

# INTEL 430MX PCISSET

## 82437MX MOBILE SYSTEM CONTROLLER (MTSC)

## AND 82438MX MOBILE DATA PATH (MTDP)

- **Supports the Pentium® Processor at iCOMP® Index 815/100 MHz, iCOMP Index 735/90 MHz, iCOMP Index 1000/120, and the 75 MHz Pentium Processor**
- **Integrated Second Level Cache Controller**
  - Direct Mapped Organization
  - Write-Back Cache Policy
  - Cacheless, 256 Kbytes, and 512 Kbytes
  - Standard, Burst and Pipelined Burst SRAMs
  - Cache Hit Read/Write Cycle Timings at 3-1-1-1 with Burst or Pipelined Burst SRAMs
  - Back-to-Back Read Cycles at 3-1-1-1-1-1-1 with Burst or Pipelined Burst SRAMs
  - Integrated Tag/Valid Status Bits for Cost Savings and Performance
  - Supports 3.3V SRAMs for Tag Address
- **Integrated DRAM Controller**
  - 64-Bit Data Path to Memory
  - 4 Mbytes to 128 Mbytes Main Memory
  - EDO/Hyper Page Mode DRAM (x-2-2-2 Reads) Provides Superior Cacheless Designs
- Standard Page Mode DRAMs
- 4 RAS Lines
- 4-Word Deep Buffer for 3-1-1-1 Posted Write Cycles
- Symmetrical and Asymmetrical DRAMs
- 3V or 5V DRAMs
- **Fully Synchronous 25/30/33 MHz PCI Bus Interface**
  - 100 MB/s Instant Access Enables Native Signal Processing on Pentium Processors
  - Synchronized CPU-to-PCI Interface for High Performance Graphics
  - PCI Bus Arbiter: MPIIX and Three PCI Bus Masters Supported
  - CPU-to-PCI Memory Write Posting with 4-Word Deep Buffers
  - Converts Back-to-Back Sequential CPU to PCI Memory Writes to PCI Burst Writes
  - PCI-to-DRAM Posting of 12 Dwords
  - PCI-to-DRAM up to 120 MB/s Bandwidth Utilizing Snoop Ahead Feature
- **NAND Tree for Board-Level ATE Testing**
- **208-Pin QFP for the 82437MX System Controller (MTSC); 100-Pin TQFP for Each 82438MX Data Path (MTDP)**

The Intel 430MX PCIsset consists of the 82437MX System Controller (MTSC), two 82438MX Data Paths (MTDP), and the 82371MX PCI I/O IDE Xcelerator (MPIIX). The PCIsset forms a Host-to-PCI bridge and provides the second level cache control and a full function 64-bit data path to main memory. The MTSC integrates the cache and main memory DRAM control functions and provides bus control for transfers between the CPU, cache, main memory, and the PCI Bus. The second level (L2) cache controller supports a write-back cache policy for cache sizes of 256 Kbytes and 512 Kbytes. Cacheless designs are also supported. The cache memory can be implemented with either standard, burst, or pipelined burst SRAMs. An external Tag RAM is used for the address tag and an internal Tag RAM for the cache line status bits. For the MTSC DRAM controller, four rows are supported for up to 128 Mbytes of main memory. The MTSC optimized PCI interface allows the CPU to sustain the highest possible bandwidth to the graphics frame buffer at all frequencies. Using the snoop ahead feature, the MTSC allows PCI masters to achieve full PCI bandwidth. The MTDPs provide the data paths between the CPU/cache, main memory, and PCI. For increased system performance, the MTDPs contain read prefetch and posted write buffers.

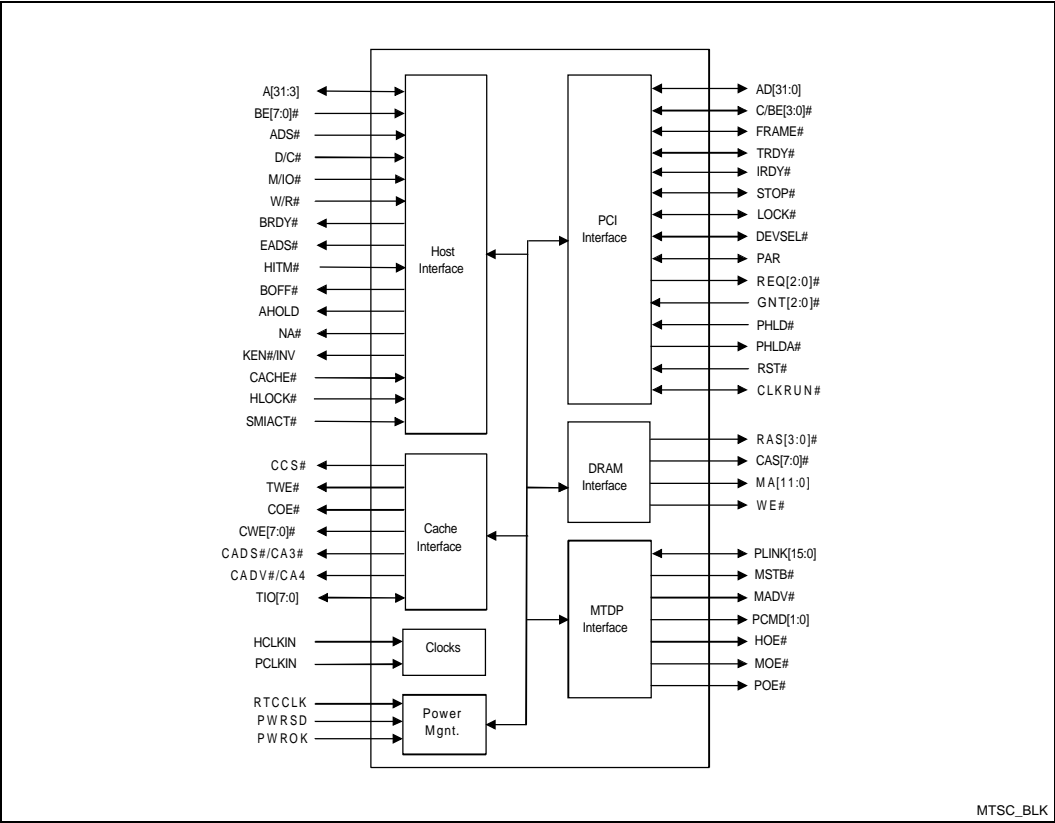
\*Other brands and names are the property of their respective owners.

Information in this document is provided in connection with Intel products. Intel assumes no liability whatsoever, including infringement of any patent or copyright, for sale and use of Intel products except as provided in Intel's Terms and Conditions of Sale for such products. Intel retains the right to make changes to those specifications at any time, without notice. Microcomputer Products may have minor variations to this specification known as errata.

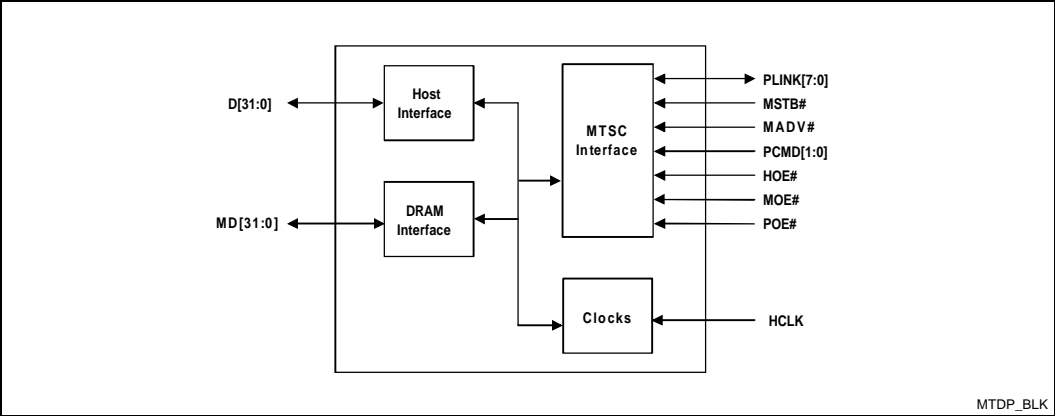
COPYRIGHT © INTEL CORPORATION, 1996

April 1996

Order Number: 290524-001



82437MX MTSC Block Diagram



82438MX MTD Block Diagram

# CONTENTS

<b>1.0. ARCHITECTURE OVERVIEW OF THE INTEL 430MX PCISSET.....</b>	<b>5</b>
<b>2.0 SIGNAL DESCRIPTION.....</b>	<b>7</b>
2.1. MTSC Signals.....	7
2.1.1. HOST INTERFACE (MTSC).....	7
2.1.2. DRAM INTERFACE (MTSC) .....	9
2.1.3. SECONDARY CACHE INTERFACE (MTSC) .....	9
2.1.4. PCI INTERFACE (MTSC) .....	10
2.1.5. MTPD INTERFACE (MTSC) .....	12
2.1.6. CLOCKS (MTSC) .....	12
2.1.7. POWER MANAGEMENT (MTSC) .....	13
2.2. MTPD Signals.....	13
2.2.1. DATA INTERFACE SIGNALS (MTPD) .....	13
2.2.2. MTSC INTERFACE SIGNALS (MTPD) .....	13
2.2.3. CLOCK SIGNAL (MTPD) .....	14
2.3. Strapping Options .....	14
<b>3.0. REGISTER DESCRIPTION.....</b>	<b>15</b>
3.1. Control Registers .....	15
3.1.1. CONFADD—CONFIGURATION ADDRESS REGISTER .....	15
3.1.2. CONFDATA—CONFIGURATION DATA REGISTER .....	16
3.2. PCI Configuration Registers .....	16
3.2.1. VID—VENDOR IDENTIFICATION REGISTER .....	18
3.2.2. DID—DEVICE IDENTIFICATION REGISTER .....	18
3.2.3. PCICMD—PCI COMMAND REGISTER .....	18
3.2.4. PCISTS—PCI STATUS REGISTER .....	19
3.2.5. RID—REVISION IDENTIFICATION REGISTER .....	20
3.2.6. CLASSC—CLASS CODE REGISTER .....	20
3.2.7. MLT—MASTER LATENCY TIMER REGISTER .....	20
3.2.8. BIST—BIST REGISTER .....	21
3.2.9. PCON—PCI CONTROL REGISTER .....	21
3.2.10. CC—CACHE CONTROL REGISTER .....	22
3.2.11. DRAMC—DRAM CONTROL REGISTER .....	23
3.2.12. DRAMT—DRAM TIMING REGISTER .....	24
3.2.13. PAM—PROGRAMMABLE ATTRIBUTE MAP REGISTERS (PAM[6:0]) .....	26
3.2.14. DRB—DRAM ROW BOUNDARY REGISTERS .....	28
3.2.15. DRT—DRAM ROW TYPE REGISTER .....	29
3.2.16. SMRAM—SYSTEM MANAGEMENT RAM CONTROL REGISTER .....	29

<b>4.0. FUNCTIONAL DESCRIPTION.....</b>	<b>31</b>
4.1. Host Interface .....	31
4.2. PCI Interface.....	31
4.3. Secondary Cache Interface .....	32
4.3.1. CLOCK LATENCIES .....	32
4.3.2. SNOOP CYCLES .....	33
4.3.3. SRAM POWER DOWN (DE-SELECT) MODE SEQUENCE .....	34
4.3.4. FLUSHING L2 CACHE .....	34
4.3.5. CACHE ORGANIZATION .....	34
4.4. DRAM Interface.....	38
4.4.1. DRAM ORGANIZATION .....	39
4.4.2. MAIN MEMORY ADDRESS MAP .....	39
4.4.3. DRAM ADDRESS TRANSLATION .....	39
4.4.4. DRAM PAGE MODE .....	41
4.4.5. EDO MODE .....	42
4.4.6. DRAM PERFORMANCE .....	43
4.4.7. DRAM REFRESH .....	45
4.4.8. SUSPEND REFRESH .....	46
4.4.9. SYSTEM MANAGEMENT RAM .....	46
4.5. 82438MX Data Path (MTDP) .....	46
4.6. PCI Bus Arbitration .....	47
4.6.1. PRIORITY SCHEME AND BUS GRANT .....	48
4.6.2. CPU POLICIES .....	48
4.7. Clocks and Reset.....	51
4.7.1. CLOCKS.....	51
4.7.2. RESET SEQUENCING .....	51
4.8. PCI Clock Control (CLKRUN#) .....	53
<b>5.0. PINOUT AND PACKAGE INFORMATION.....</b>	<b>54</b>
5.1. MTSC Pin Assignment .....	54
5.2. MTDP Pin Assignment .....	58
5.3. MTSC Package Characteristics .....	61
5.4. MTDP Package Characteristics .....	62
<b>6.0. TESTABILITY.....</b>	<b>63</b>
6.1. 82437MX MTSC TESTABILITY .....	63
6.1.1. TEST MODE OPERATION .....	63
6.1.2. TEST ISSUES.....	63
6.1.3. NAND TREE TIMING REQUIREMENTS .....	65
6.2. 82438MX MTDP TESTABILITY .....	68
6.2.1. NAND TREE TEST MODE OPERATION .....	68

## 1.0. ARCHITECTURE OVERVIEW OF THE INTEL 430MX PCISSET

The Intel 430MX PCISet (Figure 1) consists of the Mobile System Controller (MTSC), two Mobile Data Path (MTDP) units, and the 82371MB Mobile PCI I/O IDE Xcelerator (MPIIX). The MTSC and two MTDPs form a Host-to-PCI bridge. The MPIIX is a multi-function PCI device providing a PCI-to-Expansion I/O (ISA-like) bridge and a fast IDE interface. The MPIIX also provides power management and has a plug-n-play port.

The two MTDPs provide a 64-bit data path to the host and to main memory and provide a 16-bit data path (PLINK) between the MTSC and MTDP. PLINK provides the data path for CPU to PCI accesses and for PCI to main memory accesses. The MTSC and MTDP bus interfaces are designed for 3V and 5V busses. The MTSC/MTDP connects directly to the Pentium® processor 3V host bus; The MTSC/MTDP connects directly to 5V or 3V main memory DRAMs; and the MTSC connects directly to the 5V PCI bus.

### DRAM Interface

The DRAM interface is a 64-bit data path that supports both standard page mode and Extended Data Out (EDO) memory. With 60 ns EDO DRAMs, a 7-2-2-2 cycle time can be achieved at 66 MHz for reads and 3-1-1-1 for posted writes. The MTSC supports 4 Mbytes to 128 Mbytes with four RAS lines available and also supports symmetrical and asymmetrical addressing for 512K, 1-, 2-, and 4-Mbyte deep DRAMs. The MTSC supports CAS-before-RAS refresh (15.6  $\mu$ s and Extended-Refresh to 256  $\mu$ s) during normal and Suspend modes. In addition, Self-Refresh is supported for Suspend modes.

### Second Level Cache

The MTSC supports a write-back cache policy providing all necessary snoop functions and inquire cycles. The second level cache is direct mapped and supports both a 256-Kbyte or 512-Kbyte SRAM configuration using either burst or standard SRAMs. The burst 256-Kbyte configuration performance is 3-1-1-1 for read/write cycles; pipelined back-to-back reads can maintain a 3-1-1-1-1-1-1 transfer rate.

### MTDP

Two MTDPs create a 64-bit CPU and main memory data path. The MTDPs also interface to the MTSCs 16-bit PLINK inter-chip bus for PCI transactions. The combination of the 64-bit memory path and the 16-bit PLINK bus make the MTDPs a cost-effective solution, providing optimal CPU-to-main memory performance while maintaining a small package footprint (100 pins each).

### PCI Interface

The PCI interface is 2.0 compliant and supports up to 3 PCI bus masters in addition to the MPIIX bus master requests. While the MTSC and MTDP together provide the interface between PCI and main memory, only the MTSC connects to the PCI bus.

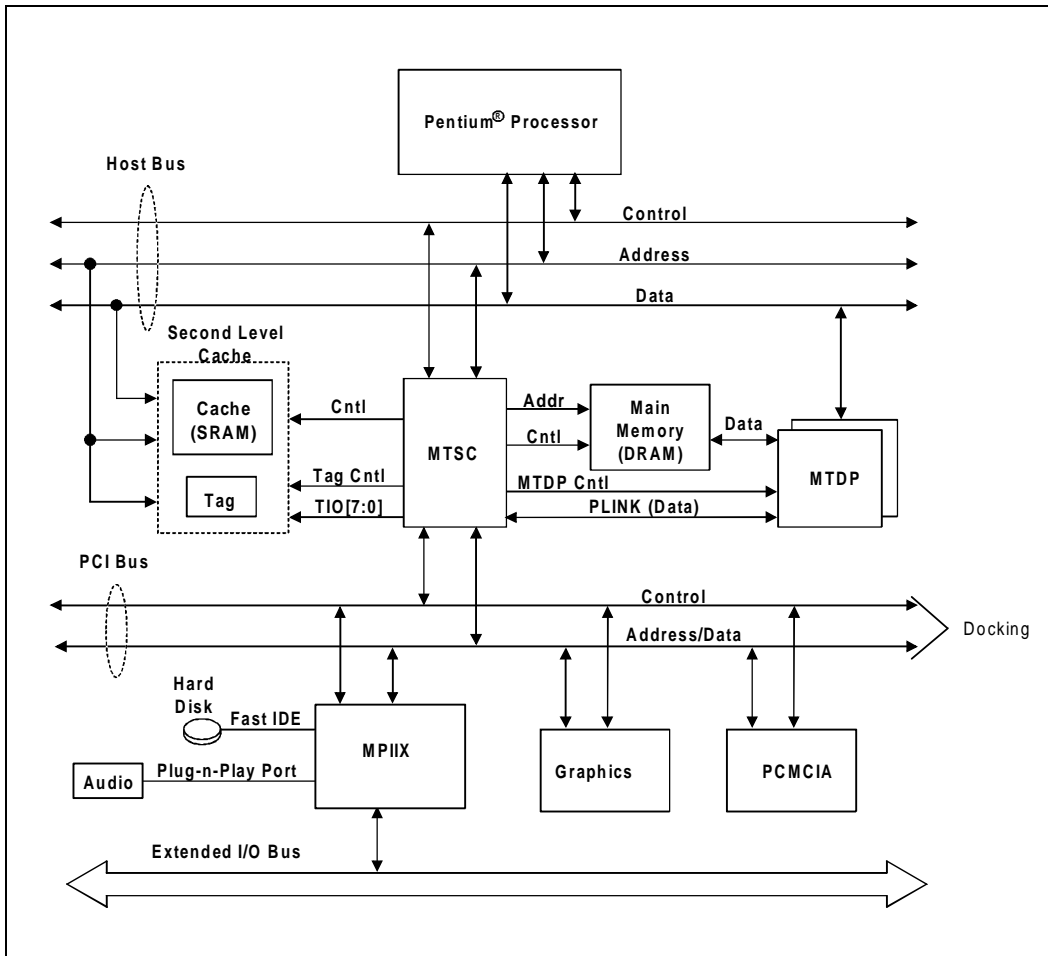


Figure 1. Intel 430MX PCiset System

### Buffers

The MTSC and MTDP together contain buffers for optimizing data flow. A 4-Qword deep buffer is provided for CPU-to-main memory writes, second level cache write back cycles, and PCI-to-main memory transfers. This buffer is used to achieve 3-1-1-1 posted writes to main memory. A 4-Dword buffer is used for CPU-to-PCI writes. In addition, a 4-Dword PCI Write Buffer is provided which is combined with the DRAM Write Buffer to supply a 12 Dword deep buffering for PCI to main memory writes.

### System Clocking

The processor, second level cache, main memory subsystem, and PLINK bus all run synchronous to the host clock. The PCI clock runs synchronously at half the host clock frequency. The MTSC and MTDP have a host clock input and the MTSC has a PCI clock input. These clocks are derived from an external source and have a maximum clock skew requirement with respect to each other. The PCI interface supports the CLKRUN# protocol as defined by the PCI Mobil Working Group.

## 2.0 SIGNAL DESCRIPTION

This section provides a detailed description of each signal. The signals are arranged in functional groups according to their associated interface.

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 terms assertion and negation are used extensively. This is done to avoid confusion when working with a mixture of 'active-low' and 'active-high' signals. The term **assert**, or **assertion** indicates that a signal is active, independent of whether that level is represented by a high or low voltage. The term **negate**, or **negation** indicates that a signal is inactive.

The "RST#" column indicates the state of the signals during reset.

The following notations are used to describe the signal type:

- I** Input is a standard input-only signal.
- O** Totem pole output is a standard active driver.
- o/d** Open drain.
- I/O** Tri-state is a bi-directional, tri-state input/output pin.
- s/t/s** Sustained tri-state is an active low tri-state signal owned and driven by one and only one agent at a time. The agent that drives a s/t/s pin low must drive it high for at least one clock before letting it float. A new agent can not start driving a s/t/s signal any sooner than one clock after the previous owner tri-states it. An external pull-up is required to sustain the inactive state until another agent drives it and must be provided by the central resource.
- 5/3V** Indicates that this signal is normally 5V, but will be powered by the RTC voltage on the VDDR "resume well" power supply pin during the suspend state at normal 3.3 volts.
- pu** Internal Pull-Up
- pd** Internal Pull-Down

### 2.1. MTSC Signals

#### 2.1.1. HOST INTERFACE (MTSC)

Signal Name	Type	RST#	Description
A[31:3]	I/O 3.3V 2mA	Low*	<b>ADDRESS BUS:</b> A[31:3] connect to the address bus of the CPU. During CPU cycles, A[31:3] are inputs. These signals are driven by the MTSC during cache snoop operations.  Note that A[31:28] provide poweron/reset strapping options for the second level cache and are inputs during reset. These signals must be strapped (high or low) and can not floated.
BE[7:0]#	I 3.3V		<b>BYTE ENABLES:</b> The CPU byte enables indicate which byte lane the current CPU cycle is accessing. All eight byte lanes are provided to the CPU if the cycle is a cacheable read regardless of the state of BE[7:0]#.
ADS#	I 3.3V		<b>ADDRESS STATUS:</b> The CPU asserts ADS# to indicate that a new bus cycle is being driven.

