe mechanism that protects the devices in the event of a catastrophic failure requires an external thermal sensor. When the thermal sensor is activated, the MCH immediately exits the “all devices on” mode and reverts to the pool mode that has been programmed by system software.

In a system based on the Intel 82860 MCH, Direct RDRAM devices operate in one of three modes: active, standby, or nap. The number of devices allowed in each state at any given time is dictated by the heat dissipation budget specified by the system designer. At any point, between 1 and 8 device-pairs may be in the “A” pool and are configured to operate in the active mode. In addition, between
1 and 16 device-pairs may be in the “B” pool and are configured to operate in the standby mode. The rest of the device-pairs are in the “C” pool and may be configured to operate in either nap mode or standby mode. Regardless of how many devices are configured into the “A” and “B” pools or whether the “C” pool devices are in napping or standby mode, the system designer is responsible for providing adequate cooling for the number of Direct RDRAM devices in the system.

After BIOS loads the system’s “target” values into the DPS register and initializes the pools, it should load a “safer” set of values into the DPS register **without** setting the POOLINIT field. The POOLINIT bit instructs the MCH to transition to the new pool sizes. There are two other conditions that cause the MCH to resize and initialize the pools:

- The transition of the OVERT# pin from electrical 1 to electrical 0
- The detection of an over-temperature condition on any Direct RDRAM device

The OVERT# method is intended to allow system designers to use external thermal sensors to monitor the system temperature and assert OVERT# when the system temperature exceeds system specifications. When the MCH detects a falling edge on the OVERT# signal, it reinitializes and resizes the pools with the values that are in the DPS register. Also, the Direct RDRAM devices report over-temperature conditions back to the MCH via a special bit asserted during their current calibration operations. When the MCH detects an over-temperature condition in any of the memory devices, the Direct RDRAM device pools are reinitialized with “safer” values. Finally, the MCH may be configured to send an SERR, SCI, or SMI hub interface message to ICH2. Software may take action to cool the system or to log the condition

# 6 Electrical Characteristics

This chapter provides the absolute maximum ratings, thermal characteristics and DC Characteristics for the Intel 82860 MCH.

AGTL+ (Assisted Gunning Transceiver Logic) signals are open-drain and require termination to a supply that provides the high signal level. Termination resistors are provided on the MCH and are terminated to VTT. This eliminates the need to terminate the bus on the system motherboard. For reliable operation, always connect unused inputs or bi-directional signals to an appropriate signal level. Unused AGTL+ inputs should be left as no connects, as AGTL+ termination is provided on the chip. Unused active high inputs should be connected through a resistor to ground (VSS). Unused outputs can be left unconnected.

The Direct RDRAM device interface introduces a new type of interface called RSL (Rambus Signaling Level) signaling. RSL signals are open-drain drivers and must be terminated to 1.8 V via a 28 Ω termination resistor.

## 6.1 Absolute Maximum Ratings

Table 25 lists the MCH’s maximum environmental stress ratings. Functional operation at the absolute maximum and minimum is neither implied nor guaranteed.

***Warning:*** 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 25. Absolute Maximum Ratings**

| Symbol | Parameter | Min | Max | Unit | Notes |
|---|---|---|---|---|---|
| Tdie | Die Temperature under Bias | 0 | 110 | °C | 1 |
| Tstorage | Storage Temperature | 0 | 105 | °C | |
| VCC | 1.8 V Supply Voltage with respect to VSS | -0.3 | 2.5 | V | |
| VTT | AGTL+ buffer DC input voltage with respect to VSS | 1.4 | 1.7 | V | |
| VDDQ | AGP bus DC input voltage with respect to VSS | 1.4 | 1.8 | V | |

**NOTES:**

1. Based on a No Heatsink condition.

## 6.2 Thermal Characteristics

The MCH is designed for operation at die temperatures between 0 °C and 110 °C. The thermal resistance of the package is provided in Table 26.

### Table 26. Intel® 860 Chipset Package Thermal Resistance

| Parameter | Air Flow (°C/Watt) | |
|---|---|---|
| | No Air Flow | 1 m/s |
| $\text{Psi}_{jt}$ | 0.0 | 0.7 |
| $\text{Theta}_{ja}$ | 20.0 | 16.0 |

**NOTE:** Typical value measured in accordance with EIA/JESD 51-2 testing standard.

## 6.3 Power Characteristics

### Table 27. DC Characteristics Functional Operating Range (VCC1_8 = 1.8V ±5%; Tdie = 110 °C)

| Symbol | Parameter | Min | Typ | Max | Unit | Notes |
|---|---|---|---|---|---|---|
| $P_{Colusa}$ | Thermal Power Dissipation for Intel® 860 chipset | | 7.2 | 9.5 | W | 1 |
| $I_{VTT}$ | Intel 860 chipset VTT supply Current | | | 2.2 | A | |
| $I_{DDQ}$ | Power supply current for AGP interface | | | 370 | mA | |
| $I_{CC}$ | Power supply current for Intel 860 chipset | | | 4.3 | A | |

**NOTES:**

1. This specification is the Thermal Design Power and it is the estimated maximum possible expected power generated in a component by a realistic application. It is based on extrapolations in both hardware and software technology over the life of the component. It does not represent the expected power generated by a power virus. Studies by Intel indicate that no application will cause thermally significant power dissipation exceeding this specification, although it is possible to concoct higher power synthetic workloads that write but never read. Under realistic read/write conditions, this higher power workload can only be transient, and is accounted in the Icc (Max) specification. For more information, refer to the *Intel® 860 Chipset Thermal Considerations Application Note (AP-721).*

## 6.4 I/O Interface Signal Groupings

The signal description includes 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.

**AGP** AGP interface signals. These signals are compatible with AGP 2.0 1.V Signaling Environment DC and AC Specifications. The buffers are not 3.3 V tolerant.

**CMOS** 1.8V CMOS buffers.

**RSL** Rambus Signaling Level interface signal. Refer to the *RDRAM* Direct Specification* for complete details.

**RCMOS** RCMOS buffers are 1.8 V CMOS buffers used for the CMOS signals on the Direct RDRAM device interface.

### Table 28. Signal Groups

| Signal Group | Signal Type | Signals | Notes |
|---|---|---|---|
| (a) | AGTL+ I/O | ADS#, AP[1:0]#, BNR#, BRO#,DBSY#, DP[3:0]#, DBI[3:0]#, DRDY#, HA[35:3]#, HADSTB[1:0] #, HD[63:0]#,HDSTBP[3:0]#, HDSTBN[3:0]#, HIT#, HITM#, HREQ[4:0]# | |
| (b) | AGTL+ Output | BPRI#, CPURST#, DEFER#, HTRDY#, RS[2:0]#, RSP# | |
| (c) | AGTL+ Input | HLOCK#, BERR# | |
| (d) | Hub Interface's CMOS I/O | HL_A[11:0], HLA_STB, HLA_STB#, HL_B[19:0], HLB_STB[1:0], HLB_STB[1:0]#, HL_C[19:0], HLC_STB[1:0], HLC_STB[1:0]# | |
| (e) | CMOS Output | CHx_HCLK[A,B], CHx_RCLK[A,B] | |
| (f) | CMOS Input | TESTIN#, OVERT#, BUSPARK, HLA_ENH# | |
| (g) | Miscellaneous CMOS Input | RSTIN#(3.3 V) | |
| (h) | CMOS Clock Input | BCLK[1:0], 66IN | 1 |
| (i) | RSL I/O | DQA_A[8:0], DQB_A[8:0], DQA_B[8:0], DQB_B[8:0] | |
| (j) | RSL Output | RQ_A[7:5], RQ_A[4:0], CFM_A, CFM_A#, EXP_A[1:0], RQ_B[7:5], RQ_B[4:0], CFM_B, CFM_B#, EXP_B[1:0] | 2 |
| (k) | RSL Input | CTM_A, CTM_A#, CTM_B, CTM_B# | 2 |
| (l) | Rambus* CMOS I/O | SIO_A, SIO_B | |
| (m) | Rambus* CMOS Output | CMD_A, SCK_A, CMD_B, SCK_B | |
| (n) | AGP Input | PIPE#, SBA[7:0], RBF#, WBF#, SB_STB, SB_STB#, G_REQ#, G_SERR# | |
| (o) | AGP Output | ST[2:0], G_GNT# | |

| Signal Group | Signal Type | Signals | Notes |
|---|---|---|---|
| (p) | AGP I/O | AD_STB0, AD_STB0#, AD_STB1, AD_STB1#, G_FRAME#, G_IRDY#, G_TRDY#, G_STOP#, G_DEVSEL#, G_AD[31:0], G_C/BE[3:0]#, G_PAR | |
| (q) | RSL Reference | CHA_REF[1:0], CHB_REF[1:0] | |
| (r) | AGTL+ Reference | HAVREF[1:0], HDVREF[3:0], CCVREF | |
| (s) | Hub interface's and AGP Reference | GREF_0, GREF_1, HLREF_A, HLREF_B, HLREF_C | |
| (t) | Host Compensation Reference Voltage | HSWNG[1:0] | |
| (u) | AGP / Hub Interface Compensation Reference Voltage | HLSWNG_B, HLSWNG_C,G_SWNG | |
| (v) | AGP I/O Voltage | VDDQ | |
| (w) | AGTL+ Termination Voltage | VTT | |
| (x) | 1.8V | VCC1_8 | |

**NOTES:**

1. 66IN is a 3.3 V signal coming from the system clock generator. A voltage translation occurs internally before the clocks are utilized in the 1.8 V MCH.
2. CTM_A, CTM_A#, CTM_B, CTM_B#, CFM_A, CFM_A#, CFM_B, CFM_B# Rambus* Channel differential clocks are also operating at a different DC Value. For further details, refer to the DRCG data sheet at www.rambus.com

## 6.5 DC Characteristics

### Table 29. DC Characteristics at VCC1_8 = 1.8V ±5%

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|---|---|---|---|---|---|---|---|
| **I/O Buffer Supply Voltage** | | | | | | | |
| VCC1_8 | (x) | CMOS I/O Supply Voltage | 1.71 | 1.8 | 1.89 | V | |
| VDDQ | (v) | AGP I/O Supply Voltage | 1.425 | 1.5 | 1.575 | V | |
| VTT | (w) | Host AGTL+ Termination Voltage | 1.44 | 1.6 | 1.7 | V | |
| **Reference Voltage** | | | | | | | |
| CCVREF | (r) | Host Common Clock Reference Voltage | 0.64 x VTT | 2/3 x VTT | 0.70 x VTT | V | |
| CHx_REF | (q) | Rambus* Channel RSL Reference Voltage | 1.32 | 1.40 | 1.48 | V | |
| GREF/ HLREF | (s) | Hub Interface Reference Voltage When Configured for Enhanced Buffer Mode | 0.64 x VCC1_8 | 2/3 x VCC1_8 | 0.70 x VCC1_8 | V | |
| | | Hub Interface Reference Voltage when Configured for Standard Buffer Mode | 0.48 x VCC1_8 | 1/2 x VCC1_8 | 0.52 x VCC1_8 | V | |
| | | AGP Reference Voltage | 0.48 x VDDQ | 1/2 x VDDQ | 0.52 x VDDQ | V | |
| HxVREF | (r) | Host Address and Data Reference Voltage | 0.64 x VTT | 2/3 x VTT | 0.70 x VTT | V | |
| HSWING | (t) | Host Compensation Reference Voltage | 0.32 x VTT | 1/3 x VTT | 0.35 x VTT | V | |
| | (u) | Hub Interface Compensation Reference Voltage | 0.32 x VCC1_8 | 1/3 x VCC1_8 | 0.35 x VCC1_8 | V | |
| | | AGP Compensation Reference Voltage | 0.48 x VDDQ | 1/2 x VDDQ | 0.52 x VDDQ | V | |

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|---|---|---|---|---|---|---|---|
| **Host Interface** | | | | | | | |
| $V_{IL\_H}$ | (a), (c) | Host AGTL+ Input Low Voltage | — | — | (2/3 x VTT) - 0.1 | V | |
| $V_{IH\_H}$ | (a), (c) | Host AGTL+ Input High Voltage | (2/3 x VTT) + 0.1 | — | — | V | |
| $V_{OL\_H}$ | (a), (b) | Host AGTL+ Output Low Voltage | — | — | (1/3 x VTT) + 0.1 | V | VTT =1.7 V |
| $V_{OH\_H}$ | (a), (b) | Host AGTL+ Output High Voltage | VTT - 0.1 | — | — | V | |
| $I_{OL\_H}$ | (a), (b) | Host AGTL+ Output Low Leakage | 19 | 26.65 | 34 | mA | $V_{OL}$ max |
| $I_{L\_H}$ | (a), (c) | Host AGTL+ Input Leakage Current | — | — | 10 | µA | $V_{OL}$ <Vpad < VTT |
| $C_{PAD}$ | (a), (c) | Host AGTL+ Input Capacitance | — | — | 3 | pF | |
| **Direct RDRAM* Device Interface** | | | | | | | |
| $V_{IL\_R}$ | (i) | RSL Input Low Voltage | CHx_REF - 0.5 | — | CHx_REF - 0.175 | V | |
| $V_{IH\_R}$ | (i) | RSL Input High Voltage | CHx_REF + 0.175 | — | CHx_REF + 0.5 | V | |
| $V_{IL\_RC}$ | (l) | Direct RDRAM Device CMOS Input Low Voltage | -0.3 | — | (VCC1_8 / 2) - 0.25 | V | |
| $V_{IH\_RC}$ | (l) | Direct RDRAM Device CMOS Input High | (VCC1_8 / 2) + 0.25 | — | VCC1_8 + 0.3 | V | |
| $V_{OL\_RC}$ | (l), (m) | Direct RDRAM Device CMOS Output Low Voltage | — | — | (VCC1_8 / 2) - 0.65 | V | terminate using a 91 Ω to VCC1_8 and 39 Ω to VSS |
| $V_{OH\_RC}$ | (l), (m) | Direct RDRAM Device CMOS Output High Voltage | (VCC1.0 / 2) + 0.45 | — | — | V | terminate using a 91 Ω to VCC1_8 and 39 Ω to VSS |
| $I_{OL\_R}$ | (i), (j) | RSL Output Low Current | 30 | — | 90 | mA | |
| $I_{OH\_R}$ | (i), (j) | RSL Output High Current | — | — | ±10 | mA | |
| $I_{OL\_RC}$ | (l), (m) | Direct RDRAM Device CMOS Output Low Current | — | — | 14.9 | MA | $V_{OL}$= (VCC1_8 / 2) - 0.6 and VCC1_8(min) |