Signal Name	Type	RST#	Description
BRDY#	O 3.3V 4mA	High	<b>BUS READY:</b> The MTSC asserts BRDY# to indicate to the CPU that data is available on reads or has been received on writes.
NA#	O 3.3V 4mA	High	<b>NEXT ADDRESS:</b> When burst SRAMs are used in the second level cache or the second level cache is disabled, the MTSC asserts NA# in T2 during CPU write cycles and with the first assertion of BRDY# during CPU read cycles. NA# is never asserted if the second level cache is enabled with asynchronous SRAMs. NA# on the MTSC must be connected to the CPU NA# pin for all configurations.
AHOLD	O 3.3V		<b>ADDRESS HOLD:</b> The MTSC asserts AHOLD when a PCI master is accessing main memory. AHOLD is held for the duration of the PCI burst transfer. The MTSC negates AHOLD when the PCI to main memory read/write cycles complete and during PCI peer transfers.
EADS#	O 3.3V	High	<b>EXTERNAL ADDRESS STROBE:</b> Asserted by the MTSC to inquire the first level cache when servicing PCI master accesses to main memory.
BOFF#	O 3.3V 2mA	High	<b>BACK OFF:</b> Asserted by the MTSC when required to terminate a CPU cycle that was in progress.
HITM#	I 3.3V		<b>HIT MODIFIED:</b> Asserted by the CPU to indicate that the address presented with the last assertion of EADS# is modified in the first level cache and needs to be written back.
M/IO#, D/C#, W/R#	I 3.3V		<b>MEMORY/IO; DATA/CONTROL; WRITE/READ:</b> Asserted by the CPU with ADS# to indicate the type of cycle on the host bus.
HLOCK#	I 3.3V		<b>HOST LOCK:</b> All CPU cycles sampled with the assertion of HLOCK# and ADS#, until the negation of HLOCK# must be atomic (i.e., no PCI activity to main memory is allowed).
CACHE#	I 3.3V		<b>CACHEABLE:</b> Asserted by the CPU during a read cycle to indicate the CPU can perform a burst line fill. Asserted by the CPU during a write cycle to indicate that the CPU will perform a burst write-back cycle. If CACHE# is asserted to indicate cacheability, the MTSC asserts KEN# either with the first BRDY#, or with NA#, if NA# is asserted before the first BRDY#.
KEN#/INV	O 3.3V 4mA	Low	<p><b>CACHE ENABLE/INVALIDATE:</b> KEN#/INV functions as both the KEN# signal during CPU read cycles and the INV signal during first level cache snoop cycles. During CPU cycles, KEN#/INV is normally low. The MTSC drives KEN# high during the first BRDY# or NA# assertion of a non-cacheable (in first level cache) CPU read cycle.</p> <p>The MTSC drives INV high during the EADS# assertion of a PCI master DRAM write snoop cycle and low during the EADS# assertion of a PCI master DRAM read snoop cycle.</p>



Signal Name	Type	RST#	Description
SMIACK#	I 3.3V		<b>SYSTEM MANAGEMENT INTERRUPT ACTIVE:</b> The CPU asserts SMIACK# when it is in system management mode as a result of an SMI. This signal must be sampled active with ADS# for the processor to access the SMM space of DRAM.

**Note:** All signals are TTL

### 2.1.2. DRAM INTERFACE (MTSC)

Signal Name	Type	RST#	Description
WE#	O 3.3V 8mA	Low	<b>WRITE ENABLE:</b> This signal enables the write to DRAM. WE# is negated during refresh cycles.
RAS[3:0]#	O 3.3V 8mA	Low	<b>ROW ADDRESS STROBE:</b> These pins select the DRAM row.
CAS[7:0]#	O 3.3V 8mA	Low	<b>COLUMN ADDRESS STROBE:</b> These pins always select which bytes are affected by a DRAM cycle.
MA[11:2]	O 3.3V 2/8mA	Low	<b>MEMORY ADDRESS:</b> This is the row and column address for DRAM. The drive strength is programmable (2 ma or 8 ma) via the DRAM Timing register.
MA[1:0]	O 3.3V 8mA	Low	<b>MEMORY ADDRESS:</b> This is the row and column address for DRAM.

**Note:** All signals are TTL

### 2.1.3. SECONDARY CACHE INTERFACE (MTSC)

Signal Name	Type	RST#	Description
CADV#/ CA4	O 3.3V 4mA	High	<p><b>CACHE ADVANCE/CACHE ADDRESS 4:</b> This pin has two modes of operation depending on the type of SRAMs selected via hardware strapping options or programming the CC Register. The CA4 mode is used when the L2 cache consists of asynchronous SRAMs. CA4 is used to sequence through the Qwords in a cache line during a burst operation.</p> <p>CADV# mode is used when the L2 cache consists of burst SRAMs. In this mode, assertion causes the burst SRAM in the L2 cache to advance to the next Qword in the cache line.</p>

Signal Name	Type	RST#	Description
CADS#/ CA3	O 3.3V 4mA	High	<b>CACHE ADDRESS STROBE/CACHE ADDRESS 3:</b> This pin has two modes of operation depending on the type of SRAMs selected via hardware strapping options or programming the CC Register. The CA3 mode is used when the L2 cache consists of asynchronous SRAMs. CA3 is used to sequence through the Qwords in a cache line during a burst operation.  CADS# mode is used when the L2 cache consists of burst SRAMs. In this mode assertion causes the burst SRAM in the L2 cache to load the burst SRAM address register from the burst SRAM address pins.
CCS#	O 3.3V 4mA	High	<b>CACHE CHIP SELECT/CACHE ADDRESS:</b> A L2 cache consisting of burst SRAMs will power up, if necessary, and perform an access if CCS# is asserted when CADS# is asserted. A L2 cache consisting of burst SRAMs will power down if CCS# is negated when CADS# is asserted. When CCS# is negated, a L2 cache consisting of burst SRAMs ignores ADS#. If CCS# is asserted when ADS# is asserted, a L2 cache consisting of burst SRAMs will power up, if necessary, and perform an access.
COE#	O 3.3V	High	<b>CACHE OUTPUT ENABLE:</b> The secondary cache data RAMs drive the CPU's data bus when COE# is asserted.
CWE[7:0]#	O 3.3V	High	<b>CACHE WRITE ENABLE:</b> Each CWE# corresponds to one byte lane. Assertion causes the byte lane to be written into the secondary cache data RAMs if they are powered up.
TIO[7:0]	I/O 3.3 V 2mA pu50 K	Tri-state	<b>TAG ADDRESS:</b> These are inputs during CPU accesses and outputs during L2 cache line fills and L2 cache line invalidates due to inquire cycles. TIO[7:0] contain the L2 tag address for 256-Kbyte L2 caches. TIO[6:0] contains the L2 tag address and TIO7 contains the L2 cache valid bit for 512-Kbyte caches. These pins have a 50 K $\Omega$ pull-up resistor that is disabled after reset if the cache size is non-zero. If the cache size is zero, these pull-up resistors are enabled, which prevents the TIO signals from floating if no L2 cache is present in the system.
TWE#	O 3.3 V 2mA	High	<b>TAG WRITE ENABLE:</b> When asserted, new state and tag addresses are written into the external tag.

**Note:** All signals are TTL

#### 2.1.4. PCI INTERFACE (MTSC)

Signal Name	Type	RST#	Description
AD[31:0]	I/O 5V	Low	<b>ADDRESS DATA BUS:</b> The standard PCI address and data lines. The address is driven with FRAME# assertion and data is driven or received in following clocks.
C/BE[3:0]#	I/O 5V	Low	<b>COMMAND, BYTE ENABLE:</b> The command is driven with FRAME# assertion. Byte enables corresponding to supplied or requested data are driven on following clocks.

Signal Name	Type	RST#	Description
FRAME#	I/O 5V	Tri-state	<b>FRAME:</b> Assertion indicates the address phase of a PCI transfer. Negation indicates that one more data transfer is desired by the cycle initiator. An external 2.7 K $\Omega$ pull-up resistor should be connected to this signal.
DEVSEL#	I/O 5V	Tri-state	<b>DEVICE SELECT:</b> The MTSC drives DEVSEL# when a PCI initiator attempts to access main memory. DEVSEL# is asserted at medium decode time. An external 2.7 K $\Omega$ pull-up resistor should be connected to this signal.
IRDY#	I/O 5V	Tri-state	<b>INITIATOR READY:</b> Asserted when the initiator is ready for a data transfer. An external 2.7 K $\Omega$ pull-up resistor should be connected to this signal.
TRDY#	I/O 5V	Tri-state	<b>TARGET READY:</b> Asserted when the target is ready for a data transfer. An external 2.7 K $\Omega$ pull-up resistor should be connected to this signal.
STOP#	I/O 5V	Tri-state	<b>STOP:</b> Asserted by the target to request the master to stop the current transaction. An external 2.7 K $\Omega$ pull-up resistor should be connected to this signal.
LOCK#	I/O 5V	Tri-state	<b>LOCK:</b> Used to establish, maintain, and release bus locks on PCI. An external 2.7 K $\Omega$ pull-up resistor should be connected to this signal.
REQ[2:0]#	I 5V pu20 K TTL		<b>REQUEST:</b> PCI master requests for PCI.
GNT[2:0]#	O 5V TTL	Tri-state	<b>GRANT:</b> Permission is given to the master to use PCI.
PHLD#	I 5V pu20 k TTL		<b>PCI HOLD:</b> This signal comes from the MPIIX. PHLD# is the MPIIX request for the PCI Bus. The MTSC flushes the DRAM Write Buffers and acquires the host bus before granting MPIIX via PHLDA#.
PHLDA#	O 5V TTL	Tri-state	<b>PCI HOLD ACKNOWLEDGE:</b> This signal is driven by the MTSC to grant PCI to the MPIIX.
PAR	I/O 5V	Low	<b>PARITY:</b> A single parity bit is provided over AD[31:0] and C/BE[3:0].
RST#	I 5V		<b>RESET:</b> When asserted, RST# resets the MTSC and sets all register bits to the default value. PCI signals tri-state compliant to the PCI Rev 2.0 specification.

Signal Name	Type	RST#	Description
CLKRUN#	I/OD 5V	Tri-state	<b>CLOCK RUN:</b> The MTSC requests the central resource (MPIIX) to start or maintain the PCI clock by asserting of CLKRUN#. This signal is tri-stated during reset. An external 2.7 K $\Omega$ pull-up resistor should be connected to this signal.

**Note:** All signals in the PCI Interface conform to the PCI Rev 2.0 specification.

#### 2.1.5. MTPD INTERFACE (MTSC)

Signal Name	Type	RST#	Description
PLINK[15:0]	I/O 3.3V 2mA	Low	<b>PCI LINK:</b> This is the data path between the CPU/main memory and PCI (via the MTSC). PCI main memory reads and CPU to PCI writes are driven onto these pins by the MTD. CPU reads from PCI and PCI writes to main memory are received on this bus by the MTD. Each MTD connects to one byte of this bus.
MSTB#	O 3.3V 4mA	High	<b>MEMORY STROBE:</b> Assertion causes data to be posted in the DRAM Write Buffer.
MADV#	O 3.3V 8mA	High	<b>MEMORY ADVANCE:</b> For memory write cycles, assertion causes a Qword to be drained from the DRAM Write Buffer and the next data to be made available to the MD pins of the MTDs. For memory read cycles, assertion causes a Qword to be latched in the DRAM Input Register.
PCMD[1:0]	O 3.3V 4mA	High	<b>PLINK COMMAND:</b> This field controls how data is loaded into the PLINK input and output registers.
HOE#	O 3.3V 4mA	High	<b>HOST OUTPUT ENABLE:</b> This signal is used as the output enable for the Host Data Bus.
MOE#	O 3.3V 4mA	Low	<b>MEMORY OUTPUT ENABLE:</b> This signal is used as the output enable for the memory data bus.
POE#	O 3.3V 4mA	Low	<b>PLINK OUTPUT ENABLE:</b> This signal is used as the output enable for the PLINK Data Bus.

**Note:** All signals are TTL

#### 2.1.6. CLOCKS (MTSC)

Signal Name	Type	Description
PWROK	I 5/3V CMOS	<b>POWER OK:</b> When asserted, PWROK is an indication to the MTSC that power has been stable for at least 1 ms. PWROK can be driven asynchronously.

HCLKIN	I 3.3V (5V Safe)	<b>HOST CLOCK IN:</b> This pin receives a buffered host clock. This clock is used by all of the MTSC logic that is in the Host clock domain. This should be the same clock net that is delivered to the CPU. The net should tee and have equal lengths from the tee to the CPU and the MTSC.
PCLKIN	I 5V	<b>PCI CLOCK IN:</b> This pin receives a buffered divide-by-2 host clock. This clock is used by all of the MTSC logic that is in the PCI clock domain.

### 2.1.7. POWER MANAGEMENT (MTSC)

Signal Name	Type	Description
RTCCLK	I 5/3V CMOS	<b>REAL TIME CLOCK:</b> This signal provides a 32 KHz input for clocking events in the suspend state (e.g., DRAM refresh)
PWRSD	I 5/3V CMOS	<b>POWER SUSPEND TO DRAM:</b> PWRSD indicates the power supply state during suspend to DRAM mode.

## 2.2. MTDP Signals

### 2.2.1. DATA INTERFACE SIGNALS (MTDP)

Signal Name	Type	RST#	Description
HD[31:0]	I/O 3.3V	Tri-state	<b>HOST DATA:</b> These signals are connected to the CPU data bus. The CPU data bus is interleaved between the two MTDPs for every byte, effectively creating an even and an odd MTDP.
MD[31:1]	I/O 3.3V/ 5V	Tri-state	<b>MEMORY DATA:</b> These signals are connected to the DRAM data bus. The DRAM data bus is interleaved between the two MTDPs for every byte, effectively creating an even and an odd MTDP.
MD[0]	I/O 3.3V/ 5V	NAND Tree Output	<b>MEMORY DATA:</b> These signals are connected to the DRAM data bus. The DRAM data bus is interleaved between the two MTDPs for every byte, effectively creating an even and an odd MTDP.
PLINK[7:0]	I/O 3.3V 2mA	Tri-state	<b>PCI LINK:</b> These signals are connected to the PLINK data bus on the MTSC. This is the data path between the MTSC and MTDP. Each MTDP connects to one byte of the 16-bit bus.

**Note:** All signals are TTL

### 2.2.2. MTSC INTERFACE SIGNALS (MTDP)

Signal Name	Type	Description
MSTB#	I 3.3V TTL	<b>MEMORY STROBE:</b> Assertion causes data to be posted in the DRAM Write Buffer.

Signal Name	Type	Description
MADV#	I 3.3V TTL	<b>MEMORY ADVANCE:</b> For memory write cycles, assertion causes a Qword to be flushed from the DRAM Write Buffer and the next data to be made available to the MD pins of the MTDps. For memory read cycles, assertion causes a Qword to be latched in the DRAM Input register.
PCMD[1:0]	I 3.3V TTL	<b>PLINK COMMAND:</b> This field controls how data is loaded into the PLINK input and output registers.
HOE#	I 3.3V TTL	<b>HOST OUTPUT ENABLE:</b> This signal is used as the output enable for the Host Data Bus.
MOE#	I 3.3V TTL	<b>MEMORY OUTPUT ENABLE:</b> This signal is used as the output enable for the Memory Data Bus.
POE#	I 3.3V TTL	<b>PLINK OUTPUT ENABLE:</b> This signal is used as the output enable for the PLINK Data Bus.

### 2.2.3. CLOCK SIGNAL (MTDP)

Signal Name	Type	Description
HCLK	I 3.3V (5V Safe) TTL	<b>HOST CLOCK:</b> Primary clock input used to drive the part.

### 2.3. Strapping Options

Name	Pin Name	Description
SCS	A[31:30]	<b>Secondary Cache Size</b> as described in the Cache Control Register (bits [7:6]). There are no pullup/pulldown resistors implemented in the A[31:30] I/O buffers. These signals must be strapped (high or low) and can not be floated.
L2RAMT	A[29:28]	<b>SRAM Type</b> as described in the Cache Control Register (bits [5:4]). There are no pullup/pulldown resistors implemented in the A[29:28] I/O buffers. These signals must be strapped (high or low) and can not be floated.

### 3.0. REGISTER DESCRIPTION

The MTSC contains two sets of software accessible registers (Control and Configuration registers), accessed via the Host CPU I/O address space. Control Registers control access to PCI configuration space. Configuration Registers reside in PCI configuration space and specify PCI configuration, DRAM configuration, cache configuration, operating parameters, and optional system features.

The MTSC internal registers (both I/O Mapped and Configuration registers) are only accessible by the Host CPU and cannot be accessed by PCI masters. The registers can be accessed as Byte, Word (16-bit), or Dword (32-bit) quantities, with the exception of CONFADD 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). The following nomenclature is used for access attributes.

- 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/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.

Some of the MTSC registers described in this section contain reserved bits. 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.

In addition to reserved bits within a register, the MTSC contains address locations in the PCI configuration space that are marked "Reserved" (Table 1). The MTSC responds to accesses to these address locations by completing the Host cycle. Software should not write to reserved MTSC configuration locations in the device-specific region (above address offset 3Fh).

During a hard reset (RST# asserted), the MTSC sets its internal configuration registers to predetermined **default** states. 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, cache configuration, operating parameters and optional system features that are applicable, and to program the MTSC registers accordingly.

#### 3.1. Control Registers

The MTSC contains two registers that reside in the CPU I/O address space—the Configuration Address (CONFADD) Register and the Configuration Data (CONFDATA) Register. These registers can not reside in PCI configuration space because of the special functions they perform. The Configuration Address Register enables/disables the configuration space and determines what portion of configuration space is visible through the Configuration Data window.

##### 3.1.1. CONFADD—CONFIGURATION ADDRESS REGISTER

I/O Address:      0CF8h (Dword access only)  
Default Value:    00000000h  
Access:            Read/Write

CONFADD is a 32-bit register accessed only when referenced as a Dword. A Byte or Word reference will "pass through" the Configuration Address Register to the PCI bus. The CONFADD Register contains the Bus Number, Device Number, Function Number, and Register Number for which a subsequent configuration access is intended.

Bit	Descriptions
31	<b>Configuration Enable (CONE):</b> 1=Enable; 0=Disable.
30:24	<b>Reserved.</b>
23:16	<b>Bus Number (BUSNUM):</b> When BUSNUM is programmed to 00h, the target of the configuration cycle is either the MTSC or the PCI Bus that is directly connected to the MTSC, depending on the Device Number field. If the Bus Number is programmed to 00h and the MTSC is not the target, a type 0 configuration cycle is generated on PCI. If the Bus Number is non-zero, a type 1 configuration cycle is generated on PCI with the Bus Number mapped to AD[23:16] during the address phase.
15:11	<b>Device Number (DEVNUM):</b> This field selects one agent on the PCI bus selected by the Bus Number. During a Type 1 Configuration cycle, this field is mapped to AD[15:11]. During a Type 0 configuration cycle, this field is decoded and one of AD[31:11] is driven to a 1. The MTSC is always Device Number 0.
10:8	<b>Function Number (FUNCNUM):</b> This field is mapped to AD[10:8] during PCI configuration cycles. This allows the configuration registers of a particular function in a multi-function device to be accessed. The MTSC responds to configuration cycles with a function number of 000b; all other function number values attempting access to the MTSC (Device Number = 0, Bus Number = 0) generate a type 0 configuration cycle on the PCI Bus with no IDSEL asserted, which results in a master abort.
7:2	<b>Register Number (REGNUM):</b> 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 AD[7:2] during PCI configuration cycles.
1:0	<b>Reserved.</b>

### 3.1.2. CONFDATA—CONFIGURATION DATA REGISTER

I/O Address: 0CFCh  
 Default Value: 00000000h  
 Access: Read/Write

CONFDATA is a 32-bit read/write window into configuration space. The portion of configuration space that is referenced by CONFDATA is determined by the contents of CONFADD.

Bit	Descriptions
31:0	<b>Configuration Data Window (CDW):</b> If bit 31 of CONFADD is 1, any I/O reference in the CONFDATA I/O space is mapped to configuration space using the contents of CONFADD.

### 3.2. PCI Configuration Registers

The PCI Bus defines a slot based "configuration space" that allows each device to contain 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**. While memory and I/O spaces are supported by the Pentium microprocessor, configuration space is not supported. The PCI specification defines two mechanisms to access configuration space, Mechanism #1 and Mechanism #2. The MTSC only supports Mechanism #1. Table 1 shows the MTSC configuration space.



The configuration access mechanism makes use of the CONFADD Register and CONFDATA Register. To reference a configuration register, a Dword I/O write cycle is used to place a value into CONFADD 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. CONFADD[31] must be 1 to enable a configuration cycle. Then, CONFDATA becomes a window onto four bytes of configuration space specified by the contents of CONFADD. Read/write accesses to CONFDATA generates a PCI configuration cycle to the address specified by CONFADD.