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|---|---|---|---|---|---|---|---|
| $I_{OH\_RC}$ | (l), (m) | Direct RDRAM Device CMOS Output High Current | 35.5 | — | — | mA | $V_{OH}$ = (VCC 1.8/ 2) + 0.4 and VCC 1.8 (max) |
| $I_{L\_RC}$ | (l) | Direct RDRAM Device CMOS Input Leakage Current | — | — | ±10 | µA | 0 < Vin < VCC1_8 |
| $C_{IN\_R}$ | (i) | RSL Input Capacitance | 4 | — | 6 | pF | $F_C$=1 MHz |
| $C_{IN\_RC}$ | (l) | Direct RDRAM Device CMOS Input (SCK, CMD) Capacitance | 7 | — | 9 | pF | $F_C$=1 MHz |
| | (l) | Direct RDRAM Device CMOS Input (SIO) Capacitance | 4 | — | 6 | pF | $F_C$=1 MHz |
| **AGP Interface** | | | | | | | |
| $V_{IL\_A}$ | (n), (p) | AGP Input Low Voltage | -0.5 | — | 0.4 x VDDQ | V | |
| $V_{IH\_A}$ | (n), (p) | AGP Input High Voltage | 0.6 x VDDQ | — | VDDQ + 0.5 | V | |
| $V_{OL\_A}$ | (o), (p) | AGP Output Low Voltage | — | — | 0.15 x VDDQ | V | |
| $V_{OH\_A}$ | (o), (p) | AGP Output High Voltage | 0.85 x VDDQ | — | — | V | |
| $I_{OL\_A}$ | (o), (p) | AGP Output Low Current | — | — | 1 | mA | $V_{OL}$=0.15 VDDQ |
| $I_{OH\_A}$ | (o), (p) | AGP Output High Current | -0.2 | — | — | mA | $V_{OH}$=0.85 VDDQ |
| $I_{L\_A}$ | (n), (p) | AGP Input Leakage Current | — | — | ±10 | µA | 0<Vin< VDDQ |
| $C_{IN\_A}$ | (n), (p) | AGP Input Capacitance | — | — | 8 | pF | $F_C$=1 MHz |
| **Hub Interface with Normal Buffer Mode** | | | | | | | |
| $V_{IL\_HL}$ | (d) | Hub Interface Input Low Voltage | — | — | HLREF - 0.15 | V | |
| $V_{IH\_HL}$ | (d) | Hub Interface Input High Voltage | HLREF + 0.15 | — | — | V | |
| $V_{OL\_HL}$ | (d) | Hub Interface Output Low Voltage | — | — | 0.1 x VCC1_8 | V | $I_{OL}$= 1 mA |
| $V_{OH\_HL}$ | (d) | Hub Interface Output High Voltage | 0.9 x VCC1_8 | — | — | V | $I_{OH}$ = 1 mA |
| $I_{OL\_HL}$ | (d) | Hub Interface Output Low Current | — | — | 1 | mA | $V_{OL}$=0.1 x VCC1_8 |

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|---|---|---|---|---|---|---|---|
| $I_{OH\_HL}$ | (d) | Hub Interface Output High Current | -1 | — | — | mA | $V_{OH}$=0.9 x VCC1_8 |
| $I_{L\_HL}$ | (d) | Hub Interface Input Leakage Current | — | — | ±10 | µA | 0<Vin< VCC1_8 |
| $C_{IN\_HL}$ | (d) | Hub Interface Capacitance | — | — | 8 | pF | $F_C$=1 MHz |
| **Hub Interface with Enhanced Buffer Mode Configured for 60 Ω** | | | | | | | |
| $V_{IL\_HL}$ | (d) | Hub Interface Input Low Voltage | — | — | HLREF - 0.15 | V | |
| $V_{IH\_HL}$ | (d) | Hub Interface Input High Voltage | HLREF + 0.15 | — | — | V | |
| $V_{OL\_HLPD}$ | (d) | Hub Interface Output Low Voltage | 0.33 x VCC1_8 -0.06 | — | 0.33 x VCC1_8 +0.06 | V | 60 Ω load to VCC1_8, buffer pulling down |
| $V_{OL\_HLPU}$ | (d) | Hub Interface Output Low Voltage | 0.33 x VCC1_8 -0.06 | — | 0.33 x VCC1_8 +0.06 | V | 30 Ω load to ground, buffer pulling up |
| $V_{OH\_HL}$ | (d) | Hub Interface Output High Voltage | VCC1_8 - 0.1 | — | — | V | $I_{OH}$ = 1 mA |
| $I_{OL\_HL}$ | (d) | Hub Interface Output Low Current | 17.5 | — | 22.5 | mA | VCC1_8/3 |
| $I_{OH\_HL}$ | (d) | Hub Interface Output High Current | — | — | 1 | mA | |
| IL_HL | (d) | Hub Interface Input Leakage Current | — | — | ±10 | µA | 0<Vin< VCC1_8 |
| CIN_HL | (d) | Hub Interface Capacitance | — | — | 8 | pF | FC=1 MHz |
| **Hub Interface with Enhanced Buffer Mode Configured for 50 Ω** | | | | | | | |
| $V_{IL\_HL}$ | (d) | Hub Interface Input Low Voltage | — | — | HLREF - 0.15 | V | |
| $V_{IH\_HL}$ | (d) | Hub Interface Input High Voltage | HLREF + 0.15 | — | — | V | |
| $V_{OL\_HLPD}$ | (d) | Hub Interface Output Low Voltage | 0.33 x VCC1_8 -0.06 | — | 0.33 x VCC1_8 +0.06 | V | 50 Ω load to VCC1_8, buffer pulling down |
| $V_{OL\_HLPU}$ | (d) | Hub Interface Output Low Voltage | 0.33 x VCC1_8 -0.06 | — | 0.33 x VCC1_8 +0.06 | V | 25 Ω load to ground, buffer pulling up |
| $V_{OH\_HL}$ | (d) | Hub Interface Output High Voltage | VCC1_8 - 0.1 | — | — | V | $I_{OH}$ = 1 mA |
| $I_{OL\_HL}$ | (d) | Hub Interface Output Low Current | 21.3 | — | 27.3 | mA | VCC1_8/3 |

| Symbol | Signal Group | Parameter | Min | Nom | Max | Unit | Notes |
|---|---|---|---|---|---|---|---|
| $I_{OH\_HL}$ | (d) | Hub Interface Output High Current | — | — | 1 | mA | |
| $I_{L\_HL}$ | (d) | Hub Interface Input Leakage Current | — | — | ±10 | µA | 0<Vin< VCC1_8 |
| $C_{IN\_HL}$ | (d) | Hub Interface Capacitance | — | — | 8 | pF | $F_C$=1MHz |
| **1.8V CMOS Signals** | | | | | | | |
| $V_{IL\_C}$ | (f) | CMOS Input Low Voltage | -0.3 | — | (VCC1_8)/2 - 0.25 | V | |
| $V_{IH\_C}$ | (f) | CMOS Input High Voltage | (VCC1_8)/2 + 0.25 | — | VCC1_8 + 0.3 | V | |
| $V_{OL\_C}$ | (e) | CMOS Output Low Voltage | — | — | 0.3 | V | |
| $V_{OH\_C}$ | (e) | CMOS Output High Voltage | VCC1_8 - 0.3 | — | — | V | |
| $I_{OL\_C}$ | (e) | CMOS Output Low Current | — | — | 1 | mA | $V_{OL}$=0.3 V |
| $I_{OH\_C}$ | (e) | CMOS Output High Current | -0.25 | — | — | mA | $V_{OH}$=VCC1_8 - 0.3 V |
| $I_{L\_C}$ | (f) | CMOS Input Leakage Current | —v | — | ±10 | µA | 0<Vin< VCC1_8 |
| $C_{IN\_C}$ | (f) | CMOS Input Capacitance | — | — | 9 | pF | $F_C$=1 MHz |
| **Miscellaneous Interface** | | | | | | | |
| $V_{IL}$ | (g) | CMOS Input Low Voltage | -0.3 | — | 0.7 | V | |
| $V_{IH}$ | (g) | CMOS Input High Voltage | 1.7 | — | 2.625 | V | |
| $I_L$ | (g) | CMOS Input Leakage Current | — | — | ±10 | µA | |
| $C_{IN}$ | (g) | CMOS Input Capacitance | — | — | 7 | pF | |
| $C_{IN\_CLK}$ | (h) | CMOS Clock Input Capacitance | — | — | TBD | pF | $F_C$=1 MHz |

This page is intentionally left blank.

# 7 Ballout and Package Information

## 7.1.1 Ballout Information

This section lists the MCH ballout assignment. Figure 10 and Figure 11 show the footprint ballout assignment from a top view of he package. Table 30 lists the ballout assignment.

If Hub Interface_B or Hub Interface_C is not used, only the corresponding HLSWNG and HLREF should be connected to VCC1_8 and the hub interface VREF circuit respectively. The rest of the unused hub interface pins may be left as no connect.

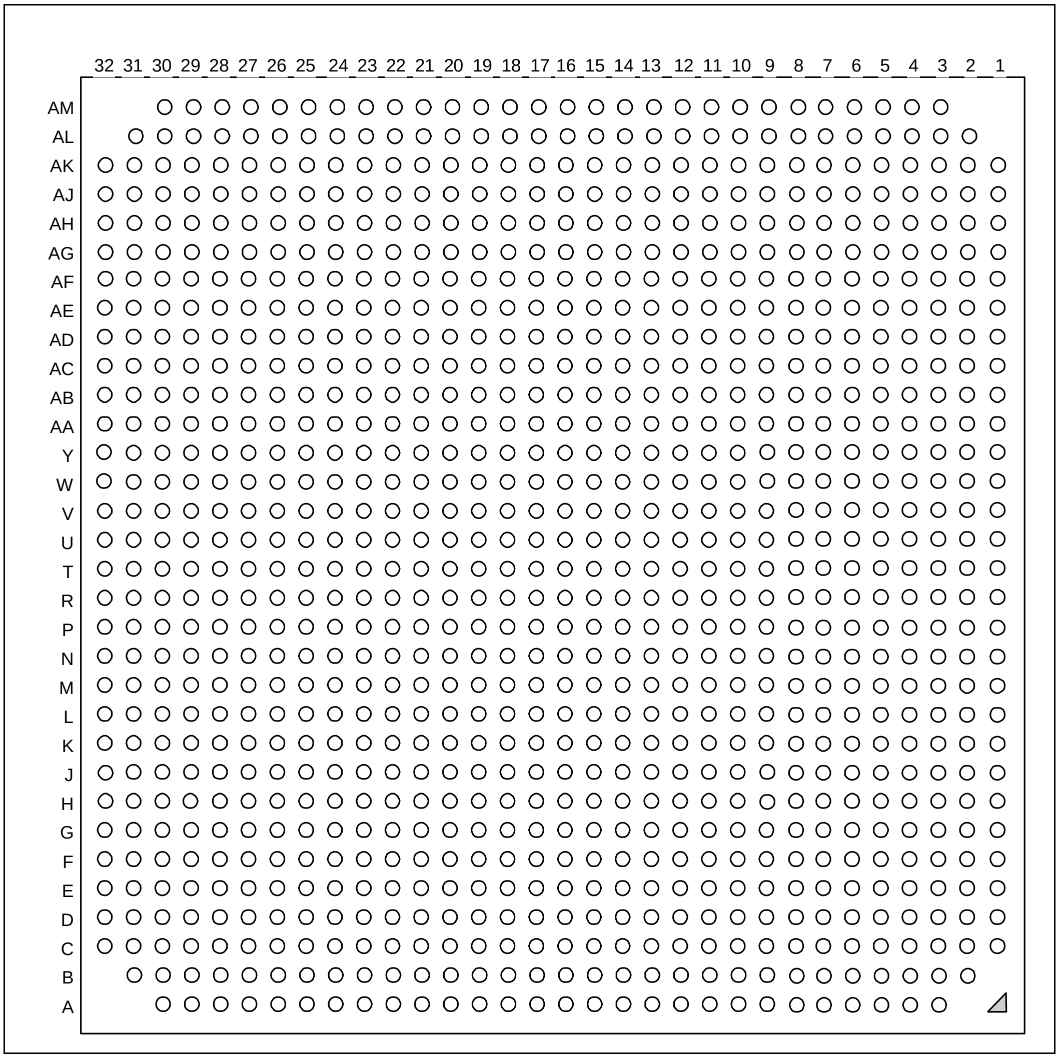
## Figure 10. MCH Ballout with AGP and Hub Interface Ball Names (Top View — Left Side)

| | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AM | | | G_AD8 | VDDQ | G_AD15 | G_C/BE1# | VSS | G_AD18 | G_AD22 | VDDQ | G_AD24 | G_AD28 | VSS | SB_STB | WBF# | G_GNT# |
| AL | | G_AD6 | G_C/BE0# | G_AD11 | VSS | Rsvd | G_C/BE2# | VDDQ | Rsvd | G_C/BE3# | VSS | AD_STB1 | SBA7 | VDDQ | SBA0 | ST0 |
| AK | G_AD2 | G_AD5 | VSS | AD_STB0 | Rsvd | VDDQ | G_IRDY# | Rsvd | VSS | G_AD21 | AD_STB1# | VDDQ | G_AD31 | SB_STB# | VSS | ST1 |
| AJ | VDDQ | G_AD4 | AD_STB0# | VDDQ | G_AD14 | G_STOP# | VSS | G_AD17 | G_AD20 | VDDQ | Rsvd | G_AD27 | VSS | SBA4 | SBA1 | ST2 |
| AH | G_AD1 | VSS | Rsvd | G_AD9 | VSS | G_TRDY# | G_DEVSEL# | VDDQ | G_AD19 | Rsvd | VSS | G_AD26 | G_AD30 | VDDQ | SBA2 | RBF# |
| AG | G_AD0 | Rsvd | VDDQ | G_AD10 | G_AD13 | G_SERR# | G_FRAME# | G_AD16 | VSS | 66IN | G_AD23 | VDDQ | G_AD29 | SBA5 | SBA3 | PIPE# |
| AF | G_AD3 | VSS | G_AD7 | VSS | G_AD12 | G_PAR | GREF_0 | HL1_8 | Rsvd | VSS | VCC1_8 | G_AD25 | G_SWNG | SBA6 | GREF_1 | GRCOMP |
| AE | VSS | DQA_A6 | VSS | DQA_A8 | VSS | DQA_A7 | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 |
| AD | VSS | VSS | DQA_A4 | VSS | DQA_A5 | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 |
| AC | VSS | DQA_A2 | VSS | DQA_A0 | VSS | DQA_A1 | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 |
| AB | VSS | VSS | VCC1_8 | VSS | DQA_A3 | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 |
| AA | CTM_A | CTM_A# | VSS | CHA_REF1 | VSS | VCC1_8 | VSS | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS |
| Y | CFM_A# | CFM_A | VSS | CHA_REF0 | VSS | VCC1_8 | VSS | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS |
| W | VSS | VSS | VCC1_8 | VSS | RQ_A7 | VSS | VSS | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS |
| V | VSS | RQ_A6 | VSS | EXP_A1 | VSS | RQ_A5 | VSS | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS |
| U | RQ_A4 | VSS | EXP_A0 | VSS | RQ_A3 | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 |
| T | VSS | RQ_A2 | VSS | RQ_A1 | VSS | DQB_A0 | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 |
| R | RQ_A0 | VSS | VCC1_8 | VSS | VSS | VCC1_8 | VSS | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS |
| P | VSS | DQB_A1 | VSS | DQB_A2 | VSS | DQB_A4 | VSS | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS |
| N | VSS | VSS | DQB_A6 | VSS | DQB_A8 | VSS | VSS | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS |
| M | VSS | DQB_A5 | VSS | DQB_A3 | VSS | DQB_A7 | VSS | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS |
| L | CMD_A | VSS | SIO_A | VCC1_8 | VSS | VCC1_8 | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 |
| K | SCK_A | CHA_HCLKOUT | CHA_RCLKOUT | VSS | HLREF_B | HLRCOMP_B | HLSWNG_B | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 |
| J | HL_B14 | HL_B8 | VCC1_8 | HL_B15 | HL_B11 | VCC1_8 | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 |
| H | HLB_STB1 | VSS | HL_B12 | HL_B13 | VSS | HL_B18 | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 |
| G | VCC1_8 | HLB_STB1# | HL_B16 | VCC1_8 | HL_B19 | HL_B17 | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS | VSS |
| F | HL_B6 | HL_B5 | VSS | HL_B10 | HL_B9 | HLREF_A | HL_A10 | VCC1_8 | VSS | DQA_B1 | VSS | VCC1_8 | VCC1_8 | VSS | RQ_B5 | VSS |
| E | HLB_STB0 | VCC1_8 | HL_B7 | HLRCOMP_A | VSS | HL_A9 | HL_A8 | VSS | DQA_B5 | VSS | DQA_B3 | VSS | VSS | RQ_B7 | VSS | RQ_B3 |
| D | VSS | HLB_STB0# | HL_B1 | HL_A0 | HLA_STB | VCC1_8 | VCC1_8 | DQA_B7 | VSS | DQA_B0 | VSS | CHB_REF1 | CHB_REF0 | VSS | EXP_B1 | VSS |
| C | HL_B4 | HL_B3 | VSS | HL_A1 | VSS | HLA_STB# | HL_A11 | VSS | DQA_B6 | VSS | VCC1_8 | VSS | VSS | VCC1_8 | VSS | EXP_B0 |
| B | | VCC1_8 | HL_B0 | VCC1_8 | HL_A3 | HL_A5 | VSS | DQA_B8 | VSS | DQA_B2 | VSS | CTM_B# | CFM_B | VSS | RQ_B6 | VSS |
| A | | | HL_B2 | HL_A2 | HL_A4 | HL_A6 | HL_A7 | VSS | DQA_B4 | VSS | VSS | CTM_B | CFM_B# | VSS | VSS | RQ_B4 |
| | 32 | 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 |

## Figure 11. MCH Ballout with AGP and Hub Interface Ball Names (Top View — Right Side)

| 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| RSTIN# | BRO# | HTRDY# | DRDY# | BNR# | RS0# | RS2# | VTT | HA27# | HA26# | VTT | HA20# | HA13# | VTT | | | AM |
| G_REQ# | RESERVD | VTT | HIT# | VSS | CPURST# | RS1# | HA32# | VSS | HA35# | HA34# | VSS | HA14# | HA10# | HA16# | | AL |
| TESTIN# | Rsvd | RSP# | HITM# | BPRI# | AP0# | VSS | HA31# | HA33# | VTT | HA22# | HA30# | VTT | HA12# | HA15# | VTT | AK |
| OVERT# | Rsvd | BERR# | VSS | DBSY# | VTT | HA21# | VTT | HA23# | HADSTB1# | VSS | HA25# | HA11# | VSS | HADSTB0# | HA9# | AJ |
| BUSPARK | HLA_ENH# | VSS | HLOCK# | DEFER# | AP1# | HA29# | HA19# | VSS | HA24# | HA28# | VTT | HA8# | HA6# | VTT | HA5# | AH |
| VSS | VSS | VTT | CCVREF | VTT | ADS# | VTT | HA18# | VSS | BCLK1 | BCLK0 | VCC1_8 | VSS | HREQ0# | HA7# | VSS | AG |
| VSS | VSS | VSS | HRCOMP0 | HSWNG0 | VTT | VSS | VTT | HAVREF0 | HA17# | VTT | HREQ3# | HA4# | VTT | HREQ1# | HREQ2# | AF |
| VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | HA3# | VSS | HREQ4# | HD59# | VSS | HD63# | AE |
| VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VTT | HAVREF1 | HD60# | VTT | HD58# | HD57# | VTT | AD |
| VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | HD56# | HD61# | VSS | HD62# | DBI3# | AC |
| VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VTT | HD51# | VTT | HD55# | HD53# | VTT | HD54# | AB |
| VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VSS | HD49# | HD50# | VSS | HDSTBN3# | HDSTBP3# | VSS | AA |
| VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VTT | VTT | HD46# | HD47# | VTT | HD48# | HD52# | Y |
| VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VSS | HDVREF3 | VSS | HD44# | HD45# | VSS | HD42# | W |
| VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VTT | HD43# | HD40# | VTT | HD41# | DBI2# | VTT | V |
| VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | HD37# | HD39# | VSS | HD35# | HD38# | U |
| VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VTT | HDVREF2 | VTT | HD32# | HDSTBN2# | VTT | HDSTBP2# | T |
| VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VSS | HD36# | HD34# | VSS | HD33# | DP3# | VSS | R |
| VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | HRCOMP1 | VTT | DP2# | DP1# | VTT | HD31# | DP0# | P |
| VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | HSWNG1 | HD30# | VSS | HDSTBN1# | HDSTBP1# | VSS | HD24# | N |
| VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | VSS | VSS | VTT | HDVREF1 | HD27# | VTT | HD25# | HD28# | VTT | M |
| VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VSS | HD23# | HD22# | VSS | HD26# | HD29# | L |
| VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VTT | HD16# | VTT | HD18# | HD20# | VTT | DBI1# | K |
| VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | HDVREF0 | HD17# | VSS | HD19# | HD21# | VSS | J |
| VCC1_8 | VSS | VSS | VSS | VSS | VCC1_8 | VCC1_8 | VCC1_8 | VCC1_8 | VSS | VTT | HD15# | HD14# | VTT | HD12# | HDSTBP0# | H |
| VSS | VSS | VSS | VSS | VSS | VCC1_8 | VSS | VCC1_8 | HLSWNG_C | VCC1_8 | HD13# | VSS | HD10# | HDSTBN0# | VSS | HD11# | G |
| DQB_B0 | VCC1_8 | DQB_B4 | VSS | DQB_B7 | VSS | HL_C11 | HL_C13 | HL_C9 | HLRCOMP_C | HLREF_C | HL_C0 | VTT | HD9# | HD5# | VTT | F |
| VSS | VSS | VSS | DQB_B8 | VSS | VSS | VCC1_8 | HL_C8 | HL_C14 | VCC1_8 | HL_C2 | HL_C1 | HD3# | VSS | HD6# | HD2# | E |
| RQ_B1 | VSS | DQB_B2 | VSS | DQB_B3 | VCC1_8 | HL_C15 | VSS | HLC_STB1# | HL_C17 | VSS | HLC_STB0# | HD7# | HD0# | VTT | HD8# | D |
| VSS | VCC1_8 | VSS | DQB_B6 | VSS | SIO_B | CHB_RCLKOUT | HLC_STB1 | VCC1_8 | HL_C19 | HLC_STB0 | VCC1_8 | HL_C3 | VSS | HD4# | VSS | C |
| RQ_B2 | VSS | DQB_B1 | VSS | DQB_B5 | VSS | CHB_HCLKOUT | HL_C12 | HL_C10 | VSS | HL_C6 | HL_C7 | VSS | VTT | HD1# | | B |
| VSS | RQ_B0 | VSS | VSS | VSS | CMD_B | SCK_B | VCC1_8 | HL_C16 | HL_C18 | VCC1_8 | HL_C4 | HL_C5 | DBI0# | | | A |
| 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | |

## Figure 12. MCH Ballout Topside View (looking through the top of the package)



### Table 30. MCH Alphabetical Ballout List

| Signal | Ball # |
|---|---|
| 66IN | AG23 |
| AD_STB0 | AK29 |
| AD_STB0# | AJ30 |
| AD_STB1 | AL21 |
| AD_STB1# | AK22 |
| ADS# | AG11 |
| AP0# | AK11 |
| AP1# | AH11 |
| BCLK0 | AG6 |
| BCLK1 | AG7 |
| BERR# | AJ14 |
| BNR# | AM12 |
| BPRI# | AK12 |
| BRO# | AM15 |
| BUSPARK | AH16 |
| CCVREF | AG13 |
| CFM_A | Y31 |
| CFM_A# | Y32 |
| CFM_B | B20 |
| CFM_B# | A20 |
| CHA_HCLKOUT | K31 |
| CHA_RCLKOUT | K30 |
| CHA_REF0 | Y29 |
| CHA_REF1 | AA29 |
| CHB_HCLKOUT | B10 |
| CHB_RCLKOUT | C10 |
| CHB_REF0 | D20 |
| CHB_REF1 | D21 |
| CMD_A | L32 |
| CMD_B | A11 |
| CPURST# | AL11 |
| CTM_A | AA32 |
| CTM_A# | AA31 |
| CTM_B | A21 |
| CTM_B# | B21 |
| DBI0# | A3 |
| DBI1# | K1 |
| DBI2# | V2 |
| DBI3# | AC1 |
| DBSY# | AJ12 |
| DEFER# | AH12 |
| DP0# | P1 |
| DP1# | P4 |
| DP2# | P5 |
| DP3# | R2 |
| DQA_A0 | AC29 |
| DQA_A1 | AC27 |
| DQA_A2 | AC31 |
| DQA_A3 | AB28 |
| DQA_A4 | AD30 |
| DQA_A5 | AD28 |
| DQA_A6 | AE31 |
| DQA_A7 | AE27 |
| DQA_A8 | AE29 |
| DQA_B0 | D23 |
| DQA_B1 | F23 |
| DQA_B2 | B23 |
| DQA_B3 | E22 |
| DQA_B4 | A24 |
| DQA_B5 | E24 |
| DQA_B6 | C24 |
| DQA_B7 | D25 |
| DQA_B8 | B25 |
| DQB_A0 | T27 |
| DQB_A1 | P31 |
| DQB_A2 | P29 |
| DQB_A3 | M29 |
| DQB_A4 | P27 |
| DQB_A5 | M31 |
| DQB_A6 | N30 |
| DQB_A7 | M27 |
| DQB_A8 | N28 |
| DQB_B0 | F16 |
| DQB_B1 | B14 |
| DQB_B2 | D14 |
| DQB_B3 | D12 |
| DQB_B4 | F14 |
| DQB_B5 | B12 |
| DQB_B6 | C13 |
| DQB_B7 | F12 |
| DQB_B8 | E13 |
| DRDY# | AM13 |
| EXP_A0 | U30 |
| EXP_A1 | V29 |
| EXP_B0 | C17 |
| EXP_B1 | D18 |
| G_AD0 | AG32 |
| G_AD1 | AH32 |
| G_AD2 | AK32 |
| G_AD3 | AF32 |
| G_AD4 | AJ31 |
| G_AD5 | AK31 |
| G_AD6 | AL31 |
| G_AD7 | AF30 |
| G_AD8 | AM30 |
| G_AD9 | AH29 |
| G_AD10 | AG29 |
| G_AD11 | AL29 |
| G_AD12 | AF28 |
| G_AD13 | AG28 |
| G_AD14 | AJ28 |
| G_AD15 | AM28 |

| Signal | Ball # |
|---|---|
| G_AD16 | AG25 |
| G_AD17 | AJ25 |
| G_AD18 | AM25 |
| G_AD19 | AH24 |
| G_AD20 | AJ24 |
| G_AD21 | AK23 |
| G_AD22 | AM24 |
| G_AD23 | AG22 |
| G_AD24 | AM22 |
| G_AD25 | AF21 |
| G_AD26 | AH21 |
| G_AD27 | AJ21 |
| G_AD28 | AM21 |
| G_AD29 | AG20 |
| G_AD30 | AH20 |
| G_AD31 | AK20 |
| G_C/BE0# | AL30 |
| G_C/BE1# | AM27 |
| G_C/BE2# | AL26 |
| G_C/BE3# | AL23 |
| G_DEVSEL# | AH26 |
| G_FRAME# | AG26 |
| G_GNT# | AM17 |
| G_IRDY# | AK26 |
| G_PAR | AF27 |
| G_REQ# | AL16 |
| G_SERR# | AG27 |
| G_STOP# | AJ27 |
| G_SWNG | AF20 |
| G_TRDY# | AH27 |
| GRCOMP | AF17 |
| GREF_0 | AF26 |
| GREF_1 | AF18 |
| HA3# | AE6 |
| HA4# | AF4 |

| Signal | Ball # |
|---|---|
| HA5# | AH1 |
| HA6# | AH3 |
| HA7# | AG2 |
| HA8# | AH4 |
| HA9# | AJ1 |
| HA10# | AL3 |
| HA11# | AJ4 |
| HA12# | AK3 |
| HA13# | AM4 |
| HA14# | AL4 |
| HA15# | AK2 |
| HA16# | AL2 |
| HA17# | AF7 |
| HA18# | AG9 |
| HA19# | AH9 |
| HA20# | AM5 |
| HA21# | AJ10 |
| HA22# | AK6 |
| HA23# | AJ8 |
| HA24# | AH7 |
| HA25# | AJ5 |
| HA26# | AM7 |
| HA27# | AM8 |
| HA28# | AH6 |
| HA29# | AH10 |
| HA30# | AK5 |
| HA31# | AK9 |
| HA32# | AL9 |
| HA33# | AK8 |
| HA34# | AL6 |
| HA35# | AL7 |
| HADSTB0# | AJ2 |
| HADSTB1# | AJ7 |
| HAVREF0 | AF8 |
| HAVREF1 | AD6 |

| Signal | Ball # |
|---|---|
| HD0# | D3 |
| HD1# | B2 |
| HD2# | E1 |
| HD3# | E4 |
| HD4# | C2 |
| HD5# | F2 |
| HD6# | E2 |
| HD7# | D4 |
| HD8# | D1 |
| HD9# | F3 |
| HD10# | G4 |
| HD11# | G1 |
| HD12# | H2 |
| HD13# | G6 |
| HD14# | H4 |
| HD15# | H5 |
| HD16# | K6 |
| HD17# | J5 |
| HD18# | K4 |
| HD19# | J3 |
| HD20# | K3 |
| HD21# | J2 |
| HD22# | L4 |
| HD23# | L5 |
| HD24# | N1 |
| HD25# | M3 |
| HD26# | L2 |
| HD27# | M5 |
| HD28# | M2 |
| HD29# | L1 |
| HD30# | N6 |
| HD31# | P2 |
| HD32# | T4 |
| HD33# | R3 |
| HD34# | R5 |

| Signal | Ball # |
|---|---|
| HD35# | U2 |
| HD36# | R6 |
| HD37# | U5 |
| HD38# | U1 |
| HD39# | U4 |
| HD40# | V5 |
| HD41# | V3 |
| HD42# | W1 |
| HD43# | V6 |
| HD44# | W4 |
| HD45# | W3 |
| HD46# | Y5 |
| HD47# | Y4 |
| HD48# | Y2 |
| HD49# | AA6 |
| HD50# | AA5 |
| HD51# | AB6 |
| HD52# | Y1 |
| HD53# | AB3 |
| HD54# | AB1 |
| HD55# | AB4 |
| HD56# | AC5 |
| HD57# | AD2 |
| HD58# | AD3 |
| HD59# | AE3 |
| HD60# | AD5 |
| HD61# | AC4 |
| HD62# | AC2 |
| HD63# | AE1 |
| HDSTBN0# | G3 |
| HDSTBN1# | N4 |
| HDSTBN2# | T3 |
| HDSTBN3# | AA3 |
| HDSTBP0# | H1 |
| HDSTBP1# | N3 |

| Signal | Ball # |
|---|---|
| HDSTBP2# | T1 |
| HDSTBP3# | AA2 |
| HDVREF0 | J6 |
| HDVREF1 | M6 |
| HDVREF2 | T6 |
| HDVREF3 | W6 |
| HIT# | AL13 |
| HITM# | AK13 |
| HL_A0 | D29 |
| HL_A1 | C29 |
| HL_A2 | A29 |
| HL_A3 | B28 |
| HL_A4 | A28 |
| HL_A5 | B27 |
| HL_A6 | A27 |
| HL_A7 | A26 |
| HL_A8 | E26 |
| HL_A9 | E27 |
| HL_A10 | F26 |
| HL_A11 | C26 |
| HL_B0 | B30 |
| HL_B1 | D30 |
| HL_B2 | A30 |
| HL_B3 | C31 |
| HL_B4 | C32 |
| HL_B5 | F31 |
| HL_B6 | F32 |
| HL_B7 | E30 |
| HL_B8 | J31 |
| HL_B9 | F28 |
| HL_B10 | F29 |
| HL_B11 | J28 |
| HL_B12 | H30 |
| HL_B13 | H29 |
| HL_B14 | J32 |

| Signal | Ball # |
|---|---|
| HL_B15 | J29 |
| HL_B16 | G30 |
| HL_B17 | G27 |
| HL_B18 | H27 |
| HL_B19 | G28 |
| HL_C0 | F5 |
| HL_C1 | E5 |
| HL_C2 | E6 |
| HL_C3 | C4 |
| HL_C4 | A5 |
| HL_C5 | A4 |
| HL_C6 | B6 |
| HL_C7 | B5 |
| HL_C8 | E9 |
| HL_C9 | F8 |
| HL_C10 | B8 |
| HL_C11 | F10 |
| HL_C12 | B9 |
| HL_C13 | F9 |
| HL_C14 | E8 |
| HL_C15 | D10 |
| HL_C16 | A8 |
| HL_C17 | D7 |
| HL_C18 | A7 |
| HL_C19 | C7 |
| HL1_8 | AF25 |
| HLA_ENH# | AH15 |
| HLA_STB | D28 |
| HLA_STB# | C27 |
| HLB_STB0 | E32 |
| HLB_STB0# | D31 |
| HLB_STB1 | H32 |
| HLB_STB1# | G31 |
| HLC_STB0 | C6 |
| HLC_STB0# | D5 |

| Signal | Ball # |
|---|---|
| HLC_STB1 | C9 |
| HLC_STB1# | D8 |
| HLOCK# | AH13 |
| HLRCOMP_A | E29 |
| HLRCOMP_B | K27 |
| HLRCOMP_C | F7 |
| HLREF_A | F27 |
| HLREF_B | K28 |
| HLREF_C | F6 |
| HLSWNG_B | K26 |
| HLSWNG_C | G8 |
| HRCOMP0 | AF13 |
| HRCOMP1 | P7 |
| HREQ0# | AG3 |
| HREQ1# | AF2 |
| HREQ2# | AF1 |
| HREQ3# | AF5 |
| HREQ4# | AE4 |
| HSWNG0 | AF12 |
| HSWNG1 | N7 |
| HTRDY# | AM14 |
| OVERT# | AJ16 |
| PIPE# | AG17 |
| RBF# | AH17 |
| RQ_A0 | R32 |
| RQ_A1 | T29 |
| RQ_A2 | T31 |
| RQ_A3 | U28 |
| RQ_A4 | U32 |
| RQ_A5 | V27 |
| RQ_A6 | V31 |
| RQ_A7 | W28 |
| RQ_B0 | A15 |
| RQ_B1 | D16 |
| RQ_B2 | B16 |
| RQ_B3 | E17 |
| RQ_B4 | A17 |
| RQ_B5 | F18 |
| RQ_B6 | B18 |
| RQ_B7 | E19 |
| RS0# | AM11 |
| RS1# | AL10 |
| RS2# | AM10 |
| RSP# | AK14 |
| RSTIN# | AM16 |
| Rsvd | AL15 |
| Rsvd | AK15 |
| Rsvd | AJ15 |
| Rsvd | AH30 |
| Rsvd | AG31 |
| Rsvd | AL27 |
| Rsvd | AK28 |
| Rsvd | AL24 |
| Rsvd | AK25 |
| Rsvd | AJ22 |
| Rsvd | AH23 |
| Rsvd | AF24 |
| SB_STB | AM19 |
| SB_STB# | AK19 |
| SBA0 | AL18 |
| SBA1 | AJ18 |
| SBA2 | AH18 |
| SBA3 | AG18 |
| SBA4 | AJ19 |
| SBA5 | AG19 |
| SBA6 | AF19 |
| SBA7 | AL20 |
| SCK_A | K32 |
| SCK_B | A10 |
| SIO_A | L30 |
| SIO_B | C11 |
| ST0 | AL17 |
| ST1 | AK17 |
| ST2 | AJ17 |
| TESTIN# | AK16 |
| VCC1_8 | AB25 |
| VCC1_8 | U25 |
| VCC1_8 | T25 |
| VCC1_8 | L25 |
| VCC1_8 | K25 |
| VCC1_8 | J25 |
| VCC1_8 | H25 |
| VCC1_8 | AE24 |
| VCC1_8 | AD24 |
| VCC1_8 | AC24 |
| VCC1_8 | AB24 |
| VCC1_8 | U24 |
| VCC1_8 | T24 |
| VCC1_8 | L24 |
| VCC1_8 | K24 |
| VCC1_8 | J24 |
| VCC1_8 | H24 |
| VCC1_8 | AE23 |
| VCC1_8 | AD23 |
| VCC1_8 | AC23 |
| VCC1_8 | AB23 |
| VCC1_8 | U23 |
| VCC1_8 | T23 |
| VCC1_8 | L23 |
| VCC1_8 | K23 |
| VCC1_8 | J23 |
| VCC1_8 | H23 |
| VCC1_8 | AF22 |
| VCC1_8 | AE22 |
| VCC1_8 | AD22 |

| Signal | Ball # |
|---|---|
| VCC1_8 | AC22 |
| VCC1_8 | AB22 |
| VCC1_8 | U22 |
| VCC1_8 | T22 |
| VCC1_8 | L22 |
| VCC1_8 | K22 |
| VCC1_8 | J22 |
| VCC1_8 | H22 |
| VCC1_8 | Y21 |
| VCC1_8 | W21 |
| VCC1_8 | V21 |
| VCC1_8 | R21 |
| VCC1_8 | P21 |
| VCC1_8 | N21 |
| VCC1_8 | M21 |
| VCC1_8 | AA20 |
| VCC1_8 | Y20 |
| VCC1_8 | T16 |
| VCC1_8 | L16 |
| VCC1_8 | W20 |
| VCC1_8 | V20 |
| VCC1_8 | R20 |
| VCC1_8 | P20 |
| VCC1_8 | N20 |
| VCC1_8 | M20 |
| VCC1_8 | AA19 |
| VCC1_8 | Y19 |
| VCC1_8 | W19 |
| VCC1_8 | V19 |
| VCC1_8 | R19 |
| VCC1_8 | P19 |
| VCC1_8 | N19 |
| VCC1_8 | M19 |
| VCC1_8 | AA18 |
| VCC1_8 | Y18 |
| VCC1_8 | W18 |
| VCC1_8 | V18 |
| VCC1_8 | R18 |
| VCC1_8 | P18 |
| VCC1_8 | N18 |
| VCC1_8 | M18 |
| VCC1_8 | AE17 |
| VCC1_8 | AD17 |
| VCC1_8 | AC17 |
| VCC1_8 | AB17 |
| VCC1_8 | U17 |
| VCC1_8 | T17 |
| VCC1_8 | L17 |
| VCC1_8 | K17 |
| VCC1_8 | J17 |
| VCC1_8 | H17 |
| VCC1_8 | AE16 |
| VCC1_8 | AD16 |
| VCC1_8 | AC16 |
| VCC1_8 | AB16 |
| VCC1_8 | U16 |
| VCC1_8 | K16 |
| VCC1_8 | J16 |
| VCC1_8 | H16 |
| VCC1_8 | AA15 |
| VCC1_8 | Y15 |
| VCC1_8 | W15 |
| VCC1_8 | V15 |
| VCC1_8 | R15 |
| VCC1_8 | P15 |
| VCC1_8 | N15 |
| VCC1_8 | AE10 |
| VCC1_8 | AD10 |
| VCC1_8 | M15 |
| VCC1_8 | F15 |
| VCC1_8 | C15 |
| VCC1_8 | AA14 |
| VCC1_8 | Y14 |
| VCC1_8 | W14 |
| VCC1_8 | V14 |
| VCC1_8 | R14 |
| VCC1_8 | P14 |
| VCC1_8 | N14 |
| VCC1_8 | M14 |
| VCC1_8 | AA13 |
| VCC1_8 | Y13 |
| VCC1_8 | W13 |
| VCC1_8 | V13 |
| VCC1_8 | R13 |
| VCC1_8 | P13 |
| VCC1_8 | N13 |
| VCC1_8 | M13 |
| VCC1_8 | AA12 |
| VCC1_8 | Y12 |
| VCC1_8 | W12 |
| VCC1_8 | V12 |
| VCC1_8 | R12 |
| VCC1_8 | P12 |
| VCC1_8 | N12 |
| VCC1_8 | M12 |
| VCC1_8 | AE11 |
| VCC1_8 | AD11 |
| VCC1_8 | AC11 |
| VCC1_8 | AB11 |
| VCC1_8 | U11 |
| VCC1_8 | T11 |
| VCC1_8 | L11 |
| VCC1_8 | K11 |
| VCC1_8 | J11 |
| VCC1_8 | H11 |

| Signal | Ball # |
|---|---|
| VCC1_8 | G11 |
| VCC1_8 | D11 |
| VCC1_8 | AC10 |
| VCC1_8 | AB10 |
| VCC1_8 | U10 |
| VCC1_8 | T10 |
| VCC1_8 | L10 |
| VCC1_8 | K10 |
| VCC1_8 | J10 |
| VCC1_8 | H10 |
| VCC1_8 | E10 |
| VCC1_8 | AE9 |
| VCC1_8 | G32 |
| VCC1_8 | E31 |
| VCC1_8 | B31 |
| VCC1_8 | J30 |
| VCC1_8 | L29 |
| VCC1_8 | G29 |
| VCC1_8 | B29 |
| VCC1_8 | L27 |
| VCC1_8 | J27 |
| VCC1_8 | D27 |
| VCC1_8 | H26 |
| VCC1_8 | D26 |
| VCC1_8 | AE25 |
| VCC1_8 | AD25 |
| VCC1_8 | AC25 |
| VCC1_8 | AD9 |
| VCC1_8 | AC9 |
| VCC1_8 | AB9 |
| VCC1_8 | U9 |
| VCC1_8 | T9 |
| VCC1_8 | L9 |
| VCC1_8 | K9 |
| VCC1_8 | J9 |

| Signal | Ball # |
|---|---|
| VCC1_8 | H9 |
| VCC1_8 | G9 |
| VCC1_8 | A9 |
| VCC1_8 | AE8 |
| VCC1_8 | AD8 |
| VCC1_8 | AC8 |
| VCC1_8 | AB8 |
| VCC1_8 | U8 |
| VCC1_8 | T8 |
| VCC1_8 | L8 |
| VCC1_8 | K8 |
| VCC1_8 | J8 |
| VCC1_8 | H8 |
| VCC1_8 | C8 |
| VCC1_8 | AG5 |
| VCC1_8 | G7 |
| VCC1_8 | E7 |
| VCC1_8 | A6 |
| VCC1_8 | C5 |
| VCC1_8 | AA21 |
| VCC1_8RAC | F25 |
| VCC1_8RAC | F21 |
| VCC1_8RAC | F20 |
| VCC1_8RAC | C19 |
| VCC1_8RAC | AB30 |
| VCC1_8RAC | W30 |
| VCC1_8RAC | R30 |
| VCC1_8RAC | AA27 |
| VCC1_8RAC | Y27 |
| VCC1_8RAC | R27 |
| VCC1_8RAC | C22 |
| VDDQ | AG21 |
| VDDQ | AL19 |
| VDDQ | AJ32 |
| VDDQ | AG30 |