#### Type 0 Access

If the Bus Number field of CONFADD is 0, a type 0 configuration cycle is generated on PCI. CONFADD[10:2] is mapped directly to AD[10:2]. The Device Number field of CONFADD is decoded onto AD[31:11]. The MTSC is Device #0 and does not pass its configuration cycles to PCI. Thus, AD11 is never asserted. (For accesses to device #1, AD12 is asserted, etc., to Device #20 which asserts AD31.) Only one AD line is asserted at a time. All device numbers higher than 20 cause a type 0 configuration access with no IDSEL asserted, which results in a master abort.

#### Type 1 Access

If the Bus Number field of CONFADD is non-zero, a type 1 configuration cycle is generated on PCI. CONFADD[23:2] are mapped directly to AD[23:2]. AD[1:0] are driven to 01 to indicate a Type 1 Configuration cycle. All other lines are driven to 0.

**Table 1. MTSC Configuration Space**

Address Offset	Symbol	Register Name	Access
00–01h	VID	Vendor Identification	RO
02–03h	DID	Device Identification	RO
04–05h	PCICMD	Command	R/W
06–07h	PCISTS	Status	RO, R/WC
08	RID	Revision Identification	RO
09–0Bh	CLASSC	Class Code	RO
0Ch	—	Reserved	—
0Dh	MLT	Master Latency Timer	R/W
0Eh	—	Reserved	—
0Fh	BIST	BIST	R/W
10–4Fh	—	Reserved	—
50	PCON	PCI Control	R/W
51h	—	Reserved	—
52h	CC	Cache Control	R/W
53–56h	—	Reserved	—
57h	DRAMC	DRAM Control	R/W
58h	DRAMT	DRAM Timing	R/W
59–5Fh	PAM[6:0]	Programmable Attribute Map (7 registers)	R/W

Address Offset	Symbol	Register Name	Access
60–63h	DRB[3:0]	DRAM Row Boundary (4 registers)	R/W
64–67h	—	Reserved	—
68h	DRT	DRAM Row Type	R/W
69–71h	—	Reserved	—
72h	SMRAM	System Management RAM Control	R/W
73–FFh	—	Reserved	—

### 3.2.1. VID—VENDOR IDENTIFICATION REGISTER

Address Offset: 00–01h  
 Default Value: 8086h  
 Attribute: Read Only

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

Bit	Description
15:0	<b>Vendor Identification Number.</b> This is a 16-bit value assigned to Intel.

### 3.2.2. DID—DEVICE IDENTIFICATION REGISTER

Address Offset: 02–03h  
 Default Value: 1235h  
 Attribute: Read Only

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	<b>Device Identification Number.</b> This is a 16-bit value assigned to the MTSC.

### 3.2.3. PCICMD—PCI COMMAND REGISTER

Address Offset: 04–05h  
 Default: 06h  
 Access: Read/Write

This register controls the MTSC's ability to respond to PCI cycles.

Bit	Descriptions
15:10	<b>Reserved.</b>
9	<b>Fast Back-to-Back. (Not Implemented)</b> This bit is hardwired to 0.
8	<b>SERR# Enable (SERRE). (Not Implemented)</b> This bit is hardwired to 0.

Bit	Descriptions
7	<b>Address/Data Stepping. (Not Implemented)</b> This bit is hardwired to 0.
6	<b>Parity Error Enable (PERRE). (Not Implemented)</b> This bit is hardwired to 0.
5:3	<b>Reserved.</b> These bits are hardwired to 0.
2	<b>Bus Master Enable (BME). (Not Implemented)</b> The MTSC does not support disabling of its bus master capability on the PCI Bus. This bit is hardwired to 1.
1	<b>Memory Access Enable (MAE).</b> 1=Enable PCI master access to main memory, if the PCI address selects enabled DRAM space; 0=Disable (MTSC does not respond to main memory accesses).
0	<b>I/O Access Enable (IOAE). (Not Implemented)</b> This bit is hardwired to 0. The MTSC does not respond to PCI I/O cycles.

### 3.2.4. PCISTS—PCI STATUS REGISTER

Address Offset: 06–07h  
Default Value: 0200h  
Access: Read Only, Read/Write Clear

PCISTS reports the occurrence of a PCI master abort and PCI target abort. PCISTS also indicates the DEVSEL# timing that has been set by the MTSC hardware.

Bit	Descriptions
15	<b>Detected Parity Error (DPE). (Not Implemented)</b> This bit is hardwired to 0.
14	<b>Signaled System Error (SSE)—R/WC.</b> This bit is hardwired to 0.
13	<b>Received Master Abort Status (RMAS)—R/WC.</b> When the MTSC terminates a Host-to-PCI transaction (MTSC is a PCI master) with an unexpected master abort, this bit is set to 1. Note that master abort is the normal and expected termination of PCI special cycles. Software sets this bit to 0 by writing a 1 to it.
12	<b>Received Target Abort Status (RTAS)—R/WC.</b> When a MTSC-initiated PCI transaction is terminated with a target abort, RTAS is set to 1. Software sets RTAS to 0 by writing a 1 to it.
11	<b>Signaled Target Abort Status (STAS).</b> This bit is hardwired to 0. The MTSC never terminates a PCI cycle with a target abort.
10:9	<b>DEVSEL# Timing (DEVT)—RO.</b> This 2-bit field indicates the timing of the DEVSEL# signal when the MTSC responds as a target, and is hard-wired to the value 01b (medium) to indicate the slowest time that DEVSEL# is generated.
8	<b>Data Parity Detected (DPD)—R/WC.</b> This bit is hardwired to 0
7	<b>Fast Back-to-Back (FB2B). (Not Implemented)</b> This bit is hardwired to 0.
6:0	<b>Reserved.</b>

### 3.2.5. RID—REVISION IDENTIFICATION REGISTER

Address Offset: 08h  
 Default Value: See stepping information document  
 Access: Read Only

This register contains the revision number of the MTSC.

Bit	Description
7:0	<b>Revision Identification Number.</b> This is an 8-bit value that indicates the revision identification number for the MTSC.

### 3.2.6. CLASSC—CLASS CODE REGISTER

Address Offset: 09–0Bh  
 Default Value: 060100h  
 Attribute: Read Only

This register contains the device programming interface information related to the Sub-Class Code and Base Class Code definition for the MTSC. This register also identifies the Base Class Code and the function sub-class in relation to the Base Class Code.

Bit	Description
23:16	<b>Base Class Code (BASEC).</b> 06h=Bridge device.
15:8	<b>Sub-Class Code (SCC).</b> 00h=Host Bridge.
7:0	<b>Programming Interface (PI).</b> 00h=No register-level programming interface defined.

### 3.2.7. MLT—MASTER LATENCY TIMER REGISTER

Address Offset: 0Dh  
 Default Value: 00h  
 Access: Read/Write

MLT is an 8-bit register that controls the amount of time the MTSC, as a bus master, can burst data on the PCI Bus. The Count Value is an 8-bit quantity. However, MLT[2:0] are hardwired to 0. MLT is also used to guarantee the host CPU a minimum amount of the system resources as described in the PCI Bus Arbitration section.

Bit	Description
7:3	<b>Master Latency Timer Count Value.</b> The number of clocks programmed in the MLT represents the minimum guaranteed time slice (measured in PCI clocks) allotted to the MTSC, after which it must surrender the bus as soon as other PCI masters are granted the bus. The default value of MLT is 00h or 0 PCI clocks. However, this field should always be programmed to a non-zero value (recommended value is 20h or 32 PCI clocks). If the MTSC MLT register is programmed to 0, there exists a possibility that a system with multiple PCI masters could keep the CPU from obtaining the PCI bus.
2:0	<b>Reserved.</b> Hardwired to 0.

### 3.2.8. BIST—BIST REGISTER

Address Offset: 0Fh  
Default: 00h  
Access: Read/Write

The Built In Self Test (BIST) function is not supported by the MTSC. Writes to this register have no affect.

Bit	Descriptions
7	<b>BIST Supported—RO (Not Implemented).</b> Hardwired to 0.
6	<b>Start BIST (Not Implemented).</b> Hardwired to 0.
5:4	<b>Reserved.</b>
3:0	<b>Completion Code—RO (Not Implemented).</b> Hardwired to 0.

### 3.2.9. PCON—PCI CONTROL REGISTER

Address Offset: 50h  
Default: 40h  
Access: Read/Write

The PCON Register enables/disables peer concurrency.

Bit	Descriptions																				
7:5	<p><b>CPU Inactivity Timer (CIT).</b> This field selects the value used in the CPU Inactivity Timer. This timer counts CPU inactivity in PCI clocks. The inactivity window is defined as the last BRDY# to the next ADS#. When active, the CPU is default owner of the PCI Bus. If the CPU is inactive, PHOLD and REQx# lines are given priority.</p> <table><tr><th>Bits[7:5]</th><th>PCI Clocks</th><th>Bits[7:5]</th><th>PCI Clocks</th></tr><tr><td>000</td><td>1</td><td>100</td><td>5 *</td></tr><tr><td>001</td><td>2</td><td>101</td><td>6</td></tr><tr><td>010</td><td>3 *</td><td>110</td><td>7</td></tr><tr><td>011</td><td>4</td><td>111</td><td>8</td></tr></table> <p>* Recommended Settings</p>	Bits[7:5]	PCI Clocks	Bits[7:5]	PCI Clocks	000	1	100	5 *	001	2	101	6	010	3 *	110	7	011	4	111	8
Bits[7:5]	PCI Clocks	Bits[7:5]	PCI Clocks																		
000	1	100	5 *																		
001	2	101	6																		
010	3 *	110	7																		
011	4	111	8																		
4	<b>Reserved.</b>																				
3	<p><b>Peer Concurrency Enable (PCE).</b> 1=Enable. 0=Disable. When Peer Concurrency is disabled, MTSC will back off CPU transactions during PCI peer to peer traffic. When enabled, the MTSC allows the CPU to run DRAM or L2 cache cycles when non-PHLD# PCI masters are running non-locked cycles targeting non-MTSC PCI peer devices.</p>																				
2	<p><b>PCI Bursting Disable (PBD):</b> 1=Disable. 0=Enable. This bit disables the CPU to PCI burst transfer. If disabled, CPU writes to PCI will not be bursted (note that postable cycles are still posted and QWORD data is still bursted together). If enabled, write bursting will function normally. Note that this bit has no effect on the ability for PCI masters to burst to DRAM, nor does it affect the posting of write data in the CPU2PCI buffer.</p>																				

Bit	Descriptions
1	<b>PCI Streaming Disable (PSD):</b> 1=Disable. 0=Enable. This bit disables the PCI streaming transfers to/from DRAM. If disabled, the MTSC only generates a snoop for the starting address of a DRAM transfer. Additionally, the MTSC will disconnect the PCI transfer at the final DWord of a cache line. If enabled, streaming and snoop ahead will function normally.
0	<b>Bus Concurrency Disable (CD):</b> 1=Disable. 0=Enable. When disabled, the PCI arbiter handles all PCI agent requests as if they were PHLD#-type requests. The host bus is acquired, and the CPU-to-PCI and DRAM write buffers are drained prior to granting to PCI, CPU/PCI-peer transfer concurrency and CPU/CPU-to-PCI drain concurrency are eliminated. When enabled, normal arbitration and buffer management policies are used.

### 3.2.10. CC—CACHE CONTROL REGISTER

Address Offset: 52h  
Default: SSSS0010b (S = Strapping option)  
Access: Read/Write

The CC Register selects the secondary cache operations. This register enables/disables the L2 cache, adjusts cache size, defines the cache SRAM type, and controls tag initialization. After a hard reset, CC[7:4] reflect the inverted signal levels on the host address lines A[31:28].

Note: An I/O read cycle (e.g., reading this register) is required immediately following any write that changes a value in the CC Register.

Bit	Description										
7:6	<p><b>Secondary Cache Size (SCS).</b> This field reflects the inverted signal level on the A[31:30] pins at the rising edge of the RST# signal (default). The default values can be overwritten with subsequent writes to the CC Register. The options for this field are:</p> <table> <tr> <th>Bits[7:6]</th><th>Secondary Cache Size</th></tr> <tr> <td>0 0</td><td>Cache Disabled</td></tr> <tr> <td>0 1</td><td>256 Kbytes</td></tr> <tr> <td>1 0</td><td>512 Kbytes</td></tr> <tr> <td>1 1</td><td>Reserved</td></tr> </table> <p style="text-align: center;"><b>NOTE</b></p> <ol style="list-style-type: none"> <li>When SCS=00, the L2 cache is disabled and the cache tag state is frozen.</li> <li>When SCS≠00, the FLCE bit must be set to 1 (enable L1 cache).</li> </ol>	Bits[7:6]	Secondary Cache Size	0 0	Cache Disabled	0 1	256 Kbytes	1 0	512 Kbytes	1 1	Reserved
Bits[7:6]	Secondary Cache Size										
0 0	Cache Disabled										
0 1	256 Kbytes										
1 0	512 Kbytes										
1 1	Reserved										
5:4	<p><b>SRAM Type (SRAMT).</b> This field reflects the inverted signal level on the A[29:28] pins at the rising edge of the RST# signal. (default). The default values can be overwritten with subsequent writes to the CC Register. The options for this field are:</p> <table> <tr> <th>Bits[7:6]</th><th>SRAM Type</th></tr> <tr> <td>0 0</td><td>Pipelined Burst</td></tr> <tr> <td>0 1</td><td>Burst</td></tr> <tr> <td>1 0</td><td>Asynchronous</td></tr> <tr> <td>1 1</td><td>Pipelined Burst for 512K/dual-bank implementations</td></tr> </table> <p>When selected for 512K dual-bank pipelined burst (SRAMT=11), the back-to-back burst timings with NA# enabled are 3-1-1-1-2-1-1-1 instead of 3-1-1-1-1-1-1. An extra clock is inserted for bank turn-around. SCS must be set to 01.</p>	Bits[7:6]	SRAM Type	0 0	Pipelined Burst	0 1	Burst	1 0	Asynchronous	1 1	Pipelined Burst for 512K/dual-bank implementations
Bits[7:6]	SRAM Type										
0 0	Pipelined Burst										
0 1	Burst										
1 0	Asynchronous										
1 1	Pipelined Burst for 512K/dual-bank implementations										

Bit	Description															
3	<b>NA# Disable (NAD).</b> 1=NA# (CPU pipelining) disabled. 0=NA# (CPU pipelining) is enabled. This bit is to be set to 1 for factory level board testing only. When NAD = 0, CPU cycle pipelining is enabled and NA# is asserted as appropriate by the MTSC. When NAD = 1, CPU cycle pipelining is disabled and NA# will never be asserted. This allows the NA# signal to be disconnected from the processor on the system board. This bit should be configured before either the L1 or L2 caches are enabled.															
2	<b>Reserved.</b>															
1	<b>Secondary Cache Force Miss or Invalidate (SCFMI).</b> When SCFMI=1, the L2 hit/miss detection is disabled, and all tag lookups result in a miss. If the L2 is enabled, the cycle is processed as a miss. If the L2 is populated but disabled (FLCE=0) and SCFMI=1, any CPU read cycle invalidates the selected tag entry. When SCFMI=0, normal L2 cache hit/miss detection and cycle processing occurs.  Software can flush the cache (cause all modified lines to be written back to main memory) by setting SCFMI to a 1 with the L2 cache enabled (SCS≠00 and FLCE=1), and reading all L2 cache tag address locations.															
0	<b>First Level Cache Enable (FLCE).</b> FLCE enables/disables the first level cache. When FLCE=1, the MTSC responds to CPU cycles with KEN# asserted for cacheable memory cycles. When FLCE=0, KEN# is always negated and line fills to either the first level or L2 cache are prevented. Note that, when FLCE=1 and SCFMI=1, writes to the cache are also forced as misses. Thus, it is possible to create incoherent data between main memory and the L2 cache. A summary of FLCE/SCFMI bit interactions is as follows:  <table><tr><th>FLCE</th><th>SCFMI</th><th>L2 Cache Result</th></tr><tr><td>0</td><td>0</td><td>Disabled</td></tr><tr><td>0</td><td>1</td><td>Disabled; tag invalidate on reads</td></tr><tr><td>1</td><td>0</td><td>Normal L2 cache operation (dependent on SCS)</td></tr><tr><td>1</td><td>1</td><td>Enabled; miss forced on reads/writes</td></tr></table>	FLCE	SCFMI	L2 Cache Result	0	0	Disabled	0	1	Disabled; tag invalidate on reads	1	0	Normal L2 cache operation (dependent on SCS)	1	1	Enabled; miss forced on reads/writes
FLCE	SCFMI	L2 Cache Result														
0	0	Disabled														
0	1	Disabled; tag invalidate on reads														
1	0	Normal L2 cache operation (dependent on SCS)														
1	1	Enabled; miss forced on reads/writes														

### 3.2.11. DRAMC—DRAM CONTROL REGISTER

Address Offset: 57h  
Default Value: 00h  
Access: Read/Write

This 8-bit register controls main memory DRAM operating modes and features.

Bit	Description										
7:6	<p><b>Hole Enable (HEN).</b> This field enables a memory hole in main memory space. CPU cycles matching an enabled hole are passed on to PCI. PCI cycles matching an enabled hole are ignored by the MTSC (no DEVSEL#). Note that a selected hole is not remapped. Note that this field should not be changed while the L2 cache is enabled.</p> <table> <tr> <th>Bits[7:6]</th><th>Hole Enabled</th></tr> <tr> <td>00</td><td>None</td></tr> <tr> <td>01</td><td>512–640 Kbytes</td></tr> <tr> <td>10</td><td>15–16 Mbytes</td></tr> <tr> <td>11</td><td>Reserved</td></tr> </table>	Bits[7:6]	Hole Enabled	00	None	01	512–640 Kbytes	10	15–16 Mbytes	11	Reserved
Bits[7:6]	Hole Enabled										
00	None										
01	512–640 Kbytes										
10	15–16 Mbytes										
11	Reserved										
5	Reserved.										

Bit	Description																		
4	<b>Suspend Refresh Type (SRT).</b> This bit selects the type of refresh used during suspend to DRAM. 1=Self refreshing DRAMs are in system. 0=CAS-before-RAS refresh.																		
3	<b>EDO Detect Mode Enable (EDME).</b> This bit, if set to a 1, enables a special timing mode for BIOS to detect EDO DRAM type on a bank-by-bank basis. Once all DRAM row banks have been tested for EDO, the EDME bit should be set to 0. Otherwise, performance will be seriously impacted. An algorithm for using the EDME bit 3 is provide in the Functional Description Section (DRAM Interface).																		
2:0	<p><b>DRAM Refresh Rate (DRR).</b> The DRAM refresh rate is adjusted to accommodate low-power extended refresh capable DRAMs. The refresh rates are based on a 32 KHz RTC clock.</p> <table> <tr> <th>Bits[2:0]</th><th>Refresh Rate</th></tr> <tr> <td>000</td><td>15.6 <math>\mu</math>s</td></tr> <tr> <td>001</td><td>31.2 <math>\mu</math>s</td></tr> <tr> <td>010</td><td>62.4 <math>\mu</math>s</td></tr> <tr> <td>011</td><td>125 <math>\mu</math>s</td></tr> <tr> <td>100</td><td>250 <math>\mu</math>s</td></tr> <tr> <td>101</td><td>Reserved</td></tr> <tr> <td>110</td><td>Reserved</td></tr> <tr> <td>111</td><td>Reserved</td></tr> </table>	Bits[2:0]	Refresh Rate	000	15.6 $\mu$ s	001	31.2 $\mu$ s	010	62.4 $\mu$ s	011	125 $\mu$ s	100	250 $\mu$ s	101	Reserved	110	Reserved	111	Reserved
Bits[2:0]	Refresh Rate																		
000	15.6 $\mu$ s																		
001	31.2 $\mu$ s																		
010	62.4 $\mu$ s																		
011	125 $\mu$ s																		
100	250 $\mu$ s																		
101	Reserved																		
110	Reserved																		
111	Reserved																		

### 3.2.12. DRAMT—DRAM TIMING REGISTER

Address Offset: 58h  
Default Value: 00h  
Access: Read/Write

This 8-bit register controls main memory DRAM timings. While most system designs will be able to use one of the faster burst mode timings, slower rates may be required in certain system designs to support layouts with longer trace lengths or slower DRAMs.

Bit	Description															
7	<b>Buffer Strength MA[11:2]:</b> 1 = select 8 mA buffers. 0 = select 2 mA buffers (default). The MA[11:2] buffers are programmable for 2mA or 8 mA drive strength. The 8mA drive allows for increased loading without need for external buffers on the DRAM address bus.															
6:5	<p><b>DRAM Read Burst Timing (DRBT).</b> The DRAM read burst timings are controlled by the DRBT field. The timing used depends on the type of DRAM on a per-bank basis, as indicated by the DRT register.</p> <table><tr><th>DRBT</th><th>EDO Burst Rate</th><th>Standard Page Mode Rate</th></tr><tr><td>00</td><td>x444</td><td>x444</td></tr><tr><td>01</td><td>x333</td><td>x444</td></tr><tr><td>10</td><td>x222</td><td>x333</td></tr><tr><td>11</td><td>Reserved</td><td>Reserved</td></tr></table>	DRBT	EDO Burst Rate	Standard Page Mode Rate	00	x444	x444	01	x333	x444	10	x222	x333	11	Reserved	Reserved
DRBT	EDO Burst Rate	Standard Page Mode Rate														
00	x444	x444														
01	x333	x444														
10	x222	x333														
11	Reserved	Reserved														



Bit	Description																									
4:3	<p><b>DRAM Write Burst Timing (DWBT).</b> The DRAM write burst timings are controlled by the DWBT field. Slower rates may be required in certain system designs to support layouts with longer trace lengths or slower DRAMs. Most system designs will be able to use one of the faster burst mode timings.</p> <p><b>DWBT    Standard Page Mode Rate</b></p> <table><tr><td>00</td><td>x444</td></tr><tr><td>01</td><td>x333</td></tr><tr><td>10</td><td>x222 (see note)</td></tr><tr><td>11</td><td>Reserved</td></tr></table> <p style="text-align: center;"><b>NOTE</b></p> <p><b>Minimum 3-Clock CAS# Cycle Time for Single Writes.</b> The DWBT field controls the minimum CAS# cycle time for single and burst write cycles, except for the x222 programming case in which the minimum cycle time for <i>single writes</i> is limited to 3-clocks. Burst writes (L1/L2 writebacks and PCI burst writes within a cache line) are still completed with a 2-clock CAS# cycle time. 3-clocks is the minimum cycle time for single writes due to the minimum MA[11:2] to CAS# assertion setup time requirements. In addition, a PCI write burst crossing a cache line boundary also has a 3 clock write for the first data of the new line.</p>	00	x444	01	x333	10	x222 (see note)	11	Reserved																	
00	x444																									
01	x333																									
10	x222 (see note)																									
11	Reserved																									
2	<p><b>RAS to CAS Delay (RCD).</b> RCD controls the DRAM page miss and row miss leadoff timings. When RCD=1, the RAS active to CAS active delay is 2 clocks. When RCD=0, the timing is 3 clocks. Note that RCD timing adjustments are independent to DLT timing adjustments.</p> <p><b>RCD            RAS to CAS Delay</b></p> <table><tr><td>0</td><td>3</td></tr><tr><td>1</td><td>2</td></tr></table>	0	3	1	2																					
0	3																									
1	2																									
1:0	<p><b>DRAM Leadoff Timing (DLT).</b> The DRAM leadoff timings for page/row miss cycles are controlled by the DLT bits. DLT controls the MA setup to the first CAS# assertion.</p> <table><tr><th>DLT</th><th>Read Leadoff</th><th>Write Leadoff</th><th>RAS# Precharge</th><th>Refresh RAS# Assertion</th></tr><tr><td>00</td><td>8</td><td>6</td><td>3</td><td>4</td></tr><tr><td>01</td><td>7</td><td>5</td><td>3</td><td>4</td></tr><tr><td>10</td><td>8</td><td>6</td><td>4</td><td>5</td></tr><tr><td>11</td><td>7</td><td>5</td><td>4</td><td>5</td></tr></table> <p>Note that the DLT field and RCD bit have cumulative effects (i.e., setting DLT0=0 and RCD=0 results in two additional clocks between RAS# assertion and CAS# assertion).</p>	DLT	Read Leadoff	Write Leadoff	RAS# Precharge	Refresh RAS# Assertion	00	8	6	3	4	01	7	5	3	4	10	8	6	4	5	11	7	5	4	5
DLT	Read Leadoff	Write Leadoff	RAS# Precharge	Refresh RAS# Assertion																						
00	8	6	3	4																						
01	7	5	3	4																						
10	8	6	4	5																						
11	7	5	4	5																						

### 3.2.13. PAM—PROGRAMMABLE ATTRIBUTE MAP REGISTERS (PAM[6:0])

Address Offset: PAM0 (59h) — PAM6 (5Fh)  
 Default Value: 00h  
 Attribute: Read/Write

The MTSC allows programmable memory and cacheability attributes on 14 memory segments of various sizes in the 640-Kbyte to 1-Mbyte address range. Seven Programmable Attribute Map (PAM) Registers are used to support these features. Three bits are used to specify L1 cacheability and memory attributes for each memory segment. These attributes are:

- RE - Read Enable.** When RE=1, the CPU read accesses to the corresponding memory segment are directed to main memory. Conversely, when RE=0, the CPU read accesses are directed to PCI.
- WE - Write Enable.** When WE=1, the CPU write accesses to the corresponding memory segment are directed to main memory. Conversely, when WE=0, the CPU write accesses are directed to PCI.
- CE - Cache Enable.** When CE=1, the corresponding memory segment is L1 cacheable. CE must not be set to 1 when RE=0 for any particular memory segment. When CE=1 and WE=0, the corresponding memory segment is cached in the first level cache only on CPU code read cycles.

The RE and WE attributes permit a memory segment to be Read Only, Write Only, Read/Write, or disabled (Table 2.). For example, if a memory segment has RE=1 and WE=0, the segment is Read Only.

**Table 2. Attribute Definition**

Read/Write Attribute	Definition
Read Only	<p><b>Read cycles.</b> CPU cycles are serviced by the main memory or second level cache in a normal manner.</p> <p><b>Write cycles.</b> CPU initiated write cycles are ignored by the DRAM interface as well as the second level cache. Instead, the cycles are passed to PCI for termination.</p> <p>Areas marked as read only are L1 cacheable for code accesses only. These regions are not cached in the second level cache.</p>
Write Only	<p><b>Read cycles.</b> All read cycles are ignored by main memory as well as the second level cache. CPU-initiated read cycles are passed onto PCI for termination. The write only state can be used while copying the contents of a ROM, accessible on PCI, to main memory for shadowing, as in the case of BIOS shadowing.</p> <p><b>Write cycles:</b> CPU write cycles are serviced by main memory and L2 cache in a normal manner.</p>
Read/Write	This is the normal operating mode of main memory. Both read and write cycles from the CPU and PCI are serviced by main memory and L2 cache interface.
Disabled	All read and write cycles to this area are ignored by the main memory and cache interface. These cycles are forwarded to PCI for termination.

Each PAM Register controls two regions, typically 16 Kbytes 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 Table 3.

PCI master access to main memory space is also controlled by the PAM Registers. If the PAM programming indicates a region is writeable, then PCI master writes are accepted (DEVSEL# generated). If the PAM programming indicates a region is readable, PCI master reads are accepted. If a PCI write to a non-writeable main memory region or a PCI read of a non-readable main memory region occurs, the MTSC does not accept the cycle (DEVSEL# is not asserted). PCI master accesses to enabled memory hole regions are not accepted by the MTSC.

Table 3. Attribute Bit Assignment

Bits [7,3] Reserved	Bits [6,2] Cache Enable	Bits [5,1] Write Enable	Bits [4,0] Read Enable	Description
x	x	0	0	Main memory disabled; accesses directed to PCI.
x	0	0	1	read only; main memory write protected; non-cacheable.
x	1	0	1	read only; main memory write protected; L1 cacheable for code accesses only.
x	0	1	0	write only.
x	0	1	1	read/write; non-cacheable.
x	1	1	1	read/write; cacheable.

As an example, consider a BIOS that is implemented on the expansion bus. During the initialization process, BIOS can be shadowed in main memory to increase the system performance. To shadow BIOS, the attributes for that address range should be set to write only. BIOS is shadowed by first performing a read of that address. This read is forwarded to the expansion bus. The CPU then performs 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.

The Intel 430MX PCIs
et does not support read-modify-write cycles to memory space defined by the PAM registers as write protected. System software should not perform these cycles to areas that is write protected.

Table 4. PAM Registers and Associated Memory Segments

PAM Reg	Attribute Bits				Memory Segment	Comments	Offset
PAM0[3:0]	Reserved						59h
PAM0[7:4]	R	CE	WE	RE	0F0000–0FFFFFh	BIOS Area	59h
PAM1[3:0]	R	CE	WE	RE	0C0000–0C3FFFh	ISA Add-on BIOS	5Ah
PAM1[7:4]	R	CE	WE	RE	0C4000–0C7FFFh	ISA Add-on BIOS	5Ah
PAM2[3:0]	R	CE	WE	RE	0C8000–0CBFFFh	ISA Add-on BIOS	5Bh
PAM2[7:4]	R	CE	WE	RE	0CC000–0CFFFFh	ISA Add-on BIOS	5Bh
PAM3[3:0]	R	CE	WE	RE	0D0000–0D3FFFh	ISA Add-on BIOS	5Ch
PAM3[7:4]	R	CE	WE	RE	0D4000–0D7FFFh	ISA Add-on BIOS	5Ch
PAM4[3:0]	R	CE	WE	RE	0D8000–0DBFFFh	ISA Add-on BIOS	5Dh
PAM4[7:4]	R	CE	WE	RE	0DC000–0DFFFFh	ISA Add-on BIOS	5Dh
PAM5[3:0]	R	CE	WE	RE	0E0000–0E3FFFh	BIOS Extension	5Eh
PAM5[7:4]	R	CE	WE	RE	0E4000–0E7FFFh	BIOS Extension	5Eh
PAM6[3:0]	R	CE	WE	RE	0E8000–0EBFFFh	BIOS Extension	5Fh
PAM6[7:4]	R	CE	WE	RE	0EC000–0EFFFFh	BIOS Extension	5Fh

**NOTE:** The CE bit should not be changed while the L2 cache is enabled.

**DOS Application Area (00000–9FFFh).** Read, write, and cacheability attributes are always enabled and are not programmable for the 0–640-Kbyte DOS application region.

**Video Buffer Area (A0000–BFFFFh).** This 128-Kbyte area is not controlled by attribute bits. CPU-initiated cycles in this region are always forwarded to PCI for termination. This area is not cacheable. See section 3.2.16 for details on the use of this range as SMRAM.

**Expansion Area (C0000–DFFFFh).** This 128-Kbyte area is divided into eight 16-Kbyte segments. Each segment can be assigned one of four read/write states: read-only, write-only, read/write, or disabled memory that is disabled is not remapped. Cacheability status can also be specified for each segment.

**Extended System BIOS Area (E0000–EFFFFh).** This 64-Kbyte area is divided into four 16-Kbyte segments. Each segment can be assigned independent cacheability, read, and write attributes. Memory segments that are disabled are not remapped elsewhere.

**System BIOS Area (F0000–FFFFFh).** This area is a single 64-Kbyte segment. This segment can be assigned cacheability, read, and write attributes. When disabled, this segment is not remapped.

**Extended Memory Area (100000–FFFFFFFh).** The extended memory area can be split into several parts:

- Flash BIOS area from 4 Gbytes to 4 Gbytes - 512 Kbytes (aliased on ISA at 16 Mbytes - 15.5 Mbytes)
- Main Memory from 1 Mbyte to a maximum of 128 Mbytes
- PCI Memory space from the top of main memory to 4 Gbytes - 512 Kbytes

On power-up or reset, the CPU vectors to the Flash BIOS area, mapped in the range of 4 Gbytes to 4 Gbytes–512 Kbytes. This area is physically mapped on the expansion bus. Since these addresses are in the upper 4-Gbyte range, the request is directed to PCI. The main memory space can occupy extended memory from a minimum of 1 Mbyte up to 128 Mbytes. This memory is cacheable. PCI memory space from the top of main memory to 4 Gbytes is always non-cacheable.

### 3.2.14. DRB—DRAM ROW BOUNDARY REGISTERS

Address Offset: DRB0 (60h) — DRB3 (63h)  
Default Value: 02h  
Access: Read/Write

The MTSC supports 4 rows of DRAM. Each row is 64 bits wide. The DRB Registers define upper and lower addresses for each DRAM row. Contents of these 8-bit registers represent the boundary addresses in 4-Mbyte granularity. Note that bit 0 of each DRB must always be programmed to 0 for proper operation.

DRB0 = Total amount of memory in row 0 (in 4Mbytes)  
DRB1 = Total amount of memory in row 0 + row 1 (in 4Mbytes)  
DRB2 = Total amount of memory in row 0 + row 1 + row 2 (in 4Mbytes)  
DRB3 = Total amount of memory in row 0 + row 1 + row 2 + row 3 (in 4Mbytes)

The DRAM array can be configured with 1Mx36, 4Mx36, and 16Mx36 SIMMs. Each register defines an address range that causes a particular RAS# line to be asserted (e.g. if the first DRAM row is 8 Mbytes in size then accesses within the 0 to 8-Mbyte range causes RAS0# to be asserted).

Bit	Description
7:6	Reserved.

5:0	<b>Row Boundary Address.</b> This 6-bit field is compared against address lines A[27:22] to determine the upper address limit of a particular row (i.e., DRB minus previous DRB=row size).
-----	--

### Row Boundary Address

These 6 bit values represent the upper address limits of the 4 rows (i.e., this row minus previous row = row size). Unpopulated rows have a value equal to the previous row (row size = 0). DRB3 reflects the maximum amount of DRAM in the system. The top of memory is determined by the value written into DRB3. If DRB3 is greater than 128 Mbytes, then 128 Mbytes of DRAM are available.

### 3.2.15. DRT—DRAM ROW TYPE REGISTER

Address Offset: 68h  
Default Value: 00h  
Access: Read/Write

This 8-bit register identifies the type of DRAM (EDO or page mode) used in each row, and should be programmed by BIOS for optimum performance if EDO DRAMs are used. The MTSC uses these bits to determine the correct cycle timing on DRAM cycles.

Bit	Description
7:4	<b>Reserved.</b>
3:0	<b>DRAM Row Type (DRT[3:0]).</b> Each bit in this field corresponds to the DRAM row identified by the corresponding DRB Register. Thus, DRT0 corresponds to row 0, DRT1 to row 1, etc. When DRTx=0, page mode DRAM timings are used for that bank. When DRTx=1, EDO DRAM timings are used for that bank.

### 3.2.16. SMRAM—SYSTEM MANAGEMENT RAM CONTROL REGISTER

Address Offset: 72h  
Default Value: 02h  
Access: Read/Write

The System Management RAM Control Register controls how accesses to this space are treated. The Open, Close, and Lock SMRAM Space bits function only when the SMRAM enable bit is set to a 1. Also, the OPEN bit (DOPEN) should be set to 0 before the LOCK bit (DLCK) is set to 1.

Bit	Description
7	<b>Reserved.</b>
6	<b>SMM Space Open (DOPEN).</b> When DOPEN=1 and DLCK=0, SMM space DRAM is made visible, even when SMIACK# is negated. This is intended to help BIOS initialize SMM space. Software should ensure that DOPEN=1 is mutually exclusive with DCLS=1. When DLCK is set to a 1, DOPEN is set to 0 and becomes read only.
5	<b>SMM Space Closed (DCLS).</b> When DCLS=1, SMM space DRAM is not accessible to data references, even if SMIACK# is asserted. Code references may still access SMM space DRAM. This allows SMM software to reference "through" SMM space to update the display, even when SMM space is mapped over the VGA range. Software should ensure that DOPEN=1 is mutually exclusive with DCLS=1.

Bit	Description
4	<b>SMM Space Locked (DLCK).</b> When DLCK is set to 1, the MTSC sets DOPEN to 0 and both DLCK and DOPEN become read only. DLCK can be set to 1 via a normal configuration space write but can only be cleared by a power-on reset. The combination of DLCK and DOPEN provide convenience with security. The BIOS can use the DOPEN function to initialize SMM space and then use DLCK 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 DOPEN function.
3	<b>SMRAM Enable (SMRAM).</b> When SMRAM=1, the SMRAM function is enabled, providing 128 Kbytes of DRAM accessible at the address defined by bits [2:0] while in SMM (ADS# with SMIACK#).
2:0	<b>SMM Space Base Segment (DBASESEG).</b> This field selects 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 PCI. DBASESEG=010 selects the SMM space as A0000–BFFFFh. DBASESEG=100 selects the SMM space as C0000–CFFFFh. All other values are reserved. PCI masters are not allowed access to SMM space. If the C segment is selected as SMRAM, the two PAM registers associated with the segment must be programmed for read enable, write enable, and cache disable.

Table 5 summarizes the operation of SMRAM space cycles targeting SMI space addresses (A segment):

**Table 5. SMRAM Space Cycles**

SMRAM	DLCK	DCLS	DOPEN	SMIACK#	Code Fetch	Data Reference
0	X	X	X	X	PCI	PCI
1	0	0	0	0	DRAM	DRAM
1	0	X	0	1	PCI	PCI
1	0	0	1	X	DRAM	DRAM
1	0	1	0	0	DRAM	PCI
1	0	1	1	X	INVALID	INVALID
1	1	0	X	0	DRAM	DRAM
1	1	X	X	1	PCI	PCI
1	1	1	X	0	DRAM	PCI

## 4.0. FUNCTIONAL DESCRIPTION

This section provides a functional description of the MTSC and MTD.

### 4.1. Host Interface

The Host Interface of the MTSC is designed to support the Pentium® processor at 120, 90, and 75 MHz. The host interface of the MTSC supports 50, 60, and 66 MHz bus speeds. The Intel 430MX PCIset supports the Pentium processor with a full 64-bit data bus, 32-bit address bus, and associated internal write-back cache logic. Host bus addresses are decoded by the MTSC for accesses to main memory, PCI memory, and PCI I/O. The MTSC also supports the pipelined addressing capability of the Pentium processor.

When the Pentium processor initiates a special cycle, the cycle definition pins will be: M/IO# = 0, D/C# = 0, and W/R# = 1. The action taken by the MTSC is shown in Table 6.

The Pentium processor generates interrupt acknowledge cycles in response to maskable interrupt requests reported on the INTR pin. The MTSC generates a PCI Interrupt Acknowledge in response to a host Interrupt Acknowledge.

**Table 6. Special Bus Cycle Action Taken**

Special Bus Cycle	Action Taken
Shutdown	Propagated to PCI as a Special Shutdown cycle, not posted, MPIIX generates INIT to CPU. INIT asserted for a minimum of 16 HCLKs.  Cycle on PCI terminates as Master Abort and then BRDY is returned to the CPU.
Halt	Propagated to PCI as Special Halt cycle, not posted. Cycle on PCI terminates as Master Abort and then BRDY is returned to the CPU. Enter Power Down Mode for Second level cache.
STOP GRANT	Propagated to PCI as a Special Stop Grant cycle, with 0002h in the message field [AD(15:0)] and 0012h in the message dependent data field [AD(31:16)] during the first data phase (IRDY# asserted). Cycle on PCI terminates as Master Abort and then BRDY is returned to the CPU. Enter Power Down Mode for Second level cache.
Flush (INVD, WBINVD), Write Back (WBINVD), Flush Acknowledge (FLUSH# assertion), and Branch Trace Message	Return BRDY# to CPU.

### 4.2. PCI Interface

The 82437MX integrates a high performance interface to the PCI local bus taking full advantage of the high bandwidth and low latency of PCI. Four PCI masters are supported by the integrated arbiter including a PCI-to-ISA bridge and three general PCI masters. The MTSC acts as a PCI master for CPU accesses to PCI. The PCI bus is clocked at one half the frequency of the CPU clock. This divided synchronous interface minimizes latency for CPU-to-PCI cycles and PCI-to-main memory cycles.

The MTSC/MTDPs integrate posted write buffers for CPU memory writes to PCI. Back-to-back sequential memory writes to PCI are converted to burst writes on PCI. This feature allows the CPU to continue posting

dword writes at the maximum bandwidth for the Pentium processor for the highest possible transfer rates to the graphics frame buffer.

Read prefetch and write posting buffers in the MTSC/MTDPs enable PCI masters to access main memory at up to 120 MB/s. The MTSC incorporates a snoop-ahead feature which allows PCI masters to continue bursting on both reads and writes even as the bursts cross cache line boundaries.

### 4.3. Secondary Cache Interface

The MTSC integrates a high performance second level cache controller using internal/external tags and provides a full first level and second level cache coherency mechanism. The second level cache is direct mapped, non-sectored, and supports a write-back cache policy. Cache lines are allocated on read misses (no write allocate).

The second level cache can be configured for either 256- or 512-Kbyte cache sizes using either synchronous burst or pipelined burst SRAMs, or standard asynchronous SRAMs. For the 256-Kbyte configurations, an 8kx8 standard SRAM is used to store the tags. For the 512-Kbyte configurations, a 16kx8 standard SRAM is used to store the tags and the valid bits.

A second level cache line is 32 bytes wide. In the 256-Kbyte configurations, the second level cache contains 8K lines, while the 512-Kbyte configurations contain 16K lines. Valid and modified status bits are kept on a per-line basis. Cacheability of the entire memory space in the first level cache is supported. For the second level cache, only the lower 64 Mbytes of main memory are cacheable (only main memory controlled by the MTSC DRAM interface is cached). PCI memory is not cached. Table 7 shows the different standard SRAM access time requirements for different host clock frequencies.

**Table 7. SRAM Access Time Requirements**

Host Clock Frequency (MHz)	Standard SRAM Access Time (ns)	Burst SRAM Clock-to-Output Access Time (ns)	Tag RAM Access Time (ns)	Tag Burst SRAM Access Time (ns)
50	20 (17 ns buffer)	13.5	30	20
60	17 (10 ns buffer)	10	20	15
66	15 (7 ns buffer)	8.5	15	15

#### 4.3.1. CLOCK LATENCIES

Table 8 and Table 9 list the latencies for various processor transfers to and from the second level cache for standard and burst SRAM. The clock counts are identical for pipelined and non-pipelined burst SRAM.