| Signal | Ball # |
|---|---|
| VDDQ | AM29 |
| VDDQ | AJ29 |
| VDDQ | AK27 |
| VDDQ | AL25 |
| VDDQ | AH25 |
| VDDQ | AM23 |
| VDDQ | AJ23 |
| VDDQ | AK21 |
| VDDQ | AH19 |
| VSS | M32 |
| VSS | D32 |
| VSS | AH31 |
| VSS | AF31 |
| VSS | AD31 |
| VSS | AB31 |
| VSS | W31 |
| VSS | U31 |
| VSS | R31 |
| VSS | N31 |
| VSS | L31 |
| VSS | H31 |
| VSS | AK30 |
| VSS | AE30 |
| VSS | AC30 |
| VSS | AA30 |
| VSS | Y30 |
| VSS | V30 |
| VSS | T30 |
| VSS | P30 |
| VSS | M30 |
| VSS | F30 |
| VSS | C30 |
| VSS | AF29 |
| VSS | AD29 |
| VSS | AB29 |

| Signal | Ball # |
|---|---|
| VSS | W29 |
| VSS | U29 |
| VSS | R29 |
| VSS | N29 |
| VSS | K29 |
| VSS | AL28 |
| VSS | AH28 |
| VSS | AE28 |
| VSS | AC28 |
| VSS | AA28 |
| VSS | Y28 |
| VSS | V28 |
| VSS | T28 |
| VSS | R28 |
| VSS | P28 |
| VSS | M28 |
| VSS | L28 |
| VSS | H28 |
| VSS | E28 |
| VSS | C28 |
| VSS | AD27 |
| VSS | AB27 |
| VSS | W27 |
| VSS | U27 |
| VSS | N27 |
| VSS | M16 |
| VSS | G16 |
| VSS | E16 |
| VSS | C16 |
| VSS | A16 |
| VSS | AG15 |
| VSS | AF15 |
| VSS | AE15 |
| VSS | AD15 |
| VSS | AC15 |
| VSS | AB15 |
| VSS | U15 |
| VSS | T15 |
| VSS | L15 |
| VSS | K15 |
| VSS | J15 |
| VSS | H15 |
| VSS | G15 |
| VSS | E15 |
| VSS | D15 |
| VSS | B15 |
| VSS | AH14 |
| VSS | AF14 |
| VSS | AE14 |
| VSS | AD14 |
| VSS | AC14 |
| VSS | AB14 |
| VSS | U14 |
| VSS | T14 |
| VSS | L14 |
| VSS | K14 |
| VSS | J14 |
| VSS | H14 |
| VSS | G14 |
| VSS | E14 |
| VSS | C14 |
| VSS | A14 |
| VSS | AJ13 |
| VSS | AE13 |
| VSS | AD13 |
| VSS | AC13 |
| VSS | AB13 |
| VSS | U13 |
| VSS | T13 |
| VSS | L13 |
| VSS | K13 |
| VSS | J13 |
| VSS | H13 |
| VSS | G13 |
| VSS | F13 |
| VSS | D13 |
| VSS | AM26 |
| VSS | AJ26 |
| VSS | AE26 |
| VSS | AD26 |
| VSS | AC26 |
| VSS | AB26 |
| VSS | AA26 |
| VSS | Y26 |
| VSS | W26 |
| VSS | V26 |
| VSS | U26 |
| VSS | T26 |
| VSS | R26 |
| VSS | P26 |
| VSS | N26 |
| VSS | M26 |
| VSS | L26 |
| VSS | J26 |
| VSS | G26 |
| VSS | B26 |
| VSS | AA25 |
| VSS | Y25 |
| VSS | W25 |
| VSS | V25 |
| VSS | R25 |
| VSS | P25 |
| VSS | N25 |
| VSS | M25 |
| VSS | G25 |

| Signal | Ball # |
|---|---|
| VSS | E25 |
| VSS | C25 |
| VSS | A25 |
| VSS | AK24 |
| VSS | AG24 |
| VSS | AA24 |
| VSS | Y24 |
| VSS | W24 |
| VSS | V24 |
| VSS | R24 |
| VSS | P24 |
| VSS | N24 |
| VSS | M24 |
| VSS | G24 |
| VSS | F24 |
| VSS | D24 |
| VSS | B24 |
| VSS | AF23 |
| VSS | AA23 |
| VSS | Y23 |
| VSS | W23 |
| VSS | V23 |
| VSS | B13 |
| VSS | A13 |
| VSS | AL12 |
| VSS | AE12 |
| VSS | AD12 |
| VSS | AC12 |
| VSS | AB12 |
| VSS | U12 |
| VSS | T12 |
| VSS | L12 |
| VSS | K12 |
| VSS | J12 |
| VSS | H12 |

| Signal | Ball # |
|---|---|
| VSS | G12 |
| VSS | E12 |
| VSS | C12 |
| VSS | A12 |
| VSS | AA11 |
| VSS | Y11 |
| VSS | W11 |
| VSS | V11 |
| VSS | R11 |
| VSS | P11 |
| VSS | N11 |
| VSS | M11 |
| VSS | F11 |
| VSS | E11 |
| VSS | B11 |
| VSS | AK10 |
| VSS | AF10 |
| VSS | AA10 |
| VSS | Y10 |
| VSS | W10 |
| VSS | V10 |
| VSS | R10 |
| VSS | P10 |
| VSS | N10 |
| VSS | M10 |
| VSS | G10 |
| VSS | AA9 |
| VSS | Y9 |
| VSS | W9 |
| VSS | V9 |
| VSS | R9 |
| VSS | P9 |
| VSS | N9 |
| VSS | M9 |
| VSS | D9 |

| Signal | Ball # |
|---|---|
| VSS | AL8 |
| VSS | AH8 |
| VSS | AG8 |
| VSS | R23 |
| VSS | P23 |
| VSS | N23 |
| VSS | M23 |
| VSS | G23 |
| VSS | E23 |
| VSS | C23 |
| VSS | A23 |
| VSS | AL22 |
| VSS | AH22 |
| VSS | AA22 |
| VSS | Y22 |
| VSS | W22 |
| VSS | V22 |
| VSS | R22 |
| VSS | P22 |
| VSS | N22 |
| VSS | M22 |
| VSS | G22 |
| VSS | F22 |
| VSS | D22 |
| VSS | B22 |
| VSS | A22 |
| VSS | AE21 |
| VSS | AD21 |
| VSS | AC21 |
| VSS | AB21 |
| VSS | U21 |
| VSS | T21 |
| VSS | L21 |
| VSS | K21 |
| VSS | J21 |

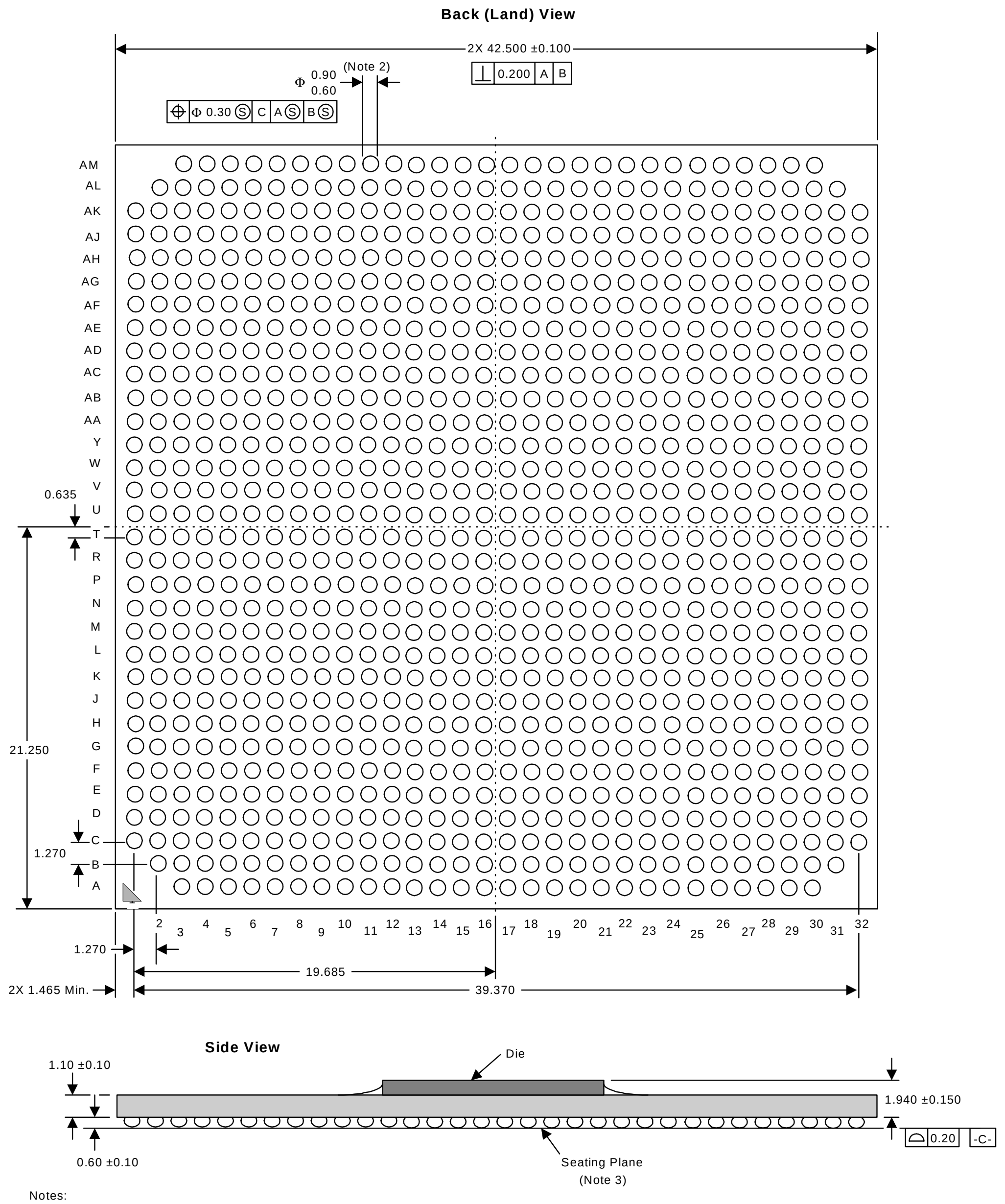
| Signal | Ball # |
|---|---|
| VSS | H21 |
| VSS | G21 |
| VSS | E21 |
| VSS | C21 |
| VSS | AM20 |
| VSS | AJ20 |
| VSS | AE20 |
| VSS | AD20 |
| VSS | AC20 |
| VSS | AB20 |
| VSS | U20 |
| VSS | T20 |
| VSS | L20 |
| VSS | K20 |
| VSS | J20 |
| VSS | H20 |
| VSS | G20 |
| VSS | E20 |
| VSS | C20 |
| VSS | AA8 |
| VSS | Y8 |
| VSS | W8 |
| VSS | V8 |
| VSS | R8 |
| VSS | P8 |
| VSS | N8 |
| VSS | M8 |
| VSS | AE7 |
| VSS | AC7 |
| VSS | AA7 |
| VSS | W7 |
| VSS | U7 |
| VSS | R7 |
| VSS | L7 |
| VSS | J7 |
| VSS | H7 |
| VSS | B7 |
| VSS | AJ6 |
| VSS | AC6 |
| VSS | U6 |
| VSS | L6 |
| VSS | D6 |
| VSS | AL5 |
| VSS | AE5 |
| VSS | W5 |
| VSS | N5 |
| VSS | G5 |
| VSS | AG4 |
| VSS | AA4 |
| VSS | R4 |
| VSS | J4 |
| VSS | B4 |
| VSS | U3 |
| VSS | L3 |
| VSS | E3 |
| VSS | C3 |
| VSS | AE2 |
| VSS | N2 |
| VSS | G2 |
| VSS | AG1 |
| VSS | AA1 |
| VSS | R1 |
| VSS | J1 |
| VSS | C1 |
| VSS | W2 |
| VSS | AE32 |
| VSS | AD32 |
| VSS | AC32 |
| VSS | AB32 |
| VSS | W32 |
| VSS | V32 |
| VSS | T32 |
| VSS | P32 |
| VSS | N32 |
| VSS | AJ3 |
| VSS | AC3 |
| VSS | AE19 |
| VSS | AD19 |
| VSS | AC19 |
| VSS | AB19 |
| VSS | U19 |
| VSS | T19 |
| VSS | L19 |
| VSS | K19 |
| VSS | J19 |
| VSS | H19 |
| VSS | G19 |
| VSS | F19 |
| VSS | D19 |
| VSS | B19 |
| VSS | A19 |
| VSS | AK18 |
| VSS | AE18 |
| VSS | AD18 |
| VSS | AC18 |
| VSS | AB18 |
| VSS | U18 |
| VSS | T18 |
| VSS | L18 |
| VSS | K18 |
| VSS | J18 |
| VSS | H18 |
| VSS | G18 |
| VSS | E18 |
| VSS | C18 |

| Signal | Ball # |
|---|---|
| VSS | A18 |
| VSS | AA17 |
| VSS | Y17 |
| VSS | W17 |
| VSS | V17 |
| VSS | R17 |
| VSS | P17 |
| VSS | N17 |
| VSS | M17 |
| VSS | G17 |
| VSS | F17 |
| VSS | D17 |
| VSS | B17 |
| VSS | AG16 |
| VSS | AF16 |
| VSS | AA16 |
| VSS | Y16 |
| VSS | W16 |
| VSS | V16 |
| VSS | R16 |
| VSS | P16 |
| VSS | N16 |
| VTT | H6 |
| VTT | K5 |
| VTT | AL14 |
| VTT | AG14 |
| VTT | AG12 |
| VTT | AJ11 |
| VTT | AF11 |
| VTT | AG10 |
| VTT | AM9 |
| VTT | AJ9 |
| VTT | AF9 |
| VTT | AK7 |
| VTT | AD7 |
| VTT | AB7 |
| VTT | Y7 |
| VTT | V7 |
| VTT | T7 |
| VTT | M7 |
| VTT | AM6 |
| VTT | AF6 |
| VTT | Y6 |
| VTT | P6 |
| VTT | AH5 |
| VTT | K7 |
| VTT | AB5 |
| VTT | T5 |
| VTT | AK4 |
| VTT | AD4 |
| VTT | V4 |
| VTT | M4 |
| VTT | F4 |
| VTT | AM3 |
| VTT | AF3 |
| VTT | Y3 |
| VTT | P3 |
| VTT | H3 |
| VTT | B3 |
| VTT | AH2 |
| VTT | AB2 |
| VTT | T2 |
| VTT | K2 |
| VTT | D2 |
| VTT | AK1 |
| VTT | AD1 |
| VTT | V1 |
| VTT | M1 |
| VTT | F1 |
| WBF# | AM18 |

## 7.2 Intel® 82860 MCH Package Information

Figure 13 shows the package dimensions for the MCH.