**Table 8. Second Level Cache Latencies with Standard SRAM**

Cycle Type	HCLK Count
Burst Read	3-2-2-2
Burst Write (write back)	4-3-3-3
Single Read	3
Single Write	4
Back-to-Back Burst Reads	3-2-2-2, 3-2-2-2 (note)

**NOTE:** The back-to-back cycles do not account for CPU idle clocks between cycles.



**Table 9. Second Level Cache Latencies with Burst SRAM**

Cycle Type	HCLK Count
Burst Read	3-1-1-1
Burst Write (write back)	3-1-1-1
Single Read	3
Single Write	3
Pipelined Back-to-Back Burst Reads	3-1-1-1,1-1-1-1 (note)
Non-Pipelined Back-to-Back Burst Reads (i.e., no NA#)	3-1-1-1,3-1-1-1 (note)

**NOTE:** The back-to-back cycles do not account for CPU idle clocks between cycles.

#### 4.3.2. SNOOP CYCLES

Snoop cycles are used to maintain coherency between the caches (first and second level) and main memory. The MTSC generates a snoop (or inquire) cycle to probe the first level and second level caches when a PCI master attempts to access main memory. Snoop cycles are performed by driving the PCI master address onto the host address bus and asserting EADS#.

To maintain optimum PCI bandwidth to main memory, the MTSC utilizes a "snoop ahead" algorithm. Once the snoop for the first cache line of a transfer has completed, the MTSC automatically snoops the next sequential cache line. This algorithm enables the MTSC to continue burst transfers across cache line boundaries.

#### NOTE

The conditions for generating snoop cycles to the CPU is identical to the conditions for DEVSEL# generation, based on the PAM programming. Snoops are prevented for Reads to non-readable regions, Writes to non-writeable regions, Reads/writes to enabled memory holes, Reads/writes above top of memory.

#### Reads

If the snoop cycle generates a first level cache hit to a modified line, the line in the first level cache is written back to main memory (via the DRAM Posted Write Buffers). The line in the second level cache (if it exists) is invalidated. Note that the line in the first level cache is not invalidated if the INV pin on the CPU is tied to the KEN# signal from the MTSC. The MTSC drives KEN#/INV low with EADS# assertion during PCI master read cycles.

At the same time as the first level snoop cycle, the MTSC performs a tag look-up to determine whether the addressed memory is in the second level cache. If the snoop cycle generates a second level cache hit to a modified line and there was not a hit in the first level cache (HITM# not asserted), the second level cache line is written back to main memory (via the DRAM Posted Write Buffers) and changed to the "clean" state. The PCI master read completes after the data has been written back to main memory.

#### Writes

PCI Master write cycles never result in a write directly into the second level cache. A snoop hit to a modified line in either the first or second caches results in a write-back of that line to main memory. If both the first and second level caches have modified lines, the line is written back from the first level cache. In all cases, lines in the first and second level caches are invalidated and the PCI write to main memory occurs after the writeback completes. A PCI master write snoop hit to an unmodified line in either the first or second level caches results in the line being invalidated. The MTSC drives KEN#/INV with EADS# assertion during PCI master write cycles.

#### 4.3.3. SRAM POWER DOWN (DE-SELECT) MODE SEQUENCE

The MTSC places the BSRAM in a “de-select” mode when the CPU issues a HALT or Stop Grant Bus cycle. The de-select sequence is shown in Figure 2.

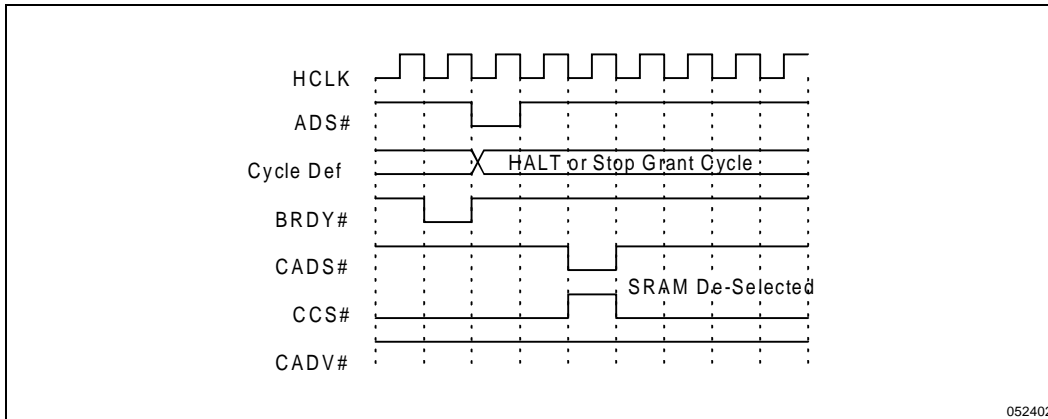


Figure 2. BSRAM De-Select Sequence

#### 4.3.4. FLUSHING L2 CACHE

The Force Miss/Invalidate mode (Selected in the Cache Control register) is intended to allow flushing of modified L2 lines back to memory. Since the flushed modified line is posted to main memory write buffer in parallel with the linefill, the new line (which is filled to both L1 and L2) is guaranteed to be stale data.

The implication of the actual operation is that any program that is attempting to flush the L2 must not be run from the L2 itself, unless the L2 code image is coherent with main memory.

For example, if a DOS program to flush the L2 is loaded, it is loaded from disk using a string move. The MOVSB may result in loading the program partially into L2, since the memory writes will (likely) hit previously valid lines in the L2. When the flush program sets the SCFMI bit to force misses, subsequent code fetches may be serviced from L2 which contain the stale code image.

An important aspect to consider when flushing L2 is to ensure that the code being used to flush the cache is not in the current 256K page (256K L2 being used here) residing in L2. The BIOS algorithm will also work if it is executed from non-cacheable or write-protected DRAM space (e.g. BIOS region). If this can't be guaranteed, then an alternate software flush algorithm is to read 2X the size of the L2. Note that this implies a protected-mode code for the 512K L2 case.

#### 4.3.5. CACHE ORGANIZATION

Figures 3, 4, 5, 6, and 7 show the connections between the MTSC and the external tag RAM and data SRAM. A 512K standard SRAM cache is implemented with 64Kx8 data SRAMs and a 16Kx8 tag RAM. The second ADS# pin from the CPU should be used to drive the ADSP# pin on Burst or Pipelined Burst SRAMs.

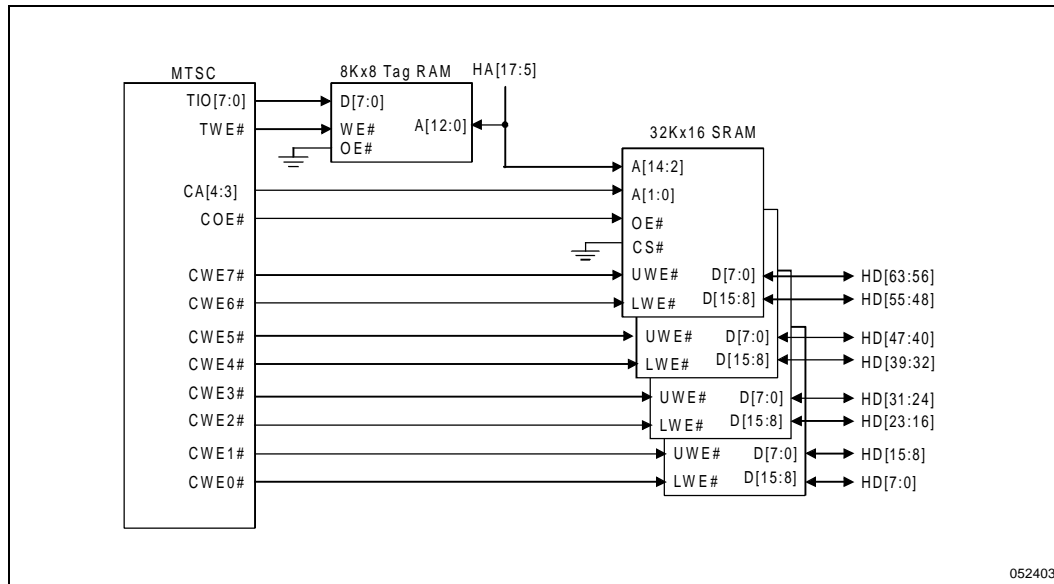


Figure 3. 256-Kbyte Second Level Cache (standard SRAM)

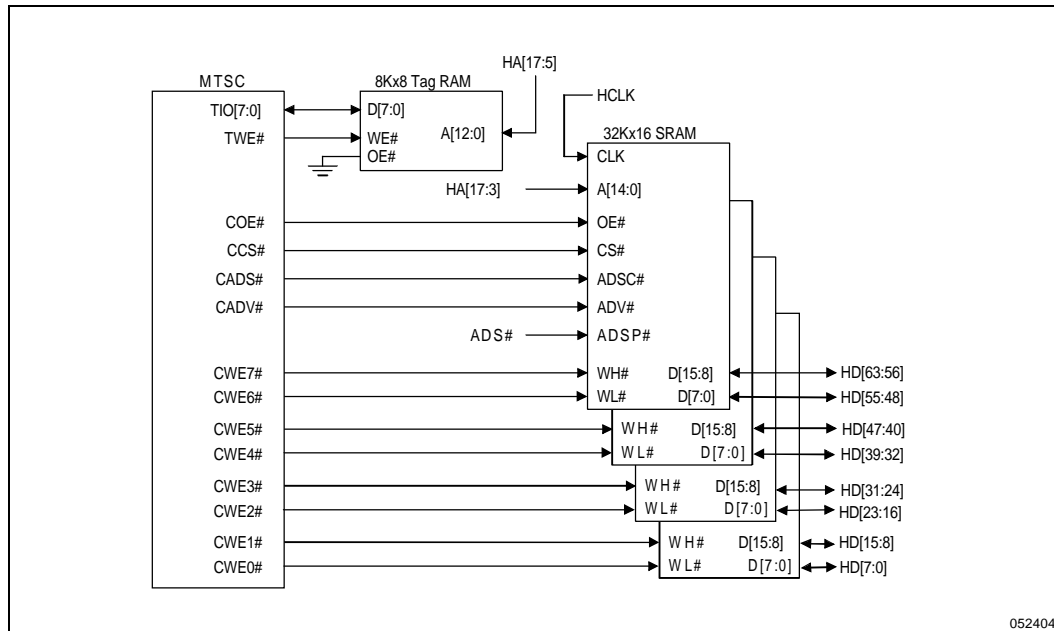


Figure 4. 256-Kbyte Second Level Cache (burst SRAM)

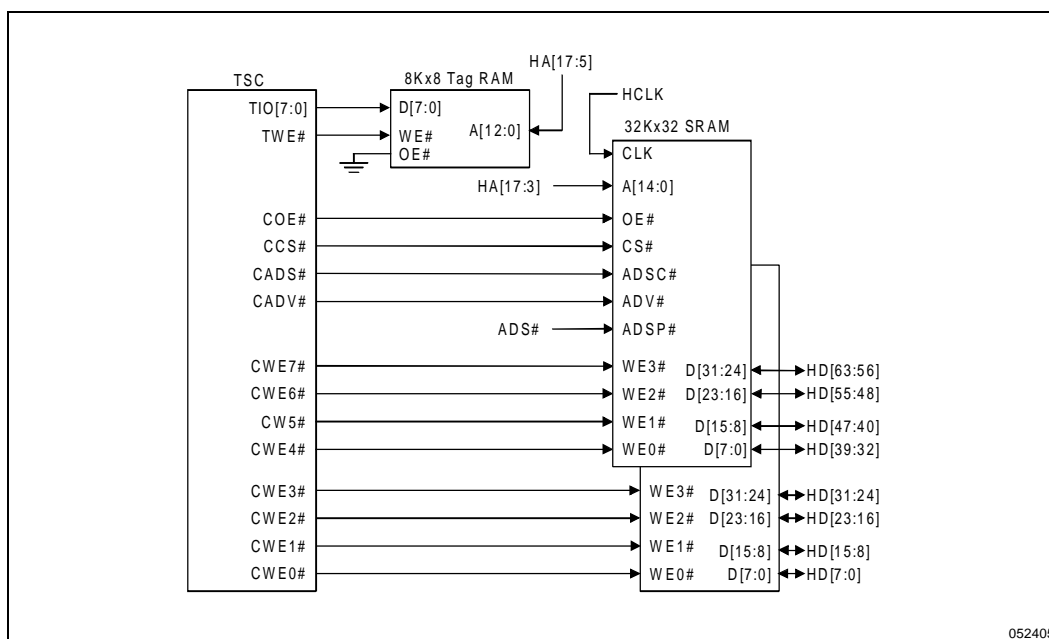


Figure 5. 256-Kbyte Second Level Cache (Burst SRAM)

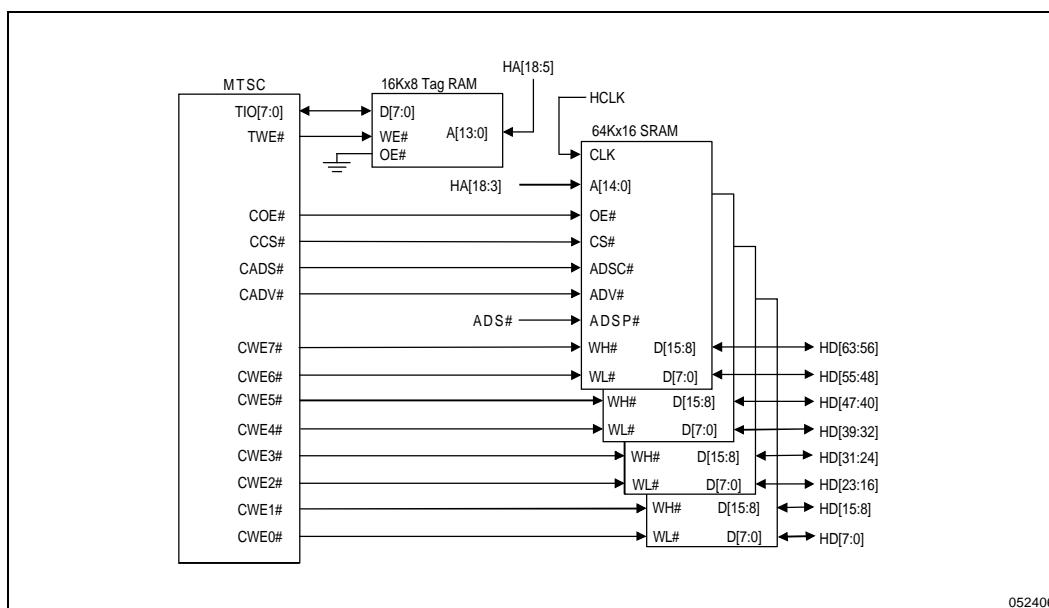


Figure 6. 512-Kbyte Second Level Cache (Burst SRAM)

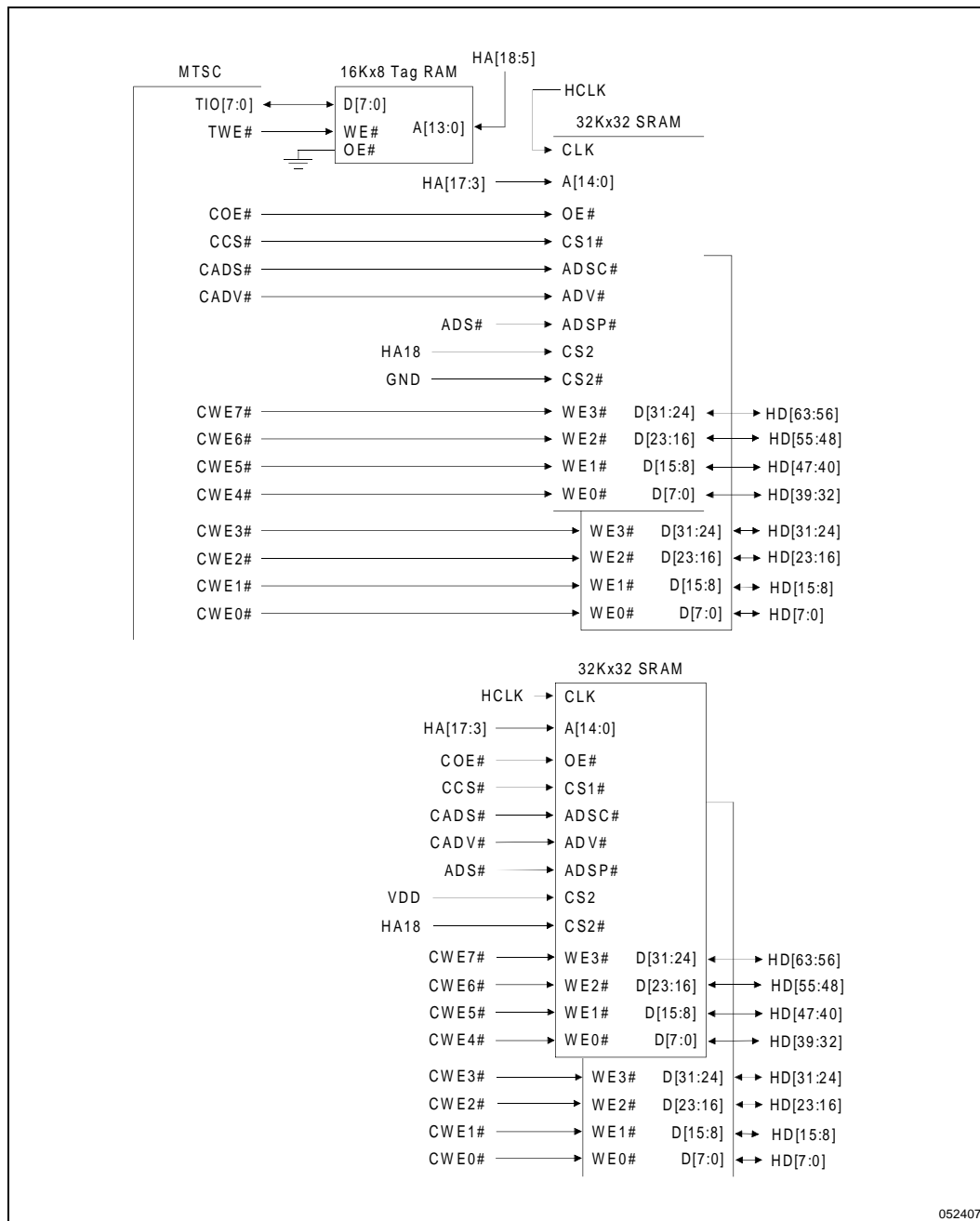


Figure 7. 2-Bank 512-Kbyte Second Level Cache (Pipelined Burst SRAM)

#### 4.4. DRAM Interface

The Intel 430MX PCIset's main memory DRAM interface supports a 64-bit wide memory array and main memory sizes from 4 to 128 Mbytes. The MTSC generates the RAS#, CAS#, WE# and multiplexed addresses for the DRAM array and controls the data flow through the 82438MX MTD. For CPU-to-DRAM cycles the address flows through the MTSC and data flows through the MTD. For PCI or DMA cycles to memory, the address flows through the MTSC and data flows to the MTD through the MTSC and PLINK bus. The MTSC and MTD DRAM interfaces are synchronous to the CPU clock.

The Intel 430MX PCIset supports industry standard 32-bit wide memory modules with fast page-mode DRAMs and EDO (Extended Data Out) DRAMs (also known as Hyper Page mode). With twelve multiplexed address lines (MA[11:0]), the MTSC supports 512Kx32, 1M32, 2Mx32, and 4Mx32 SIMM's (both symmetrical and asymmetrical addressing). Four RAS# lines permit up to four rows of DRAM and eight CAS# lines provide byte control over the array. The MTSC supports 60 and 70 ns DRAMs (both single and double-sided SIMM's). The MTSC also provides a CAS#-before-RAS refresh at programmable refresh rates from 15.6  $\mu$ s to 250  $\mu$ s. A refresh priority queue and "smart refresh" algorithm are used to minimize the performance impact due to refresh.

The DRAM controller interface is fully configurable through a set of control registers (see Register Description section for programming details). The DRAM interface is configured by the DRAM Control Register, the four DRAM Row Boundary (DRB) Registers, and the DRAM Row Type (DRT) Register. The DRAM Control and DRAM Timing Registers configure the DRAM interface to select fast page-mode or EDO DRAMs, RAS timings, and CAS rates. The four DRB Registers define the size of each row in the memory array, enabling the MTSC to assert the proper RAS# line for accesses to the array.

Seven Programmable Attribute Map (PAM) Registers are used to specify the cacheability, PCI enable, and read/write status of the memory space between 640 Kbytes and 1 Mbyte. Each PAM Register defines a specific address area enabling the system to selectively mark specific memory ranges as cacheable, read-only, write-only, read/write, or disabled. When a memory range is disabled, all CPU accesses to that range are forwarded to PCI.

The MTSC also supports one of two memory holes; either from 512 Kbytes–640 Kbytes or from 15 Mbytes–16 Mbytes in main memory. Accesses to the memory holes are forwarded to PCI. The memory hole can be enabled/disabled through the DRAM Control Register. All other memory from 1 Mbyte to the top of main memory is read/write and cacheable.

The SMRAM memory space is controlled by the SMRAM Control Register. This register selects if the SMRAM space is enabled, opened, closed, or locked. SMRAM space is between 640 Kbytes and 768 Kbytes.

The MA[11:2] buffer strengths can be programmed via the DRAM Timing Register (PCI offset 58h) to 8mA to reduce the need for external buffers on the DRAM address bus. They default to 2mA for backwards compatibility. MA[1:0] are always driven to 8mA. The MA[11:0] should have series damping resistors of approximately 22 ohms (actual values dependent on specific layout). Simulations show that the 8mA buffers can drive the 24 loads required for a 48-Mbyte DRAM configuration (dependent on actual layout and connector configuration).