### Figure 13. MCH Package Dimensions



Notes:
1. Substrate thicknes and package overall height are thicker than standard 492-L PBGA
2. Dimension is measured at the maximum solder ball diameter, parallel to primary Dataum -C-
3. Primary Dataum -C- and seating plane are determined by the spherical crowns of the solder balls
4. All dimensions are in millimeters, unless otherwise specified
5. All dimensions and tolerances conform to ANSI Y14.5M; 1982

pkg_olga_1012

## 7.3 Chipset Interface Trace Length Compensation

In this section, detailed information about the internal component package trace length is provided to enable trace length compensation. Trace length compensation is very important to maximize design flexibility. These lengths must be considered when matching trace lengths as described in the *Intel® 860 Chipset Platform Design Guide*. Note that these lengths are normalized to 0 with the longest trace on the package. They do not represent the actual lengths from pad to ball.

The data given can be renormalized to start routing from a different ball. If a different signal (other than longest trace) is used for nominalization, use the following equation:

$$\text{New } \Delta L_{pkg}' = \Delta L_{pkg} - \Delta L_{Ref}$$

- $\Delta L_{Ref}$ is the reference signal used for nominalization

Table 31 shows an example where signal MEMORY1 trace length is used for nominalization.

**Table 31. Example Nominalization Table**

| | $\Delta L_{Pkg}$ (mils) | New $\Delta L_{Pkg}$ (mils) |
|---|---|---|
| MEMORY1 | 102.756 | 0.000 |
| MEMORY2 | 118.897 | 16.141 |
| MEMORY3 | 130.315 | 27.559 |
| MEMORY4 | 152.364 | 49.608 |
| MEMORYN | 175.984 | 73.228 |

## 7.3.1 MCH RSL Trace Length Compensation

Depending on the memory configuration, different trace length compensation equations are used to determine the $\Delta L_{PCB}$ for RSL signals. The following equations are for a RIMM only solution and Intel MRH-R for the expansion channel.

To calculate the $\Delta L_{PCB}$ for RSL signals from the MCH to RIMMs, use the following formula.

$$\Delta L_{PCB} = (\Delta L_{pkg\_MCH} * V_{pkg\_MCH}) / V_{PCB}$$

- ΔLPCB is the nominal Δ PCB trace length to be added on the PCB
- ΔLPkg_MCH is the nominal Δ package trace length of the MCH
- Vpkg_MCH is the package trace velocity of the MCH, and its the nominal value is 167.64 ps/in (6.6 ps/mm)
- VPCB is the PCB trace velocity

To calculate the $\Delta L_{PCB}$ for RSL signals from the MCH to Intel MRH-R, first normalize the RSL signals to one signal. Then, use the following formula to calculate $\Delta L_{PCB}$.

$$\Delta L_{PCB} = (\Delta L_{pkg\_MCH} * V_{pkg\_MCH} + \Delta L_{pkg\_MRH\text{-}R} * V_{pkg\_MRH\text{-}R}) / V_{PCB}$$

- ΔLPCB is the nominal Δ PCB trace length to be added on the PCB
- ΔLPkg_MCH is the nominal Δ package trace length of the
- Vpkg_MCH is the package trace velocity of the MCH, and its the nominal value is 167.64 ps/in (6.6 ps/mm)
- ΔLPkg_MRH_R is the nominal Δ package trace length of the Intel MRH-R
- Vpkg_MRH-R is the package trace velocity, and its the nominal value is 180 ps/in (7.09 ps/mm)
- VPCB is the PCB trace velocity

### 7.3.1.1 MCH RSL Normalized Trace Length Data

The MCH package trace length information for Rambus Channel A and Rambus Channel B is listed in Table 32.

**Table 32. MCH $\Delta L_{Pkg}$ Data for Rambus* Channel A and Rambus Channel B**

| Channel A $\Delta L_{Pkg}$ Normalized to DQA_A7 | | |
|---|---|---|
| **Signal** | **Ball No.** | **$\Delta L_{Pkg}$ (mils)** |
| CFM_A | Y31 | 76.535 |
| CFM_A# | Y32 | 5.394 |
| CTM_A | AA32 | 13.307 |
| CTM_A# | AA31 | 69.567 |
| DQA_A0 | AC29 | 118.189 |
| DQA_A1 | AC27 | 135.512 |
| DQA_A2 | AC31 | 12.874 |
| DQA_A3 | AB28 | 162.244 |
| DQA_A4 | AD30 | 31.496 |
| DQA_A5 | AD28 | 158.346 |
| DQA_A6 | AE31 | 6.575 |
| DQA_A7 | AE27 | 0.000 |
| DQA_A8 | AE29 | 46.732 |
| DQB_A0 | T27 | 316.102 |
| DQB_A1 | P31 | 69.331 |
| DQB_A2 | P29 | 208.386 |
| DQB_A3 | M29 | 129.882 |
| DQB_A4 | P27 | 330.276 |
| DQB_A5 | M31 | 20.866 |
| DQB_A6 | N30 | 154.331 |
| DQB_A7 | M27 | 243.346 |
| DQB_A8 | N28 | 218.780 |
| EXP_A0 | U30 | 157.638 |
| EXP_A1 | V29 | 208.780 |
| RQ_A0 | R32 | 51.890 |
| RQ_A1 | T29 | 196.811 |
| RQ_A2 | T31 | 105.039 |
| RQ_A3 | U28 | 249.213 |
| RQ_A4 | U32 | 58.937 |
| RQ_A5 | V27 | 295.433 |
| RQ_A6 | V31 | 89.370 |
| RQ_A7 | W28 | 230.079 |

| Channel B $\Delta L_{Pkg}$ Normalized to DQA_B4 | | |
|---|---|---|
| **Signal** | **Ball No.** | **$\Delta L_{Pkg}$ (mils)** |
| CFM_B | B20 | 104.291 |
| CFM_B# | A20 | 32.874 |
| CTM_B | A21 | 41.024 |
| CTM_B# | B21 | 98.031 |
| DQA_B0 | D23 | 146.654 |
| DQA_B1 | F23 | 156.417 |
| DQA_B2 | B23 | 41.417 |
| DQA_B3 | E22 | 213.898 |
| DQA_B4 | A24 | 0.000 |
| DQA_B5 | E24 | 195.157 |
| DQA_B6 | C24 | 105.236 |
| DQA_B7 | D25 | 90.906 |
| DQA_B8 | B25 | 21.890 |
| DQB_B0 | F16 | 331.417 |
| DQB_B1 | B14 | 105.197 |
| DQB_B2 | D14 | 212.677 |
| DQB_B3 | D12 | 137.677 |
| DQB_B4 | F14 | 337.047 |
| DQB_B5 | B12 | 60.276 |
| DQB_B6 | C13 | 155.669 |
| DQB_B7 | F12 | 258.661 |
| DQB_B8 | E13 | 228.268 |
| EXP_B0 | C17 | 186.102 |
| EXP_B1 | D18 | 237.283 |
| RQ_B0 | A15 | 79.409 |
| RQ_B1 | D16 | 231.024 |
| RQ_B2 | B16 | 139.016 |
| RQ_B3 | E17 | 281.693 |
| RQ_B4 | A17 | 82.362 |
| RQ_B5 | F18 | 324.055 |
| RQ_B6 | B18 | 120.394 |
| RQ_B7 | E19 | 258.504 |

## 7.3.2 MCH System Bus Signal Normalized Trace Length Data

To calculate the $\Delta L_{PCB}$ for the system bus, first normal the processor and MCH to the same signal within a group. Then follow the trace length equations documented in the *Intel® Xeon™ Processor and Intel® 860 Chipset Platform Design Guide*. The MCH system bus interface normalized data per group is provided in the following tables.

### Table 33. MCH System Bus Signal Normalized Trace Length Data per Group

| HADSTB0# Group<br>Normalized to HA9# | | |
|---|---|---|
| **Signal** | **Ball No.** | **$\Delta L_{Pkg}$ (mils)** |
| HADSTB0# | AJ2 | 38.031 |
| AP0# | AK11 | 254.803 |
| HA3# | AE6 | 364.252 |
| HA4# | AF4 | 223.583 |
| HA5# | AH1 | 38.780 |
| HA6# | AH3 | 115.827 |
| HA7# | AG2 | 136.220 |
| HA8# | AH4 | 191.417 |
| HA9# | AJ1 | 0.000 |
| HA10# | AL3 | 36.772 |
| HA11# | AJ4 | 138.189 |
| HA12# | AK3 | 94.331 |
| HA13# | AM4 | 19.606 |
| HA14# | AL4 | 50.551 |
| HA15# | AK2 | 42.717 |
| HA16# | AL2 | 1.850 |
| HREQ0# | AG3 | 123.307 |
| HREQ1# | AF2 | 139.055 |
| HREQ2# | AF1 | 72.087 |
| HREQ3# | AF5 | 291.260 |
| HREQ4# | AE4 | 273.701 |

| HADSTB1# Group<br>Normalized to HA32# | | |
|---|---|---|
| **Signal** | **Ball No.** | **$\Delta L_{Pkg}$ (mils)** |
| HADSTB1# | AJ7 | 263.031 |
| AP1# | AH11 | 402.874 |
| HA17# | AF7 | 395.157 |
| HA18# | AG9 | 437.283 |
| HA19# | AH9 | 306.693 |
| HA20# | AM5 | 71.693 |
| HA21# | AJ10 | 349.961 |
| HA22# | AK6 | 200.039 |
| HA23# | AJ8 | 324.567 |
| HA24# | AH7 | 349.252 |
| HA25# | AJ5 | 235.276 |
| HA26# | AM7 | 40.906 |
| HA27# | AM8 | 100.709 |
| HA28# | AH6 | 303.543 |
| HA29# | AH10 | 380.984 |
| HA30# | AK5 | 149.016 |
| HA31# | AK9 | 265.157 |
| HA32# | AL9 | 0.000 |
| HA33# | AK8 | 249.094 |
| HA34# | AL6 | 103.189 |
| HA35# | AL7 | 163.701 |

| HDSTBx3# Group<br>Normalized to HD62# | | |
|---|---|---|
| **Signal** | **Ball No.** | **$\Delta L_{Pkg}$ (mils)** |
| HDSTBP3# | AA2 | 269.764 |
| HDSTBN3# | AA3 | 353.228 |
| HD48# | Y2 | 309.921 |
| HD49# | AA6 | 408.071 |
| HD50# | AA5 | 436.654 |
| HD51# | AB6 | 518.031 |
| HD52# | Y1 | 228.228 |
| HD53# | AB3 | 122.441 |
| HD54# | AB1 | 194.646 |
| HD55# | AB4 | 388.228 |
| HD56# | AC5 | 439.370 |
| HD57# | AD2 | 190.669 |
| HD58# | AD3 | 235.079 |
| HD59# | AE3 | 224.843 |
| HD60# | AD5 | 332.953 |
| HD61# | AC4 | 359.882 |
| HD62# | AC2 | 0.000 |
| HD63# | AE1 | 160.157 |
| DBI3# | AC1 | 104.724 |
| | | |
| | | |

### Table 34. MCH System Bus Signal Normalized Trace Length Data per Group (continued)

| HDSTBx2# Group Normalized to HD42# | | |
|---|---|---|
| Signal | Ball No. | $\Delta L_{Pkg}$ (mils) |
| HDSTBP2# | T1 | 15.354 |
| HDSTBN2# | T3 | 151.181 |
| HD32# | T4 | 215.630 |
| HD33# | R3 | 151.142 |
| HD34# | R5 | 246.417 |
| HD35# | U2 | 81.181 |
| HD36# | R6 | 281.850 |
| HD37# | U5 | 242.205 |
| HD38# | U1 | 14.291 |
| HD39# | U4 | 227.283 |
| HD40# | V5 | 261.772 |
| HD41# | V3 | 115.709 |
| HD42# | W1 | 0.000 |
| HD43# | V6 | 292.559 |
| HD44# | W4 | 196.339 |
| HD45# | W3 | 128.937 |
| HD46# | Y5 | 203.701 |
| HD47# | Y4 | 139.449 |
| DBI2# | V2 | 79.213 |

| HDSTBx1# Group Normalized to HDSTBN1# | | |
|---|---|---|
| Signal | Ball No. | $\Delta L_{Pkg}$ (mils) |
| HDSTBP1# | N3 | 26.024 |
| HDSTBN1# | N4 | 0.000 |
| HD16# | K6 | 596.378 |
| HD17# | J5 | 476.575 |
| HD18# | K4 | 499.843 |
| HD19# | J3 | 406.575 |
| HD20# | K3 | 425.669 |
| HD21# | J2 | 388.701 |
| HD22# | L4 | 515.276 |
| HD23# | L5 | 540.906 |
| HD24# | N1 | 367.953 |
| HD25# | M3 | 475.591 |
| HD26# | L2 | 390.157 |
| HD27# | M5 | 580.748 |
| HD28# | M2 | 378.543 |
| HD29# | L1 | 325.433 |
| HD30# | N6 | 621.339 |
| HD31# | P2 | 453.110 |
| DBI1# | K1 | 351.260 |

| HDSTBx0# Group Normalized to HD2# | | |
|---|---|---|
| Signal | Ball No. | $\Delta L_{Pkg}$ (mils) |
| HDSTBP0# | H1 | 10.236 |
| HDSTBN0# | G3 | 244.055 |
| HD0# | D3 | 169.803 |
| HD1# | B2 | 66.496 |
| HD2# | E1 | 0.000 |
| HD3# | E4 | 228.346 |
| HD4# | C2 | 96.181 |
| HD5# | F2 | 179.370 |
| HD6# | E2 | 132.126 |
| HD7# | D4 | 200.512 |
| HD8# | D1 | 86.850 |
| HD9# | F3 | 224.449 |
| HD10# | G4 | 278.701 |
| HD11# | G1 | 90.315 |
| HD12# | H2 | 173.543 |
| HD13# | G6 | 394.843 |
| HD14# | H4 | 324.843 |
| HD15# | H5 | 358.622 |
| DBI0# | A3 | 31.024 |

## 7.3.3 MCH 16-Bit Hub Interface Normalized Trace Length

To calculate the $\Delta L_{PCB}$ for 16-bit hub interfaces to the Intel P64H, use the following formula.

$$\Delta L_{PCB} = (\Delta L_{pkg\_MCH} * V_{pkg\_MCH}) / V_{PCB}$$

- ΔLPCB is the nominal Δ PCB trace length to be added on the PCB
- ΔLPkg_MCH is the nominal Δ package trace length of the MCH
- Vpkg_MCH is the package trace velocity of the MCH, and its the nominal value is 167.64 ps/in (6.6 ps/mm)
- VPCB is the PCB trace velocity
- Reference the *Intel® Xeon™ Processor and Intel® 860 Chipset Platform Design Guide* for information regarding how and when to use this data contained in the following sections for the 16-bit Hub Interface_B–C.

### 7.3.3.1 MCH 16-Bit Hub Interface_B Normalized Trace Length Data

Below is the MCH package trace length information for the 16-bit Hub Interface_B.

**Table 35. MCH 16-bit Hub Interface_B Signal Normalized Trace Length Data**

| Hub Interface_B Normalized to HL_B6 | | | | | |
|---|---|---|---|---|---|
| **Signal** | **Ball No.** | **$\Delta L_{Pkg}$ (mils)** | **Signal** | **Ball No.** | **$\Delta L_{Pkg}$ (mils)** |
| HL_B0 | B30 | 264.094 | HL_B2 | A30 | 229.567 |
| HL_B1 | D30 | 326.457 | HL_B3 | C31 | 270.433 |
| HL_B10 | F29 | 283.543 | HL_B4 | C32 | 199.685 |
| HL_B11 | J28 | 559.882 | HL_B5 | F31 | 190.157 |
| HL_B12 | H30 | 424.528 | HL_B6 | F32 | 0.000 |
| HL_B13 | H29 | 479.567 | HL_B7 | E30 | 301.772 |
| HL_B14 | J32 | 347.362 | HL_B8 | J31 | 419.488 |
| HL_B15 | J29 | 540.709 | HL_B9 | F28 | 449.803 |
| HL_B16 | G30 | 428.189 | HLB_STB0 | E32 | 239.567 |
| HL_B17 | G27 | 588.189 | HLB_STB0# | D31 | 312.165 |
| HL_B18 | H27 | 486.732 | HLB_STB1 | H32 | 302.205 |
| HL_B19 | G28 | 509.449 | HLB_STB1# | G31 | 372.244 |

### 7.3.3.2 MCH 16-Bit Hub Interface_C Normalized Trace Length Data

Below is the MCH package trace length information for the 16-bit Hub Interface_C.

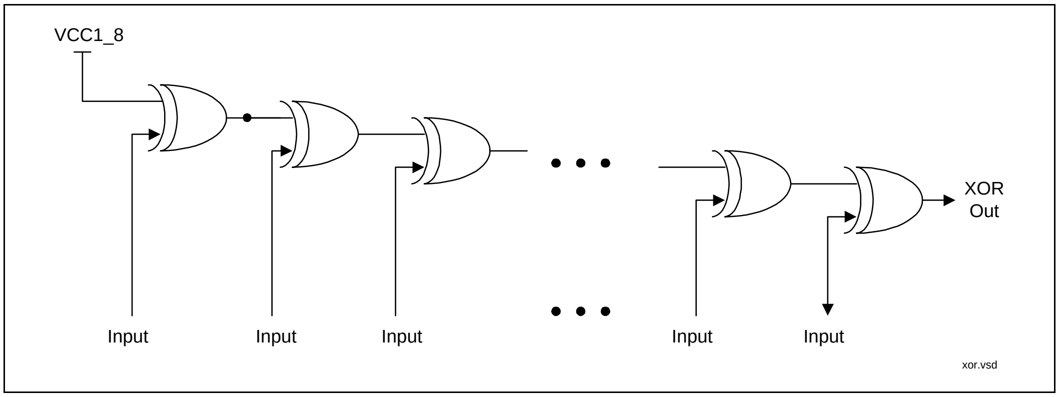
**Table 36. MCH 16-Bit Hub Interface_C Signal Normalized Trace Length Data**

| Hub Interface_C Normalized to HLC_STB0# | | | | | |
|---|---|---|---|---|---|
| **Signal** | **Ball No.** | **$\Delta L_{Pkg}$ (mils)** | **Signal** | **Ball No.** | **$\Delta L_{Pkg}$ (mils)** |
| HL_C0 | F5 | 440.906 | HL_C2 | E6 | 431.575 |
| HL_C1 | E5 | 359.331 | HL_C3 | C4 | 167.205 |
| HL_C10 | B8 | 291.260 | HL_C4 | A5 | 152.795 |
| HL_C11 | F10 | 552.244 | HL_C5 | A4 | 138.858 |
| HL_C12 | B9 | 323.543 | HL_C6 | B6 | 229.331 |
| HL_C13 | F9 | 527.953 | HL_C7 | B5 | 154.173 |
| HL_C14 | E8 | 446.890 | HL_C8 | E9 | 464.567 |
| HL_C15 | D10 | 472.283 | HL_C9 | F8 | 519.764 |
| HL_C16 | A8 | 251.535 | HLC_STB0 | C6 | 246.654 |
| HL_C17 | D7 | 222.992 | HLC_STB0# | D5 | 0.000 |
| HL_C18 | A7 | 220.827 | HLC_STB1 | C9 | 385.748 |
| HL_C19 | C7 | 282.362 | HLC_STB1# | D8 | 426.654 |

# 8 Testability

In the MCH, the testability for Automated Test Equipment (ATE) board level testing has been implemented as an XOR chain. An XOR-Tree is a chain of XOR gates, each with one Input pin connected to it.

**Figure 14. XOR-Tree Chain (High Level View)**



The algorithm used for in-circuit test is as follows:

- Drive all Input pins to their initial logic level as indicated in the tables below. Observe the output.
- Toggle pins one at a time starting from the first pin continuing to the last pin from its initial logic level to its opposite level. Observe that the output changes with each pin toggle.

Certain pin pairs must be toggled together. These pin pairs are: AD_STB0 and AD_STB0#, AD_STB1 and AD_STB1#, and SB_STB and SB_STB#. For example AD_STB0 has an initial state of 1 and AD_STB0# has an initial state of 0 when reaching AD_STB0, both AD_STB0 and AD_STB0# should be toggled together (AD_STB0: 1 -> 0 and AD_STB0#: 0 -> 1). When these pins are toggled together, the output will not change.

It is important to use the initial input states found in the following tables.

The following pins must be connected to their proper circuit in order for the XOR chain to work:

- HLREF_A
- HLREF_B
- HLREF_C
- HLSWNG_B
- HLSWNG_C
- HL1_8
- All inputs must be driven to CMOS voltage levels (0-1.8 V) regardless of signal type.

## 8.1 XOR Test Mode Initialization

The MCH uses a single pin (TESTIN#) to activate the XOR test mode.

- 32,771 clocks (66IN) after the deassertion of PCI reset (RSTIN#), drive TESTIN# from 1 to 0.
- Start to serially load the bits: 8010 0000 1000 0000 0000 3F0C 003C 0001h into the TESTIN# pin.
- Once in the XOR test mode, any high-to-low transition on TESTIN# will cause the serial test mode entry state machine to be activated. Care should be taken to keep TESTIN# stable after activating the XOR test mode.

## 8.2 XOR Chains

### Table 37. XOR Chain 1

| Chain 1 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| HA13# | AM4 | 1 | Input | 1 |
| HA14# | AL4 | 2 | Input | 1 |
| HA10# | AL3 | 3 | Input | 1 |
| HA11# | AJ4 | 4 | Input | 1 |
| HA12# | AK3 | 5 | Input | 1 |
| HA16# | AL2 | 6 | Input | 1 |
| HA8# | AH4 | 7 | Input | 1 |
| HA3# | AE6 | 8 | Input | 1 |
| HA15# | AK2 | 9 | Input | 1 |
| HREQ3# | AF5 | 10 | Input | 1 |
| HADSTB0# | AJ2 | 11 | Input | 1 |
| HA4# | AF4 | 12 | Input | 1 |
| HA9# | AJ1 | 13 | Input | 1 |
| HA6# | AH3 | 14 | Input | 1 |
| HREQ4# | AE4 | 15 | Input | 1 |
| HREQ0# | AG3 | 16 | Input | 1 |
| HA7# | AG2 | 17 | Input | 1 |
| HA5# | AH1 | 18 | Input | 1 |
| HREQ1# | AF2 | 19 | Input | 1 |
| HREQ2# | AF1 | 20 | Input | 1 |
| HD56# | AC5 | 21 | Input | 1 |
| HD59# | AE3 | 22 | Input | 1 |

| Chain 1 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| HD60# | AD5 | 23 | Input | 1 |
| HD61# | AC4 | 24 | Input | 1 |
| HD51# | AB6 | 25 | Input | 1 |
| HD49# | AA6 | 26 | Input | 1 |
| HD63# | AE1 | 27 | Input | 1 |
| HD57# | AD2 | 28 | Input | 1 |
| HD55# | AB4 | 29 | Input | 1 |
| HD58# | AD3 | 30 | Input | 1 |
| HDSTBN3# | AA3 | 31 | Input | 1 |
| HDSTBP3# | AA2 | 32 | Input | 1 |
| HD53# | AB3 | 33 | Input | 1 |
| DBI3# | AC1 | 34 | Input | 1 |
| HD50# | AA5 | 35 | Input | 1 |
| HD54# | AB1 | 36 | Input | 1 |
| HD62# | AC2 | 37 | Input | 1 |
| HD48# | Y2 | 38 | Input | 1 |
| HD52# | Y1 | 39 | Input | 1 |
| HD46# | Y5 | 40 | Input | 1 |
| HD47# | Y4 | 41 | Input | 1 |
| HD44# | W4 | 42 | Input | 1 |
| HD45# | W3 | 43 | Input | 1 |
| HD43# | V6 | 44 | Input | 1 |
| HD42# | W1 | 45 | Input | 1 |
| HD40# | V5 | 46 | Input | 1 |
| DBI_2# | V2 | 47 | Input | 1 |
| HD37# | U5 | 48 | Input | 1 |
| HD41# | V3 | 49 | Input | 1 |
| HD39# | U4 | 50 | Input | 1 |
| HD35# | U2 | 51 | Input | 1 |
| HD38# | U1 | 52 | Input | 1 |
| HDSTBN2# | T3 | 53 | Input | 1 |
| HDSTBP2# | T1 | 54 | Input | 1 |
| SBA0 | AL18 | 55 | Output | N/A |

### Table 38. XOR Chain 2

| Chain 2 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| HD32# | T4 | 1 | Input | 1 |
| HD33# | R3 | 2 | Input | 1 |
| HD34# | R5 | 3 | Input | 1 |
| HD36# | R6 | 4 | Input | 1 |
| HD31# | P2 | 5 | Input | 1 |
| HD24# | N1 | 6 | Input | 1 |
| HRCOMP1 | P7 | 7 | Input | 1 |
| HD28# | M2 | 8 | Input | 1 |
| HD29# | L1 | 9 | Input | 1 |
| HD25# | M3 | 10 | Input | 1 |
| HD26# | L2 | 11 | Input | 1 |
| DBI1# | K1 | 12 | Input | 1 |
| HD30# | N6 | 13 | Input | 1 |
| HD27# | M5 | 14 | Input | 1 |
| HDSTBN1# | N4 | 15 | Input | 1 |
| HDSTBP1# | N3 | 16 | Input | 1 |
| HD22# | L4 | 17 | Input | 1 |
| HD20# | K3 | 18 | Input | 1 |
| HD21# | J2 | 19 | Input | 1 |
| HD23# | L5 | 20 | Input | 1 |
| HD18# | K4 | 21 | Input | 1 |
| HD19# | J3 | 22 | Input | 1 |
| HD16# | K6 | 23 | Input | 1 |
| HD17# | J5 | 24 | Input | 1 |
| HD2# | E1 | 25 | Input | 1 |
| HD11# | G1 | 26 | Input | 1 |
| HD14# | H4 | 27 | Input | 1 |
| HD12# | H2 | 28 | Input | 1 |
| HD5# | F2 | 29 | Input | 1 |
| HD15# | H5 | 30 | Input | 1 |
| HD10# | G4 | 31 | Input | 1 |
| HDSTBN0# | G3 | 32 | Input | 1 |
| HDSTBP0# | H1 | 33 | Input | 1 |
| HD9# | F3 | 34 | Input | 1 |

| Chain 2 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| HD6# | E2 | 35 | Input | 1 |
| HD8# | D1 | 36 | Input | 1 |
| HD4# | C2 | 37 | Input | 1 |
| HD0# | D3 | 38 | Input | 1 |
| HD3# | E4 | 39 | Input | 1 |
| HD1# | B2 | 40 | Input | 1 |
| HD13# | G6 | 41 | Input | 1 |
| HD7# | D4 | 42 | Input | 1 |
| DBI0# | A3 | 43 | Input | 1 |
| CMD_B | A11 | 44 | Input | 1 |
| SIO_B | C11 | 45 | Input | 1 |
| SCK_B | A10 | 46 | Input | 1 |
| CHB_HCLKOUT | B10 | 47 | Input | 1 |
| CHB_RCLKOUT | C10 | 48 | Input | 1 |
| HL_A7 | A26 | 49 | Input | 1 |
| HL_A8 | E26 | 50 | Input | 1 |
| HL_A5 | B27 | 51 | Input | 1 |
| HL_A6 | A27 | 52 | Input | 1 |
| HL_A4 | A28 | 53 | Input | 1 |
| HL_A3 | B28 | 54 | Input | 1 |
| SBA1 | AJ18 | 55 | Output | N/A |