**Table 10. MA[11:2] Buffer Strength Programming**

Description	Amount	Config	Devices	Loads		
				MA[11:0], WE#	RAS#	CAS#
Base (RAS0#)	8M	1Mx16	4	4	4	1
Base (RAS1#)	8M	1Mx16	4	4	4	1
Upgrade (RAS2#)	32M	4Mx4	16	16	16	2
Total	<b>48M</b>		24	<b>24</b>	<b>16 (max)</b>	<b>4 (max)</b>

#### 4.4.1. DRAM ORGANIZATION

Figure 8 illustrates a 4-bank configuration. Among the 4 banks, the DRAM densities can be mixed in any order. Each row is controlled by up to 8 CAS lines. EDO and Standard page mode DRAM's can mixed between rows; however, a given row must contain only one type of DRAM. When DRAM types are mixed (EDO and standard page mode), each row will run optimized for that particular type of DRAM. Buffers on the address and WE# lines are not necessary if 3 or fewer banks are used.

The following rules apply to DRAM bank configuration.

- Bank 0 is the default bank to be used for the mother board DRAM. Bank 0 is always refreshed. (Other banks are only refreshed if they are populated.)
- Banks can be populated in any order (i.e., Bank 1 does not have to be populated before Bank 2/3)
- Banks can be paired. Rows 2/3 are paired as an example for use with single- and double- sided SIMMs.
- Bank **pairs** need to be populated with the same densities. For example, Bank 2/3 should be populated with identical densities. However Bank 0 and 1 can be populated with different densities than Bank pair 2/3.
- The MTSC only recognizes a maximum of 128 MB of DRAM, even if populated with more memory.
- EDO's and standard page mode can both be used. However, only one type should be used per Bank pair. For example, in Figure 8 Banks 2/3 can be populated with EDO's while Banks 0 and 1 can be populated with standard page mode.

#### 4.4.2. MAIN MEMORY ADDRESS MAP

The main memory organization (Figure 9) represents the maximum 128 Mbytes of address space. Accesses to memory space above the top of main memory, video buffer range, or the memory gaps (if enabled) are not cacheable and are forwarded to PCI. Below 1 Mbyte, there are several memory segments with selectable cacheability. The DRAM spaces occupied by the video buffer and memory space gaps are not remapped.

#### 4.4.3. DRAM ADDRESS TRANSLATION

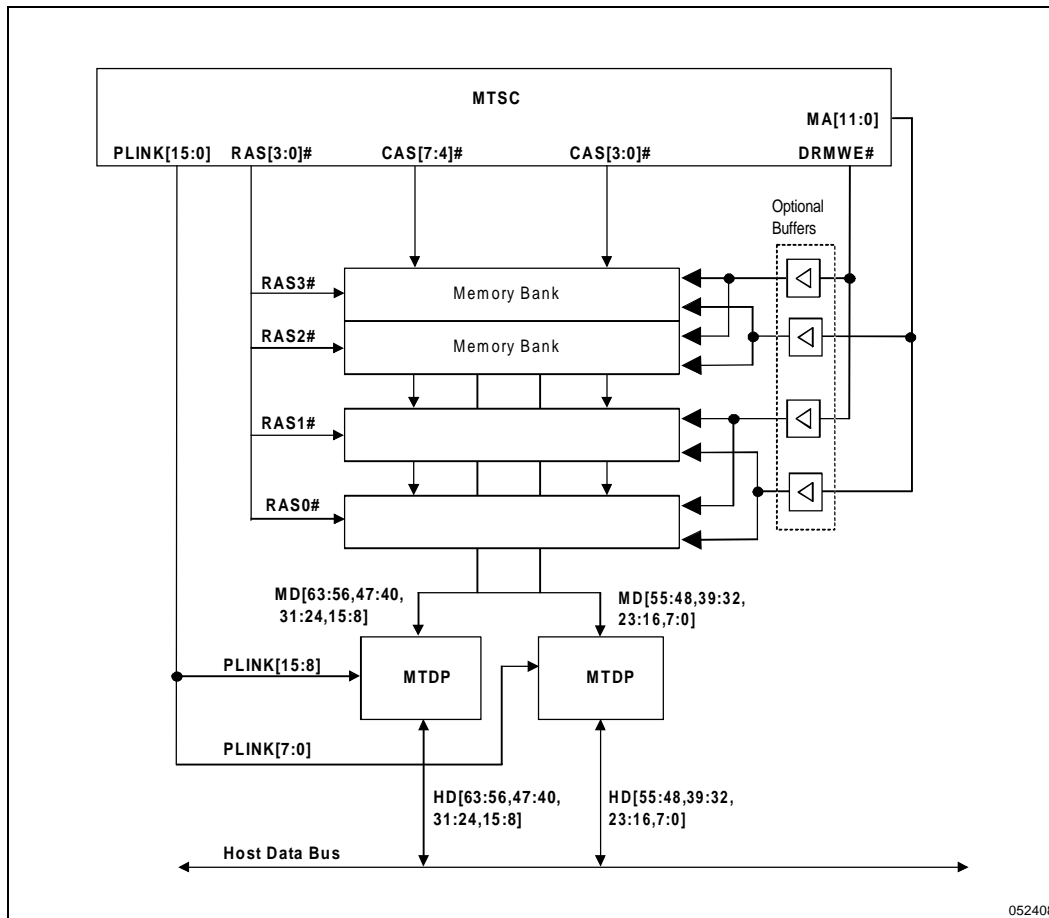
The multiplexed row/column address to the DRAM memory array is provided by MA[11:0] which are derived from the host address bus or PCI address as defined by Table 11. The MTSC has a 4-Kbyte page size. The page offset address is driven on the MA[8:0] lines when driving the column address. The MA[11:0] lines are translated from the address lines A[24:3] for all memory accesses.

**Table 11. DRAM Address Translation**

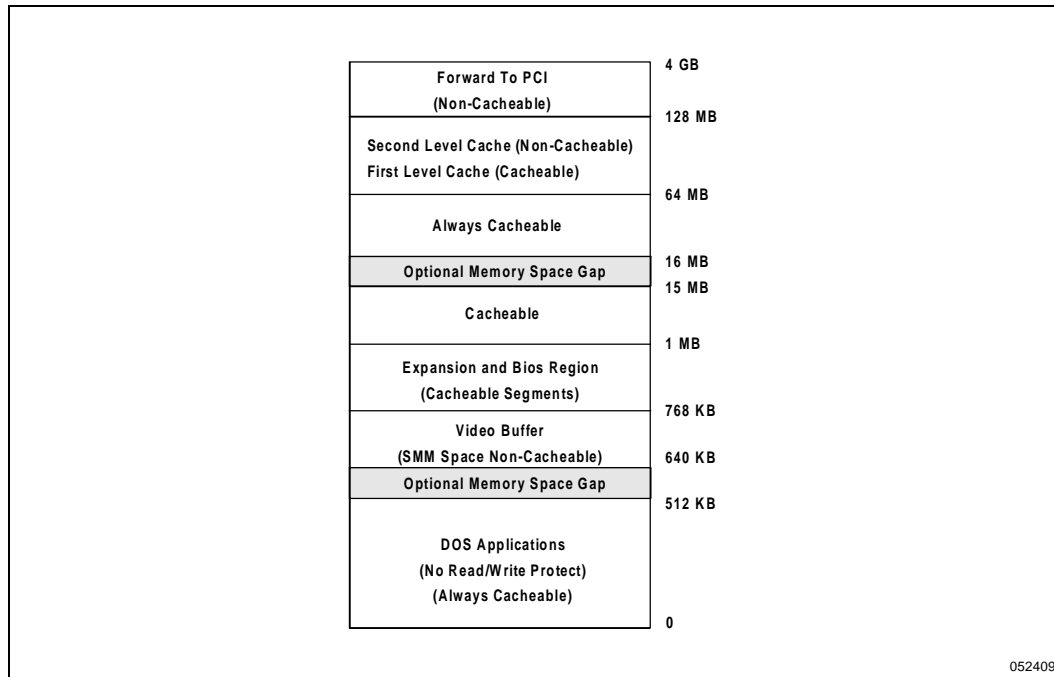
Memory Address, MA[11:0]	11	10	9	8	7	6	5	4	3	2	1	0
Row Address	A24	A23	A21	A20	A19	A18	A17	A16	A15	A14	A13	A12
Column Address	X	A24	A22	A11	A10	A9	A8	A7	A6	A5	A4	A3

The types of DRAMs depth configuration supported are:

Depth	Row Width	Column Width	Mbyte/Bank
512Kx8	10	9	4
1Mx8	10	10	8
1Mx16	10	10	8
2Mx8	11	10	16
4Mx8	11	11	32
4Mx8	12	10	32







**Figure 9. Memory Space Organization**

#### 4.4.4. DRAM PAGE MODE

The MTSC keeps the RAS lines active after a DRAM access for faster page mode accesses. The RAS remains active after the access for two clocks. The next access is then considered a page hit if the access is seen within the 2-clock window and is to the same page. The MTSC has a 4 KB page size. Page mode is always active.

If a new CPU cycle is not seen soon enough by the DRAM controller, the DRAM page will be closed. For reads, a new ADS# must occur no later than 1 HCLK prior to the last CAS# precharge in a burst linefill. For writes, if the write buffer is non-empty and the subsequent next entry is a page-hit, the page is kept open. If the buffer empties, or when following a read cycle, a new MSTB# to post write data must occur no later than the last CAS# precharge to keep the page open.

Note that, if pipelining is not enabled [e.g. asynchronous L2 enabled], paging is effectively disabled on back-to-back reads. However, reads following writes within the required window will keep the page open, as will writes occurring within the required write window. Also, if L2 cache is disabled, pipelining will still be enabled, unless asynchronous L2 has been selected.

DRAM specifications include maximum RAS# active time during fast page mode accesses. The DRAM Extended Refresh Rate (DRR) bits located in the DRAM Control (DRAMC) register (offset 57h) should be programmed to ensure that 4 refresh requests occur within this DRAM maximum RAS# active time specification. The fast page mode accesses will be broken when the fourth refresh request is queued. This ensures the negation of RAS# within the required DRAM specification.

For example:

DRAM Maximum RAS# Pulse Width (fast page mode) = 100  $\mu$ s

DRAMC DRR bits [2:0] programmed to 000 = 15.6  $\mu$ s Refresh Rate

This ensures that 4 refresh requests are generated and queued, page mode broken, and RAS# lines negated within the 100  $\mu$ s Maximum RAS# Pulse Width specification of the DRAM.

#### 4.4.5. EDO MODE

Extended Data Out (or Hyper Page Mode) DRAM is designed to improve the DRAM read performance. EDO DRAM holds the memory data valid until the next CAS# falling edge. Note that standard page mode DRAM tristates the memory data when CAS# negates to precharge. With EDO, the CAS# precharge overlaps the memory data valid time. This allows CAS# to negate earlier while still satisfying the memory data valid window time.

The EDO Detect Mode Enable bit in the DRAM Control Register enables a special timing mode for BIOS to detect the DRAM type on a row by row basis. This information must then be programmed into the DRAM row type for optimal performance. To exploit the performance improvements from EDO DRAMs, the BIOS should provide for dynamic detection of any EDO DRAMs in the DRAM rows.

An example algorithm is provided below with the following assumptions:

- The first and second level caches are disabled.
- Shadowing is not enabled.
- Memory sizing has been performed and the correct memory size has been programmed into DRB[4:0].
- The refresh rate is enabled in the DRAMC Register (bits [2:0]) are set to either 001, 010, or 011).
- The DRAM timings in the DRAMT Register are set at their fail-safe values.
- RAS to CAS Delay is 3 clocks (bit 2 is 0 in the DRAMT Register).
- DRAM Leadoff Timing is 8 clocks for read and 6 clocks for writes (bits 1:0 are 00 in the DRAMT Register).

The algorithm below dynamically detects if EDO DRAMs are installed in the system.

1. Program the DRT Register (offset 68h) to 0Fh. This sets up each row for EDO mode.
2. Write FF...FFh to a location in each row of the DRAM array. The write should cross a Dword boundary in the middle of an aligned Qword.
3. Set the EDO Detect Mode Enable bit (bit 3) in the DRAMC Register (offset 57h) to 1. This puts the MTSC into a special timing mode that allows BIOS to detect whether EDO or page mode SIMMs are installed.
4. Perform a read from each of the locations that were previously written. If FF...FFh is read back, the row contains two EDO SIMMs. If anything else is read, the row contains at least one page mode SIMM. Only rows containing two EDO SIMMs can be programmed for EDO operation. **Do not yet write to the DRT Register.**
5. Set the EDO Detect Mode bit in the DRAMC Register to 0.
6. Program the DRT Register to the appropriate DRAM type for each row. Note that the DRT Register must be written only after all rows have been read.

#### 4.4.6. DRAM PERFORMANCE

The DRAM performance is controlled by the DRAM Timing Register, processor pipelining, and by the type of DRAM used (EDO or standard page mode). Table 12 lists both EDO and standard page mode optimum timings.

**Table 12. CPU to DRAM Performance Summary**

Processor Cycle Type (pipelined)	Burst SRAM Clock Count (ADS# to BRDY#)	Async SRAM Clock Count (ADS# to BRDY#)	Comments
Burst Read Page Hit	7-2-2-2	10-2-2-2	EDO
Read Row Miss	9-2-2-2 (note 1)	10-2-2-2	EDO
Read Page Miss	12-2-2-2	10-2-2-2	EDO
Back-to-Back Burst Reads Page Hit	7-2-2-2-3-2-2-2	10-2-2-2-3-2-2-2	EDO
Burst Read Page Hit	7-3-3-3	10-3-3-3	Standard page mode
Burst Read Row Miss	9-3-3-3 (note 1)	10-3-3-3	Standard page mode
Burst Read Page Miss	12-3-3-3	10-3-3-3	Standard page mode
Back-to-Back Burst Read Page Hit	7-3-3-3-3-3-3-3	10-3-3-3-3-3-3...	Standard page mode
Write Page Hit <sup>4,5</sup>	2 (note 6)		EDO/Standard page mode
Write Row Miss <sup>4,5</sup>	3 (note 2), 4 (note 3)		EDO/Standard page mode
Write Page Miss <sup>4,5</sup>	6 (note 2), 7 (note 3)		EDO/Standard page mode
Posted Write <sup>4,5</sup>	3-1-1-1	4-1-1-1	EDO/Standard page mode
Write retire rate from Posted Write Buffer	-3-3-3	-3-3-3	

**NOTES:**

1. The above row miss cycles assume that the new page is closed from the prior cycle. Due to the MA[11:2] to RAS# setup requirements, if the page is open, 2 clocks are added to the leadoff.
2. This cycle timing assumes the write buffer is not empty.
3. This cycle timing assumes the write buffer is empty.
4. Write timing is measured from the point where data is driven on the TDP's output up to CAS# assertion for that cycle. In other words, this count is the number of cycles the data remains valid at the TDP output buffer.
5. Write data is always posted as 3-1-1-1 (ADS# to BRDY#, if buffer is empty) with burst or pipelined burst second level cache SRAMs (or cacheless operation). For Asynchronous SRAMs, write data is posted as 4-1-1-1.
6. 2 clocks only for bursts within a cache line. Otherwise, the minimum number is 3 clocks.

Table 13. 60ns DRAM Timing Considerations

Memory Speed	60 ns		
Frequency (MHz)	66	60	50
<b>DRBT bits [6:5]</b>			
00 = x444 EDO, x444 SPM 01 = x333 EDO, x444 SPM 10 = x222 EDO, x333 SPM 11 = Reserved	x222(EDO),x333(SPM)	x222(EDO),x333(SPM)	x222(EDO),x333(SPM)
<b>DWBT bits [4:3]</b>			
00 = x444 01 = x333 10 = x222 11 = Reserved	x333	x333	x333
<b>RAS to CAS bit [2]</b>			
0 = 3 clocks 1 = 2 clocks	3	3	2
<b>DRAM Leadoff bits [1:0]</b>			
00 = 8 RD, 6 WR, 3 RAS#Pre 01 = 7 RD, 5 WR, 3 RAS#Pre 10 = 8 RD, 6 WR, 4 RAS#Pre 11 = 7 RD, 5 WR, 4 RAS#Pre	7 RD, 5 WR, 3 RAS Pre	7 RD, 5 WR, 3 RAS Pre	7 RD, 5 WR, 3 RAS Pre
<b>DRAM Timing Summary</b>			
Read Page Hit/EDO	7-2-2-2	7-2-2-2	7-2-2-2
Read Page Hit/SPM	7-3-3-3	7-3-3-3	7-3-3-3
Read Row Miss/EDO	10-2-2-2	10-2-2-2	9-2-2-2
Read Row Miss/SPM	10-3-3-3	10-3-3-3	9-3-3-3
Read Page Miss/EDO	13-2-2-2	13-2-2-2	12-2-2-2
Read Page Miss/SPM	13-3-3-3	13-3-3-3	12-3-3-3
Write Page Hit	Retire @ 3 clocks	Retire @ 3 clocks	Retire @ 3 clocks
Write Page Miss	Retire @ 3 clocks	Retire @ 3 clocks	Retire @ 3 clocks
Write Row Miss	Retire @ 3 clocks	Retire @ 3 clocks	Retire @ 3 clocks

Table 14. 70ns DRAM Timing Considerations

Memory Speed	70 ns		
Frequency (MHz)	66	60	50
<b>DRBT bits [6:5]</b>			
00 = x444 EDO, x444 SPM 01 = x333 EDO, x444 SPM 10 = x222 EDO, x333 SPM 11 = Reserved	x333(EDO),x444(SPM)	x222(EDO),x333(SPM)	x222(EDO),x333(SPM)
<b>DWBT bits [4:3]</b>			
00 = x444 01 = x333 10 = x222 11 = Reserved	x333	x333	x333
<b>RAS to CAS bit [2]</b>			
0 = 3 clocks 1 = 2 clocks	3	3	3
<b>DRAM Leadoff bits [1:0]</b>			
00 = 8 RD , 6 WR, 3 RAS#Pre 01 = 7 RD, 5 WR, 3 RAS#Pre 10 = 8 RD, 6 WR, 4 RAS#Pre 11 = 7 RD, 5 WR, 4 RAS#Pre	8 RD, 6 WR, 4 RAS Pre	8 RD, 6 WR, 4 RAS Pre	7 RD, 5 WR, 3 RAS Pre
<b>DRAM Timing Summary</b>			
Read Page Hit/EDO	7-3-3-3	7-2-2-2	7-2-2-2
Read Page Hit/SPM	7-4-4-4	7-3-3-3	7-3-3-3
Read Row Miss/EDO	11-3-3-3	11-2-2-2	10-2-2-2
Read Row Miss/SPM	11-4-4-4	11-3-3-3	10-3-3-3
Read Page Miss/EDO	15-3-3-3	15-2-2-2	13-2-2-2
Read Page Miss/SPM	15-4-4-4	15-3-3-3	13-3-3-3
Write Page Hit	Retire @ 3 clocks	Retire @ 3 clocks	Retire @ 3 clocks
Write Page Miss	Retire @ 3 clocks	Retire @ 3 clocks	Retire @ 3 clocks
Write Row Miss	Retire @ 3 clocks	Retire @ 3 clocks	Retire @ 3 clocks

#### 4.4.7. DRAM REFRESH

The MTSC supports CAS-before-RAS# refresh. A CAS-before-RAS cycle causes a counter within the DRAM to increment its row address. This method saves power since MTSC does not have to generate and drive the row address.

The rate that requests are generated is determined by the DRAM Control Register. When a refresh request is generated, the request is placed in a four entry queue. The DRAM controller services a refresh request when the

refresh queue is not empty and the controller has no other requests pending. When the refresh queue is full, refresh becomes the highest priority request and is serviced next by the DRAM controller, regardless of other pending requests. When the DRAM controller begins to service a refresh request, the request is removed from the refresh queue.

There is also a "smart refresh" algorithm implemented in the refresh controller. Except for Bank 0, refresh is only performed on banks that are populated. For bank 0, refresh is always performed. If only one bank is populated, using bank 0 will result in better performance.

#### NOTE

If a high priority refresh is queued (the refresh request queue depth is four) when the MTSC is the target of a PCI cycle, the MTSC will assert STOP# as soon as the PCI specification allows.

#### 4.4.8. SUSPEND REFRESH

The MTSC supports two modes of refresh during suspend—CAS-before-RAS refresh and self-refresh. The refresh mode is selected in the DRAM Control Register.

For CAS-before-RAS refresh, CAS# signals are held low throughout suspend refresh while the RAS# signals toggle for refresh. Suspend mode refresh, in this case, is also triggered off the 32 KHz clock edges and cascading memory bank refresh takes place starting with bank 0.

An extended CAS-before-RAS cycle initiates the self-refreshing DRAMs. This special cycle tells the self-refreshing DRAMs to automatically increment their internal row address counters at their own pace. All memory control functions can be shutdown because the DRAM is refreshing itself. All RAS# and CAS# signals are held low.

The MTSC starts the suspend refresh when power management software writes to the SUSREF bit in the companion MPIIX component. During the suspend-to-DRAM mode, power can be removed from most of the MTSC but power must be supplied to the MTSC suspend refresh circuit in the "Resume Well" (via the VDDR pin). The PWRSD (Power Suspend-to-DRAM) signal should be driven by external logic to indicate a valid power supply to the "Resume Well". When the PWROK signal is negated by the main power supply, the MTSC isolates the DRAM refresh circuit in the "Resume Well".