### Table 39. XOR Chain 3

| Chain 3 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| DP3# | R2 | 1 | Input | 1 |
| DP0# | P1 | 2 | Input | 1 |
| DP2# | P5 | 3 | Input | 1 |
| DP1# | P4 | 4 | Input | 1 |
| HL_C0 | F5 | 5 | Input | 1 |
| HL_C17 | D7 | 6 | Input | 1 |
| HL_C1 | E5 | 7 | Input | 1 |
| HL_C2 | E6 | 8 | Input | 1 |
| HL_C3 | C4 | 9 | Input | 1 |
| HL_C7 | B5 | 10 | Input | 1 |
| HLC_STB0# | D5 | 11 | Input | 1 |
| HLC_STB0 | C6 | 12 | Input | 1 |
| HLRCOMP_C | F7 | 13 | Input | 1 |
| HL_C5 | A4 | 14 | Input | 1 |
| HL_C4 | A5 | 15 | Input | 1 |
| HL_C6 | B6 | 16 | Input | 1 |
| HL_C19 | C7 | 17 | Input | 1 |
| HL_C9 | F8 | 18 | Input | 1 |
| HL_C13 | F9 | 19 | Input | 1 |
| HL_C14 | E8 | 20 | Input | 1 |
| HL_C8 | E9 | 21 | Input | 1 |
| HLC_STB1# | D8 | 22 | Input | 1 |
| HLC_STB1 | C9 | 23 | Input | 1 |
| HL_C18 | A7 | 24 | Input | 1 |
| HL_C11 | F10 | 25 | Input | 1 |
| HL_C16 | A8 | 26 | Input | 1 |
| HL_C15 | D10 | 27 | Input | 1 |
| HL_C10 | B8 | 28 | Input | 1 |
| HL_C12 | B9 | 29 | Input | 1 |
| HL_A11 | C26 | 30 | Input | 1 |
| HL_B2 | A30 | 31 | Input | 1 |
| HL_B0 | B30 | 32 | Input | 1 |
| HL_B1 | D30 | 33 | Input | 1 |

| Chain 3 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| HL_B6 | F32 | 34 | Input | 1 |
| HL_B3 | C31 | 35 | Input | 1 |
| HL_B17 | G27 | 36 | Input | 1 |
| HL_B4 | C32 | 37 | Input | 1 |
| HLB_STB0# | D31 | 38 | Input | 1 |
| HLB_STB0 | E32 | 39 | Input | 1 |
| SBA2 | AH18 | 40 | Output | N/A |

### Table 40. XOR Chain 4

| Chain 4 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| HL_A2 | A29 | 1 | Input | 1 |
| HLA_STB# | C27 | 2 | Input | 1 |
| HLA_STB | D28 | 3 | Input | 0 |
| HLRCOMP_A | E29 | 4 | Input | 1 |
| HL_A9 | E27 | 5 | Input | 1 |
| HL_A0 | D29 | 6 | Input | 1 |
| HL_A1 | C29 | 7 | Input | 1 |
| HL_A10 | F26 | 8 | Input | 1 |
| CMD_A | L32 | 9 | Input | 1 |
| SIO_A | L30 | 10 | Input | 1 |
| SCK_A | K32 | 11 | Input | 1 |
| CHA_RCLKOUT | K30 | 12 | Input | 1 |
| CHA_HCLKOUT | K31 | 13 | Input | 1 |
| G_AD1 | AH32 | 14 | Input | 1 |
| G_AD5 | AK31 | 15 | Input | 1 |
| G_AD0 | AG32 | 16 | Input | 1 |
| G_AD3 | AF32 | 17 | Input | 1 |
| G_AD10 | AG29 | 18 | Input | 1 |
| G_AD9 | AH29 | 19 | Input | 1 |
| G_AD14 | AJ28 | 20 | Input | 1 |
| G_AD2 | AK32 | 21 | Input | 1 |
| G_AD15 | AM28 | 22 | Input | 1 |
| AD_STB0 | AK29 | 23 | Input | 1 |
| AD_STB0# | AJ30 | 24 | Input | 0 |
| G_AD4 | AJ31 | 25 | Input | 1 |

| Chain 4 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| G_C/BE0# | AL30 | 26 | Input | 1 |
| G_AD11 | AL29 | 27 | Input | 1 |
| G_AD7 | AF30 | 28 | Input | 1 |
| G_AD6 | AL31 | 29 | Input | 1 |
| G_AD8 | AM30 | 30 | Input | 1 |
| G_AD12 | AF28 | 31 | Input | 1 |
| G_C/BE1# | AM27 | 32 | Input | 1 |
| G_AD13 | AG28 | 33 | Input | 1 |
| G_DEVSEL# | AH26 | 34 | Input | 1 |
| G_FRAME# | AG26 | 35 | Input | 1 |
| G_IRDY# | AK26 | 36 | Input | 1 |
| G_TRDY# | AH27 | 37 | Input | 1 |
| G_STOP# | AJ27 | 38 | Input | 1 |
| G_PAR | AF27 | 39 | Input | 1 |
| G_C/BE2# | AL26 | 40 | Input | 1 |
| G_AD20 | AJ24 | 41 | Input | 1 |
| G_AD16 | AG25 | 42 | Input | 1 |
| G_AD21 | AK23 | 43 | Input | 1 |
| G_AD17 | AJ25 | 44 | Input | 1 |
| G_AD18 | AM25 | 45 | Input | 1 |
| G_AD19 | AH24 | 46 | Input | 1 |
| G_AD23 | AG22 | 47 | Input | 1 |
| G_AD22 | AM24 | 48 | Input | 1 |
| AD_STB1 | AL21 | 49 | Input | 1 |
| AD_STB1# | AK22 | 50 | Input | 0 |
| GRCOMP | AF17 | 51 | Input | 1 |
| G_C/BE3# | AL23 | 52 | Input | 1 |
| G_AD24 | AM22 | 53 | Input | 1 |
| G_AD25 | AF21 | 54 | Input | 1 |
| SBA3 | AG18 | 55 | Output | N/A |

### Table 41. XOR Chain 5

| Chain 5 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| G_AD27 | AJ21 | 1 | Input | 1 |
| G_AD26 | AH21 | 2 | Input | 1 |
| G_AD30 | AH20 | 3 | Input | 1 |
| G_AD29 | AG20 | 4 | Input | 1 |
| G_AD28 | AM21 | 5 | Input | 1 |
| G_AD31 | AK20 | 6 | Input | 1 |
| SB_STB | AM19 | 7 | Input | 1 |
| SB_STB# | AK19 | 8 | Input | 0 |
| ST0 | AL17 | 9 | Input | 1 |
| ST1 | AK17 | 10 | Input | 1 |
| ST2 | AJ17 | 11 | Input | 1 |
| RBF# | AH17 | 12 | Input | 1 |
| G_GNT# | AM17 | 13 | Input | 1 |
| PIPE# | AG17 | 14 | Input | 1 |
| WBF# | AM18 | 15 | Input | 1 |
| G_REQ# | AL16 | 16 | Input | 1 |
| HLA_ENH# | AH15 | 17 | Input | 1 |
| BRO# | AM15 | 18 | Input | 1 |
| HTRDY# | AM14 | 19 | Input | 1 |
| DRDY# | AM13 | 20 | Input | 1 |
| HIT# | AL13 | 21 | Input | 1 |
| HITM# | AK13 | 22 | Input | 1 |
| BNR# | AM12 | 23 | Input | 1 |
| HLOCK# | AH13 | 24 | Input | 1 |
| BPRI# | AK12 | 25 | Input | 1 |
| RS0# | AM11 | 26 | Input | 1 |
| CPURST# | AL11 | 27 | Input | 1 |
| DBSY# | AJ12 | 28 | Input | 1 |
| RS2# | AM10 | 29 | Input | 1 |
| HRCOMP0 | AF13 | 30 | Input | 1 |
| RS1# | AL10 | 31 | Input | 1 |
| DEFER# | AH12 | 32 | Input | 1 |
| ADS# | AG11 | 33 | Input | 1 |
| HA21# | AJ10 | 34 | Input | 1 |

| Chain 5 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| HA31# | AK9 | 35 | Input | 1 |
| HA29# | AH10 | 36 | Input | 1 |
| HA19# | AH9 | 37 | Input | 1 |
| HA27# | AM8 | 38 | Input | 1 |
| HA26# | AM7 | 39 | Input | 1 |
| HA23# | AJ8 | 40 | Input | 1 |
| HADSTB1# | AJ7 | 41 | Input | 1 |
| HA20# | AM5 | 42 | Input | 1 |
| HA18# | AG9 | 43 | Input | 1 |
| HA22# | AK6 | 44 | Input | 1 |
| HA30# | AK5 | 45 | Input | 1 |
| HA24# | AH7 | 46 | Input | 1 |
| HA25# | AJ5 | 47 | Input | 1 |
| HA28# | AH6 | 48 | Input | 1 |
| HA17# | AF7 | 49 | Input | 1 |
| SBA4 | AJ19 | 50 | Output | N/A |

### Table 42. XOR Chain 6

| Chain 6 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| HLRCOMP_B | K27 | 1 | Input | 1 |
| HL_B19 | G28 | 2 | Input | 1 |
| HL_B18 | H27 | 3 | Input | 1 |
| HL_B5 | F31 | 4 | Input | 1 |
| HL_B7 | E30 | 5 | Input | 1 |
| HL_B9 | F28 | 6 | Input | 1 |
| HL_B16 | G30 | 7 | Input | 1 |
| HL_B10 | F29 | 8 | Input | 1 |
| HL_B11 | J28 | 9 | Input | 1 |
| HLB_STB1# | G31 | 10 | Input | 1 |
| HLB_STB1 | H32 | 11 | Input | 1 |
| HL_B13 | H29 | 12 | Input | 1 |
| HL_B12 | H30 | 13 | Input | 1 |
| HL_B15 | J29 | 14 | Input | 1 |
| HL_B8 | J31 | 15 | Input | 1 |
| HL_B14 | J32 | 16 | Input | 1 |

| Chain 6 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| Rsvd | AG31 | 17 | Input | 1 |
| Rsvd | AH30 | 18 | Input | 1 |
| Rsvd | AF24 | 19 | Input | 1 |
| Rsvd | AK28 | 20 | Input | 1 |
| Rsvd | AL27 | 21 | Input | 1 |
| G_SERR# | AG27 | 22 | Input | 1 |
| Rsvd | AK25 | 23 | Input | 1 |
| Rsvd | AL24 | 24 | Input | 1 |
| Rsvd | AH23 | 25 | Input | 1 |
| Rsvd | AJ22 | 26 | Input | 1 |
| OVERT# | AJ16 | 27 | Input | 1 |
| BUSPARK | AH16 | 28 | Input | 1 |
| Rsvd | AL15 | 29 | Input | 1 |
| Rsvd | AK15 | 30 | Input | 1 |
| Rsvd | AJ15 | 31 | Input | 1 |
| RSP# | AK14 | 32 | Input | 1 |
| BERR# | AJ14 | 33 | Input | 1 |
| AP0# | AK11 | 34 | Input | 1 |
| AP1# | AH11 | 35 | Input | 1 |
| HA33# | AK8 | 36 | Input | 1 |
| HA35# | AL7 | 37 | Input | 1 |
| HA32# | AL9 | 38 | Input | 1 |
| HA34# | AL6 | 39 | Input | 1 |
| SBA5 | AG19 | 40 | Output | N/A |

### Table 43. XOR Chain 7

| Chain 7 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| DQA_A5 | AD28 | 1 | Input | 1 |
| DQA_A3 | AB28 | 2 | Input | 1 |
| DQA_A7 | AE27 | 3 | Input | 1 |
| DQA_A6 | AE31 | 4 | Input | 1 |
| DQA_A8 | AE29 | 5 | Input | 1 |
| DQA_A0 | AC29 | 6 | Input | 1 |
| DQA_A4 | AD30 | 7 | Input | 1 |
| DQA_A1 | AC27 | 8 | Input | 1 |
| DQA_A2 | AC31 | 9 | Input | 1 |
| CTM_A | AA32 | 10 | Input | 1 |
| CTM_A# | AA31 | 11 | Input | 1 |
| RQ_A7 | W28 | 12 | Input | 1 |
| EXP_A1 | V29 | 13 | Input | 1 |
| RQ_A6 | V31 | 14 | Input | 1 |
| RQ_A5 | V27 | 15 | Input | 1 |
| RQ_A4 | U32 | 16 | Input | 1 |
| EXP_A0 | U30 | 17 | Input | 1 |
| RQ_A3 | U28 | 18 | Input | 1 |
| RQ_A0 | R32 | 19 | Input | 1 |
| RQ_A2 | T31 | 20 | Input | 1 |
| RQ_A1 | T29 | 21 | Input | 1 |
| DQA_B5 | E24 | 22 | Input | 1 |
| DQA_B1 | F23 | 23 | Input | 1 |
| DQA_B0 | D23 | 24 | Input | 1 |
| DQA_B6 | C24 | 25 | Input | 1 |
| DQA_B2 | B23 | 26 | Input | 1 |
| DQA_B3 | E22 | 27 | Input | 1 |
| DQA_B4 | A24 | 28 | Input | 1 |
| DQA_B8 | B25 | 29 | Input | 1 |
| DQA_B7 | D25 | 30 | Input | 1 |
| SBA6 | AF19 | 31 | Output | N/A |

## Table 44. XOR Chain 8

| Chain 8 | Ball | Element # | Note | Initial Logic Level |
|---|---|---|---|---|
| DQB_A5 | M31 | 1 | Input | 1 |
| DQB_A3 | M29 | 2 | Input | 1 |
| DQB_A7 | M27 | 3 | Input | 1 |
| DQB_A6 | N30 | 4 | Input | 1 |
| DQB_A8 | N28 | 5 | Input | 1 |
| DQB_A0 | T27 | 6 | Input | 1 |
| DQB_A4 | P27 | 7 | Input | 1 |
| DQB_A1 | P31 | 8 | Input | 1 |
| DQB_A2 | P29 | 9 | Input | 1 |
| CTM_B | A21 | 10 | Input | 1 |
| CTM_B# | B21 | 11 | Input | 1 |
| RQ_B7 | E19 | 12 | Input | 1 |
| EXP_B1 | D18 | 13 | Input | 1 |
| RQ_B6 | B18 | 14 | Input | 1 |
| RQ_B5 | F18 | 15 | Input | 1 |
| RQ_B4 | A17 | 16 | Input | 1 |
| EXP_B0 | C17 | 17 | Input | 1 |
| RQ_B3 | E17 | 18 | Input | 1 |
| RQ_B0 | A15 | 19 | Input | 1 |
| RQ_B2 | B16 | 20 | Input | 1 |
| RQ_B1 | B14 | 21 | Input | 1 |
| DQB_B5 | B12 | 22 | Input | 1 |
| DQB_B1 | B14 | 23 | Input | 1 |
| DQB_B0 | F16 | 24 | Input | 1 |
| DQB_B6 | C13 | 25 | Input | 1 |
| DQB_B2 | D14 | 26 | Input | 1 |
| DQB_B3 | D12 | 27 | Input | 1 |
| DQB_B4 | F14 | 28 | Input | 1 |
| DQB_B8 | E13 | 29 | Input | 1 |
| DQB_B7 | F12 | 30 | Input | 1 |
| SBA7 | AL20 | 31 | Output | N/A |

RSTIN#, TESTIN#, CFM_A, CFM_A#, CHA_REF[0:1], CFM_B, CFM_B#, CHB_REF[0:1] are not part of any XOR chain. This is in addition to SBA[0:7].