#### 4.4.9. SYSTEM MANAGEMENT RAM

The Intel 430MX PCIsets support the use of main memory as System Management RAM (SMRAM), enabling the use of System Management Mode. When this function is disabled, the MTSC memory map is defined by the DRB and PAM Registers. When SMRAM is enabled, the MTSC reserves either the video buffer area (A and B segments) or C0000–CFFFFh of main memory for use as SMRAM.

SMRAM is placed at A0000–BFFFFh or C0000–CFFFFh via the SMRAM Space Register. Enhanced SMRAM features can also be enabled via this register. PCI masters can not access SMRAM when it is programmed to the video buffer area (A and B segments). When SMRAM is located at C0000–CFFFFh, this area of DRAM should be made readable, writeable, and non-cacheable via the corresponding PAM Registers.

### 4.5. 82438MX Data Path (MTDP)

The MTD provide the data path for host-to-main memory, PCI-to-main memory, and host-to-PCI cycles. Two MTD are required for the Intel 430MX PCIset system configuration. The MTSC controls the data flow through the MTD with the PCMD[1:0], HOE#, POE#, MOE#, MSTB#, and MADV# signals.

The MTDP have three data path interfaces; the host bus (HD[63:0]), the memory bus (MD[63:0]), and the PLINK[15:0] bus between the MTDP and MTSC. The data paths for the MTDP are interleaved on byte boundaries (Figure 10). Byte lanes 0, 2, 4, and 6 from the host CPU data bus connects to the even order MTDP and byte lanes 1, 3, 5, and 7 connect to the odd order MTDP. PLINK[7:0] connects to the even order MTDP and PLINK[15:8] connect to the odd order MTDP.

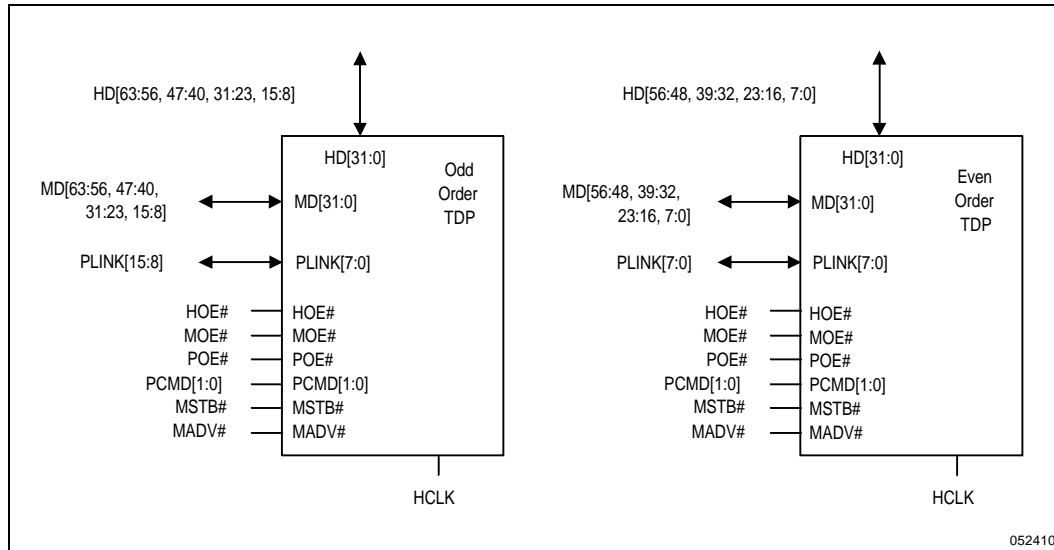


Figure 10. MTDP 64-Bit Data Path Partitioning

#### 4.6. PCI Bus Arbitration

The MTSC PCI Bus arbiter allows PCI peer-to-peer traffic concurrent with CPU main memory/second level cache cycles. The arbiter supports four PCI masters (Figure 11). REQ[2:0]#/GNT[2:0]# are used by PCI masters other than the PCI-to-EIO expansion bridge (MPIIX). PHLD#/PHLDA# are the arbitration request/grant signals for the MPIIX. PHLD#/PHLDA# also optimize system performance based on MPIIX known policies

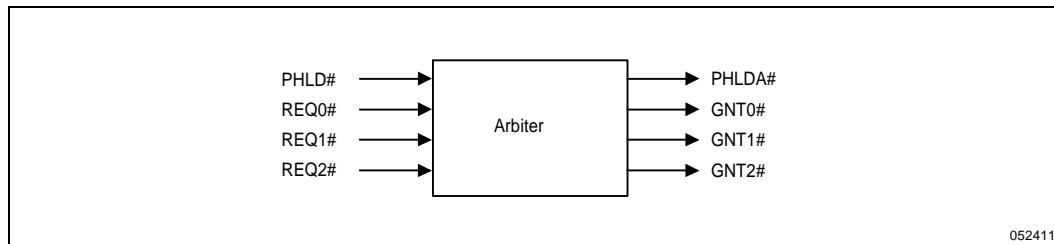


Figure 11. Arbiter

PCI Masters should follow the intent of the PCI Specification as quoted from the PCI Specification below:

1. "Agents must only use REQ# to signal a true need to use the bus."

2. "An agent must never use REQ# to "park" itself on the bus."

A "well behaved" agent should use REQ# when the bus is really needed. Typically REQ# should be removed, once the PCI master has been granted the bus, after FRAME# is asserted. Currently, PCI agents tested in Intel's compatibility labs appear to have the proper REQ# functionality.

Future PCI agents that *may not* be well behaved on REQ# lines could do the following:

1. Glitching REQ# lines for 1 or more clocks without any apparent reason.
2. Glitching REQ# lines and then not waiting for GNT# assertion
3. Continually asserting REQ# lines in an attempt to park itself on the PCI bus.
4. Failing to assert FRAME# within several clock of GNT# assertion when bus is idle.

PCI agents that are not well behaved may not function properly with the MTSC.

#### 4.6.1. PRIORITY SCHEME AND BUS GRANT

The arbitration mechanism employs two interacting priority queues; one for the CPU and one for the PCI agents. The CPU queue guarantees that the CPU is explicitly granted the bus on every fourth arbitration event. The PCI priority queue determines which PCI agent is granted when PCI wins the arbitration event.

A rotating priority scheme is used to determine the highest priority requester in the case of simultaneous requests. If the highest priority input at arbitration time does not have an active request, the next priority active requester is granted the bus. Granting the bus to a lower priority requester does not change the rotation order, but it does advance the priority rotation. The rotation priority chain is fixed. If the highest priority agent does not request the bus, the next agent in the chain is the highest priority, and so forth down the chain.

When no PCI agents are requesting the bus, the CPU is the default owner of the bus. CPU cycles incur no additional delays in this state.

The grant signals (GNTx#) are normally negated after FRAME# assertion or 24 PCLKs from grant assertion, if no cycle has started. The PCI specification allows a PCI master to have 16 PCI clocks after GNT# assertion to start the cycle before GNT# is taken away. PCI masters in a docking station will see the GNT# after a delay across the docking bridge. The MTSC arbiter therefore increases the GNT# removal latency to 24 PCI clocks. Once asserted, PHLDA# is only negated after PHLD# has been negated.

#### 4.6.2. CPU POLICIES

The CPU is either granted the bus as the high priority device or it is granted as a low priority device. Regardless of its priority, the CPU will never preempt a PCI Bus Master once that master has been granted the PCI bus. An AHOLD mechanism controls granting the bus to the CPU.

CPU locks to DRAM do not delay grants to a PCI device. However, if the device targets DRAM then the first snoop cycle and first datapath are delayed until the CPU lock is released.

##### CPU Grant as Low Priority

If the CPU was not the highest priority agent when it was granted, a new PCI request preempts the host bus grant, preventing new host bus cycles from starting. If the PCI agent's request does not target DRAM, the CPU is re-granted the host bus (AHOLD negated).

##### CPU Grant as High Priority

The MTSC priority mechanisms in conjunction with the Master Latency Timer (MLT) in each Bus Master allocate system resources between the CPU and PCI Bus Masters. These mechanisms also guarantee a programmable minimum percentage of the system to the CPU when it is the high priority device.



If the CPU was the highest priority agent when it was granted, then the CPU Inactivity Timer (CIT) and MLT are started. The CIT is retriggered when host bus activity occurs. The CPU is granted guaranteed ownership of the host and PCI busses until either the CIT timer expires (no host bus activity for awhile), the MLT timer expires (the CPU has hogged enough of the system), or a CPU to PCI access is retried. When any of these events occur, then the highest priority requesting PCI agent will be immediately granted the PCI bus. If no PCI requests exist at the time of CIT/MLT/retry events, the CPU is dropped to the low priority state, allowing any future PCI requests to preempt the CPU.

Note that a CIT value of at least 3 PCLKs is required to guarantee host activity is recognized by the arbiter, and prevent CPU preemption by PCI requests.

#### PHLD#/PHLDA# Handshake Rules

The special rules governing the PHLD#/PHLDA# handshake are described below.

#### Normal Operation

1. The minimum arbitration delay, measured from PHLD# assertion to PHLDA# sampling is 4 PCLKs (1a). (Note that this minimum timing is a MTSC implementation choice, and not a general architectural requirement for the PHLD#/PHLDA# handshake.) Once PHLD# has been asserted, PHLD# should not be negated until after PHLDA# has been sampled asserted, unless the cycle transfer is aborted. Once PHLDA# has been asserted, it will not be negated until PHLD# has been negated. PHLDA# is negated 1 PCLK from sampling PHLD# negated (1b). A master not implementing the multimedia extended protocol must release PHLD# for a minimum of 2 PCLKs (1c).

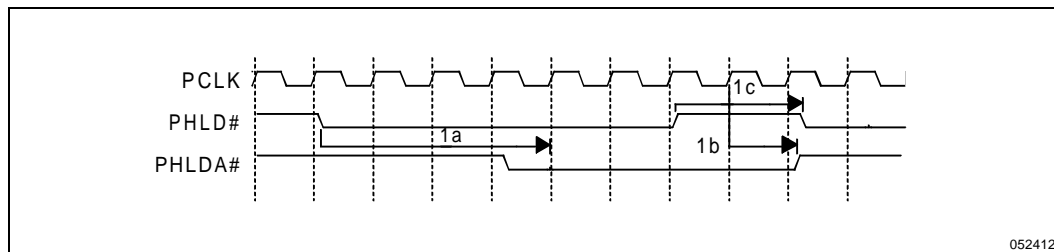


Figure 12. PHLD/PHLDA# Arbitration Timing

2. A master using PHLD# can not start a cycle without both PHLD# and PHLDA# asserted, regardless of the state of PHLDA#. This results in the latest valid assertion point for FRAME# relative to PHLD# negation

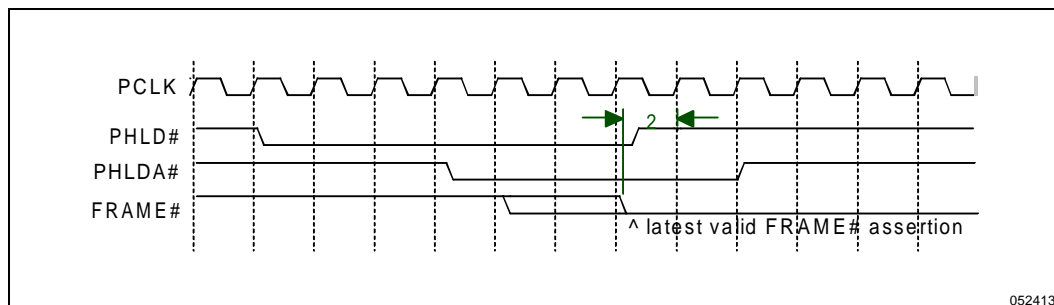


Figure 13. Master Starting a Cycle

3. A master using PHLD# can not generate fast back to back cycles to any target. The minimum delay from the last data phase to the next FRAME# assertion is 1 PCLK.

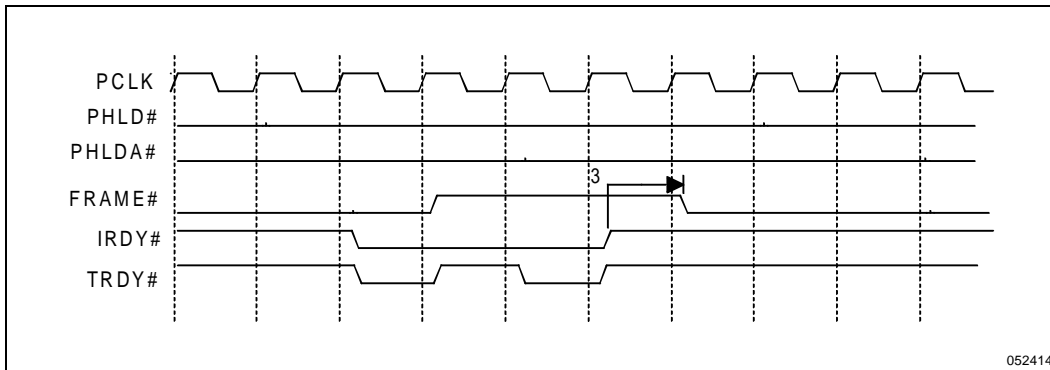


Figure 14. Delay From Last Data Phase To Next FRAME#

4. A master using PHLD# can not assert PLOCK#.

#### Multimedia Extended Operation

The basic GAT capability is provided by the PHLD#/PHLDA# handshake rules. In addition, the MTSC arbiter implements hooks to enable a preferred operation for multimedia systems based on enhanced ISA bridges (i.e., beyond the MPIIX capabilities).

During an ISA to PCI transfer, the ISA bridge may release the PCI bus during the ISA transfer by pulsing PHLD# negated for 1 PCI clock. This signals a *passive release* of the PCI bus to the arbiter. When the arbiter detects this passive release, it is free to grant the bus to other PCI masters or the CPU. Further grants to the ISA bridge follow the same rules as the initial ones w.r.t. grant priority and buffer flushing.

After the final ISA transfer has completed, the ISA bridge should release the PCI bus by negating PHLD# for a minimum of 2 PCI clocks. This signals an active release to the arbiter.

From the initial ISA grant to the active release by the ISA bridge, the minimum architectural requirement is for the arbiter to prevent any CPU to PCI write posting. However, for the MTSC implementation, the arbiter blocks all CPU to PCI cycles. CPU cycles that are in progress and prior to FRAME# assertion on the first PCI grant are BOFFed, and CPU cycles starting prior to the active release are stalled (no BRDY#).

The additional handshake rules for the extended operation follow.

5. **Passive bus release** (i.e., pulsing PHLD# negated for a single PCLK) may only be signaled by the master concurrent with the IRDY# assertion for the last data phase of a transaction (5). PHLDA# will be negated when PHLD# is sampled negated at this point (5a). PHLDA# or another GNT# will be asserted at a minimum of 2 PCLKs after PHLDA# negation (5b) TRDY# assertion in the last data phase (5b), or 1 PCLK after TRDY# assertion in the last dataphase (5c). These requirements are shown in (Note that 5b and 5c timings are a MTSC implementation choice, and not a general architectural requirement for the PHLD#/PHLDA# handshake.) Note that an additional operation requirement unrelated to PHLD# is that only single data transfers be performed on passive releases. Therefore, the first dataphase is guaranteed to also be the last one.

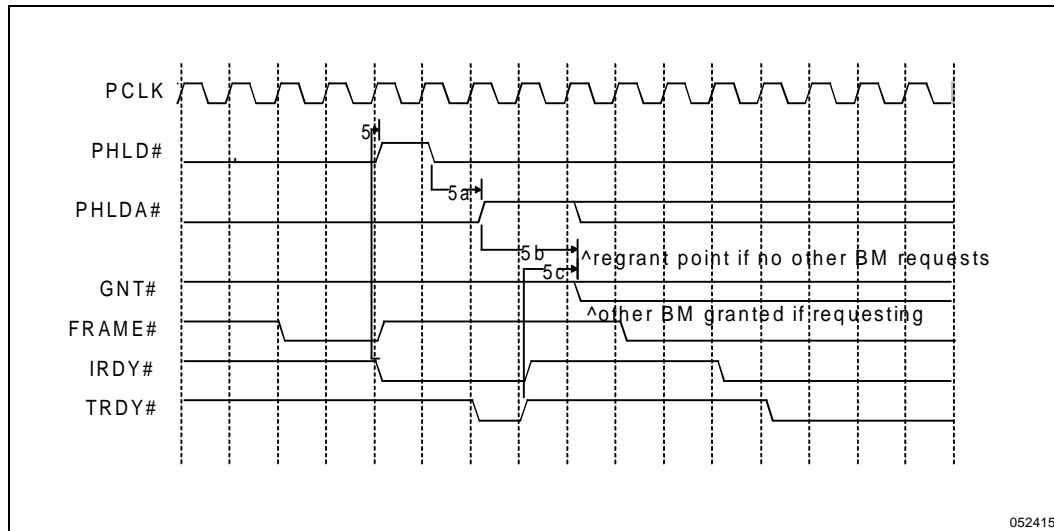


Figure 15. PHLD# and Passive Release

6. **Active bus release** requires a minimum PHLD# negation time of 2 PCLKs (Figure 15 PHLD# Assertion/Negation Rules).

## 4.7. Clocks and Reset

### 4.7.1. CLOCKS

The MTSC and CPU should be clocked from one clock driver output to minimize skew between the CPU and MTSC. The MTDs should share another clock driver output (Figure 16).

### 4.7.2. RESET SEQUENCING

The MTSC is asynchronously reset by the PCI reset (RST#). The MTSC resets the MTD by driving HOE#, MOE#, and POE# to 1 during reset. The MTSC changes HOE#, MOE#, and POE# to their default value after the MTD is reset.

#### Arbiter (Central Resource) Functions on Reset

The MTSC arbiter includes support for PCI central resource functions. These functions include driving the AD[31:0], C/BE[3:0]#, and PAR signals when no one is granted the PCI bus and the bus is idle. The MTSC drives 0's on these signals during reset and drives valid levels when no other agent is granted and the bus is idle.

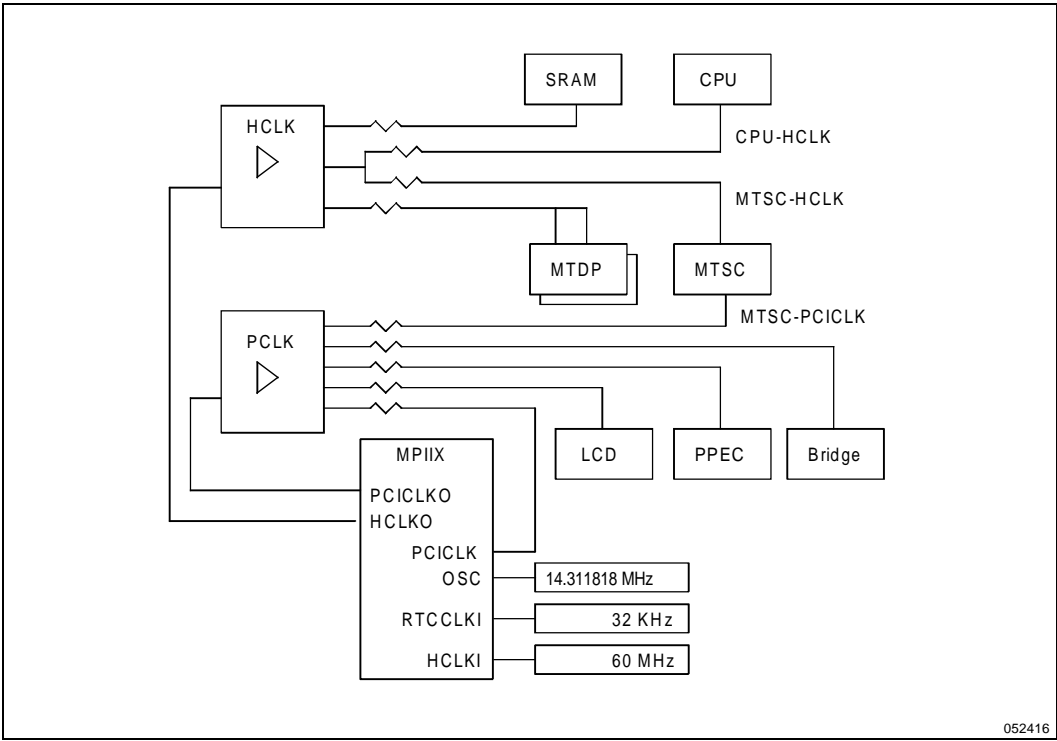


Figure 16. Clock Distribution

#### 4.8. PCI Clock Control (CLKRUN#)

There are three main states in the clocking protocol:

- **Clock Running:** The clock is running and the bus is operational.
- **About to Stop:** The central resource has indicated on the CLKRUN# line that the clock is about to stop.
- **Clock Stopped:** The clock is stopped with CLKRUN# being monitored for a restart.

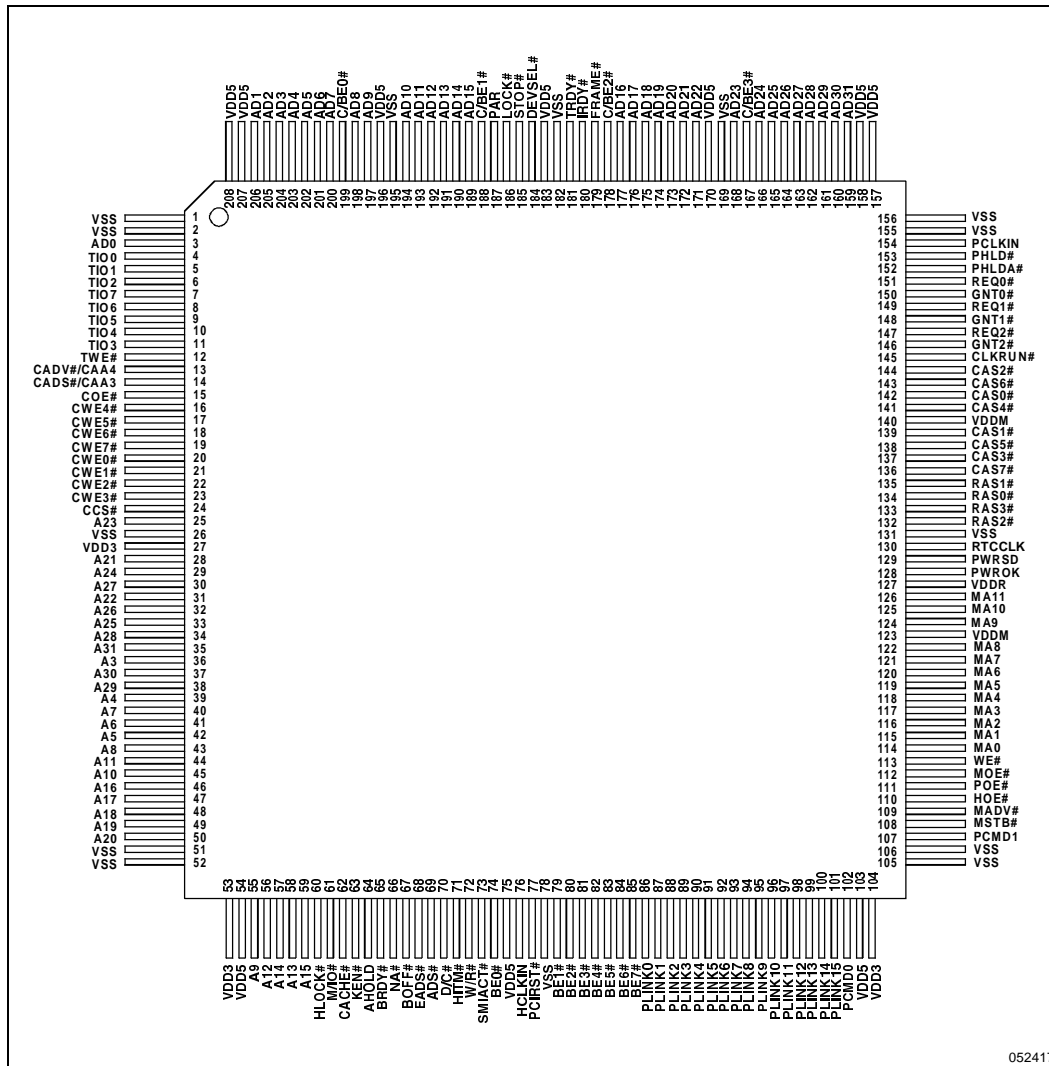
A PCI Master drives CLKRUN# low to restart the clock so that it can assert its bus request synchronously. Multiple masters requesting clock restart is allowed, but only one will receive a grant. After master releases drive of CLKRUN# the central resource takes over clock drive. A master may not assert CLKRUN# unless it is sampled HIGH with CLK (before the clock is stopped). The MTSC is a CLKRUN# master device. The MPIIX companion component controls the clocks in the system and is the CLKRUN# central resource.

If a target of an access samples CLKRUN# high, it drives CLKRUN# low to maintain the clock so that it can assure internal PCI clock related functions complete. After the target samples CLKRUN# LOW for two cycles, it releases drive of CLKRUN#. The central resource takes over low drive. In this way, the target only drives CLKRUN# for two CLK rising edges. The target may not assert CLKRUN# unless it is sampled high with CLK (before the clock is stopped).

For additional information on CLKRUN#, see the PCI Mobile Design Guide.

## 5.0. PINOUT AND PACKAGE INFORMATION

### 5.1. MTSC Pin Assignment



**Figure 17. MTSC Pinout Diagram**

Table 15. MTSC Alphabetical Pin Assignment

Name	Pin	Type	Name	Pin	Type	Name	Pin	Type
A10	45	I/O	AD0	3	I/O	AD7	200	I/O
A11	44	I/O	AD1	206	I/O	AD8	198	I/O
A12	56	I/O	AD10	194	I/O	AD9	197	I/O
A13	58	I/O	AD11	193	I/O	ADS#	69	I
A14	57	I/O	AD12	192	I/O	AHOLD	64	O
A15	59	I/O	AD13	191	I/O	BE0#	74	I
A16	46	I/O	AD14	190	I/O	BE1#	79	I
A17	47	I/O	AD15	189	I/O	BE2#	80	I
A18	48	I/O	AD16	177	I/O	BE3#	81	I
A19	49	I/O	AD17	176	I/O	BE4#	82	I
A20	50	I/O	AD18	175	I/O	BE5#	83	I
A21	28	I/O	AD19	174	I/O	BE6#	84	I
A22	31	I/O	AD2	205	I/O	BE7#	85	I
A23	25	I/O	AD20	173	I/O	BOFF#	67	O
A24	29	I/O	AD21	172	I/O	BRDY#	65	O
A25	33	I/O	AD22	171	I/O	C/BE0#	199	I/O
A26	32	I/O	AD23	168	I/O	C/BE1#	188	I/O
A27	30	I/O	AD24	166	I/O	C/BE2#	178	I/O
A28	34	I/O	AD25	165	I/O	C/BE3#	167	I/O
A29	38	I/O	AD26	164	I/O	CACHE#	62	I
A3	36	I/O	AD27	163	I/O	CADS#/ CA3	14	O
A30	37	I/O	AD28	162	I/O	CADV#/ CA4	13	O
A31	35	I/O	AD29	161	I/O	CAS0#	142	O
A4	39	I/O	AD3	204	I/O	CAS1#	139	O
A5	42	I/O	AD30	160	I/O	CAS2#	144	O
A6	41	I/O	AD31	159	I/O	CAS3#	137	O
A7	40	I/O	AD4	203	I/O	CAS4#	141	O
A8	43	I/O	AD5	202	I/O	CAS5#	138	O
A9	55	I/O	AD6	201	I/O			

Name	Pin	Type
CAS6#	143	O
CAS7#	136	O
CCS#	24	O
CLKRUN#	145	I/OD
COE#	15	O
CWE0#	20	O
CWE1#	21	O
CWE2#	22	O
CWE3#	23	O
CWE4#	16	O
CWE5#	17	O
CWE6#	18	O
CWE7#	19	O
D/C#	70	I
DEVSEL#	184	I/O
EADS#	68	O
FRAME#	179	I/O
GNT0#	150	O
GNT1#	148	O
GNT2#	146	O
HCLKIN	76	I
HITM#	71	I
HLOCK#	60	I
HOE#	110	O
IRDY#	180	I/O
KEN#	63	O
LOCK#	186	I/O
M/IO#	61	I
MA0	114	O
MA1	115	O
MA10	125	O

Name	Pin	Type
MA11	126	O
MA2	116	O
MA3	117	O
MA4	118	O
MA5	119	O
MA6	120	O
MA7	121	O
MA8	122	O
MA9	124	O
MADV#	109	O
MOE#	112	O
MSTB#	108	O
NA#	66	O
PAR	187	I/O
PCIRST#	77	I
PCLKIN	154	I
PCMD0	102	O
PCMD1	107	O
PHLD#	153	I
PHLDA#	152	O
PLINK0	86	I/O
PLINK1	87	I/O
PLINK10	96	I/O
PLINK11	97	I/O
PLINK12	98	I/O
PLINK13	99	I/O
PLINK14	100	I/O
PLINK15	101	I/O
PLINK2	88	I/O
PLINK3	89	I/O
PLINK4	90	I/O

Name	Pin	Type
PLINK5	91	I/O
PLINK6	92	I/O
PLINK7	93	I/O
PLINK8	94	I/O
PLINK9	95	I/O
POE#	111	O
PWROK	128	I
PWRSD	129	V
RAS0#	134	O
RAS1#	135	O
RAS2#	132	O
RAS3#	133	O
REQ0#	151	I
REQ1#	149	I
REQ2#	147	I
RTCCLK	130	I
SMIACK#	73	I
STOP#	185	I/O
TIO0	4	I/O
TIO1	5	I/O
TIO2	6	I/O
TIO3	11	I/O
TIO4	10	I/O
TIO5	9	I/O
TIO6	8	I/O
TIO7	7	I/O
TRDY#	181	I/O
TWE#	12	O
VDD3	27	V
VDD3	53	V
VDD3	104	V



Name	Pin	Type
VDD5	54	V
VDD5	75	V
VDD5	103	V
VDD5	157	V
VDD5	158	V
VDD5	170	V
VDD5	183	V
VDD5	196	V
VDD5	207	V
VDD5	208	V

Name	Pin	Type
VDDM	123	V
VDDM	140	V
VDDR	127	V
VSS	1	V
VSS	2	V
VSS	26	V
VSS	51	V
VSS	52	V
VSS	78	V
VSS	105	V

Name	Pin	Type
VSS	106	V
VSS	131	V
VSS	155	V
VSS	156	V
VSS	169	V
VSS	182	V
VSS	195	V
W/R#	72	I
WE#	113	O

5.2. MTD Pin Assignment

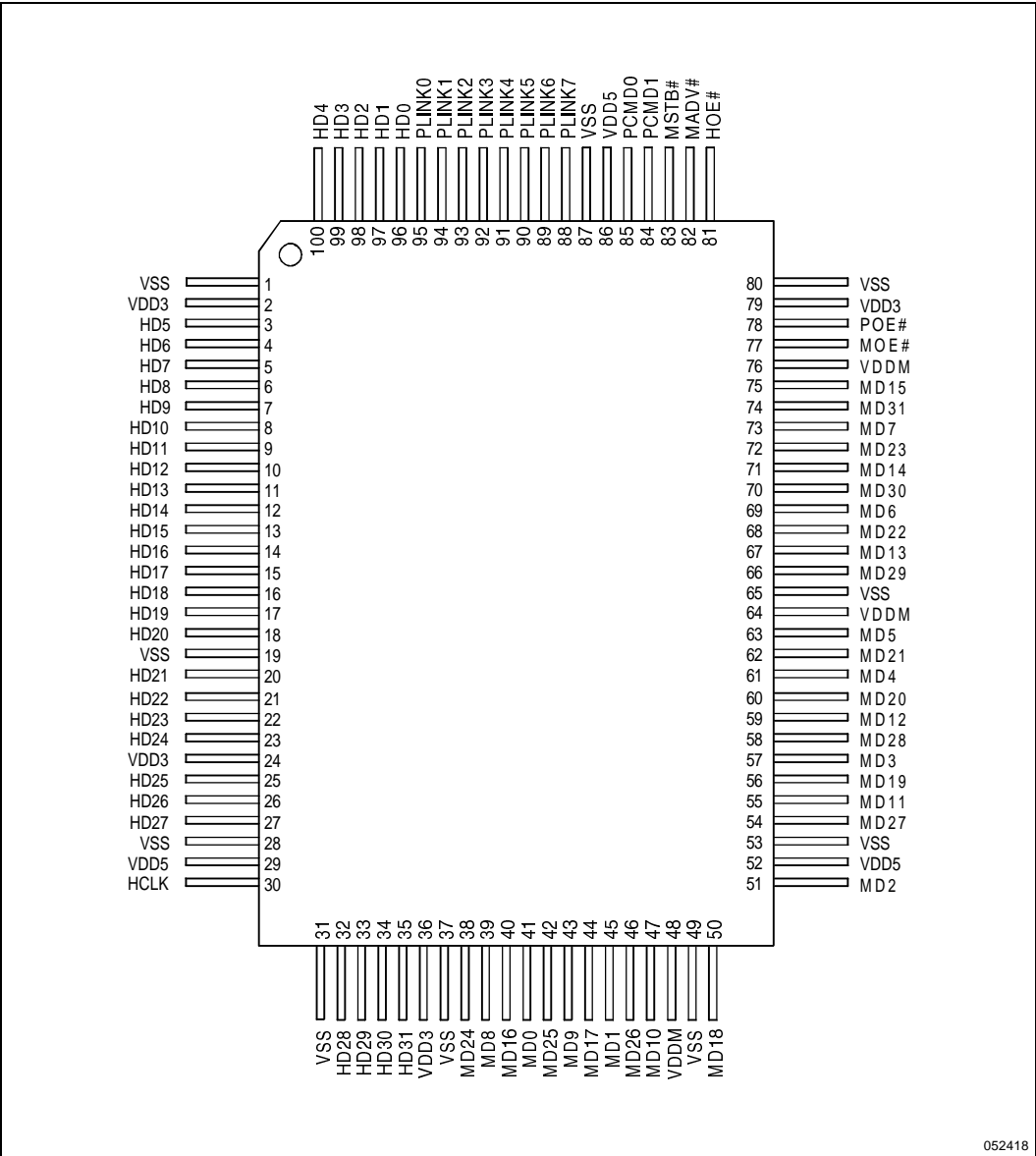


Figure 18. MTD Pinout Diagram

Table 16. MTD Alphabetical Pin Assignment

Name	Pin #	Type	Name	Pin #	Type	Name	Pin #	Type
HCLK	30	I	HD6	4	I/O	MD3	57	I/O
HD0	96	I/O	HD7	5	I/O	MD30	70	I/O
HD1	97	I/O	HD8	6	I/O	MD31	74	I/O
HD10	8	I/O	HD9	7	I/O	MD4	61	I/O
HD11	9	I/O	HOE#	81	I	MD5	63	I/O
HD12	10	I/O	MADV#	82	I	MD6	69	I/O
HD13	11	I/O	MD0	41	I/O	MD7	73	I/O
HD14	12	I/O	MD1	45	I/O	MD8	39	I/O
HD15	13	I/O	MD10	47	I/O	MD9	43	I/O
HD16	14	I/O	MD11	55	I/O	MOE#	77	I
HD17	15	I/O	MD12	59	I/O	MSTB#	83	I
HD18	16	I/O	MD13	67	I/O	PCMD0	85	I
HD19	17	I/O	MD14	71	I/O	PCMD1	84	I
HD2	98	I/O	MD15	75	I/O	PLINK0	95	I/O
HD20	18	I/O	MD16	40	I/O	PLINK1	94	I/O
HD21	20	I/O	MD17	44	I/O	PLINK2	93	I/O
HD22	21	I/O	MD18	50	I/O	PLINK3	92	I/O
HD23	22	I/O	MD19	56	I/O	PLINK4	91	I/O
HD24	23	I/O	MD2	51	I/O	PLINK5	90	I/O
HD25	25	I/O	MD20	60	I/O	PLINK6	89	I/O
HD26	26	I/O	MD21	62	I/O	PLINK7	88	I/O
HD27	27	I/O	MD22	68	I/O	POE#	78	I
HD28	32	I/O	MD23	72	I/O	VDD3	2	V
HD29	33	I/O	MD24	38	I/O	VDD3	24	V
HD3	99	I/O	MD25	42	I/O	VDD3	36	V
HD30	34	I/O	MD26	46	I/O	VDD3	79	V
HD31	35	I/O	MD27	54	I/O	VDD5	29	V
HD4	100	I/O	MD28	58	I/O	VDD5	52	V
HD5	3	I/O	MD29	66	I/O	VDD5	86	V



Name	Pin #	Type
VDDM	48	V
VDDM	64	V
VDDM	76	V
VSS	1	V
VSS	19	V

Name	Pin #	Type
VSS	28	V
VSS	31	V
VSS	37	V
VSS	49	V
VSS	53	V

Name	Pin #	Type
VSS	65	V
VSS	80	V
VSS	87	V

### 5.3. MTSC Package Characteristics

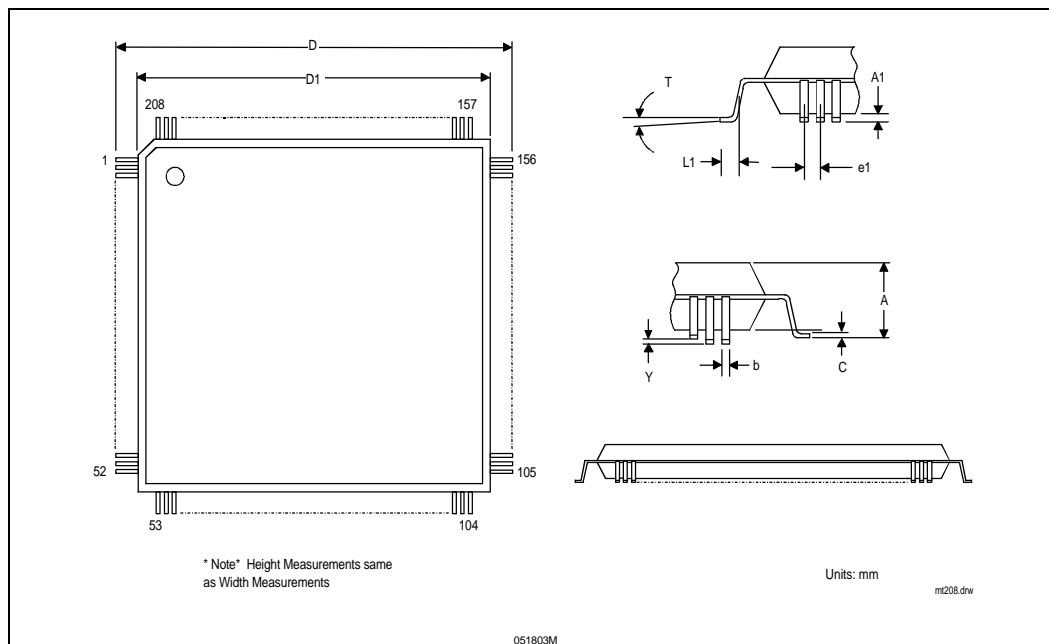


Figure 19. MTSC Physical Dimension Diagram (208-Lead TQFP)

Table 17. MTSC Physical Dimension (208-Lead TQFP)

Symbol	Dimension in Millimeters		
	Minimum	Nominal	Maximum
A		3.5	3.75
A1	0.05	0.15	0.25
b	0.13	0.18	0.28
C	0.10	0.15	0.20
D	30.3	30.6	30.9
D1	27.9	28.0	28.1
e1	0.4	0.5	0.6
L1	0.4	0.5	0.6
Y			0.08
T	0		10

5.4. MTDP Package Characteristics

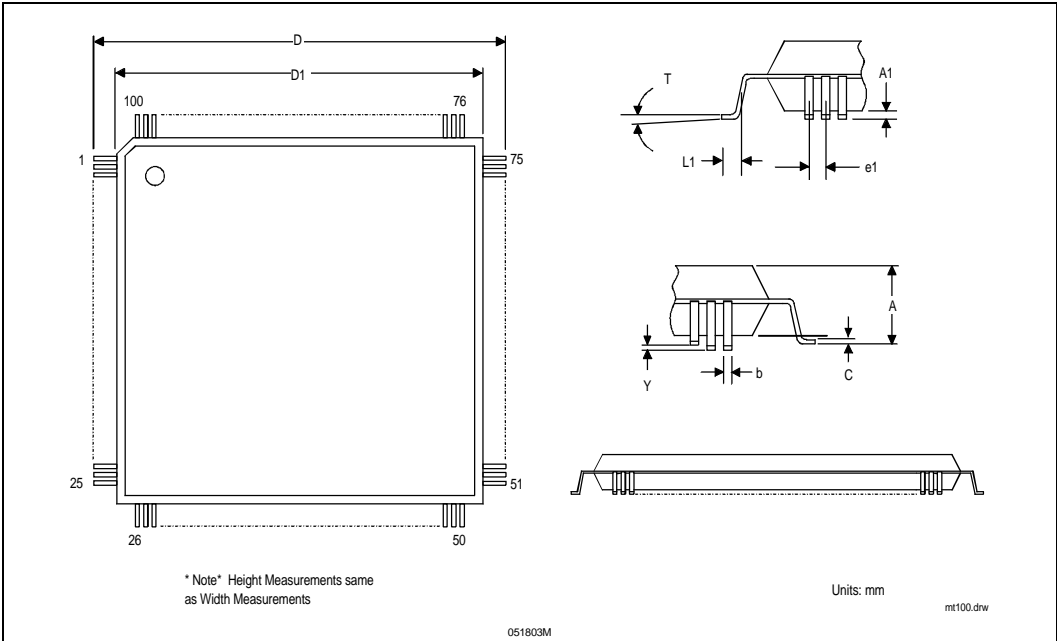


Figure 20. MTDP Physical Dimension Diagram (100-Lead TQFP)

Table 18. MTDP Physical Dimension Diagram (100-Lead TQFP)

Symbol	Dimension in Millimeters		
	Minimum	Nominal	Maximum
A			1.7
A1	0	0.1	0.2
b	0.13	0.18	0.28
C	0.105	0.125	0.170
D	15.8	16.0	16.2
D1	13.9	14.0	14.1
e1		.05	
L1		1.0	
Y			0.1
T	0		10

## 6.0. TESTABILITY

### 6.1. 82437MX MTSC TESTABILITY

In NAND Tree Mode the device pins are configured in a chain test structure terminating at an observation point. NAND Tree is decoded from the REQ[2:0]#, PWRSD, and PCIRST# pins. Mode selection is asynchronous, and the signals need to remain in their respective states for the duration of the test.

Test Mode	PCIRST#	REQ0#	REQ1#	REQ2#	PWRSD
NAND Tree	0	0	0	0	0

The NAND Tree test mode tri-states all outputs and bi-directional buffers except for PCIRST#, REQ[2:0]#, and GNT[2:1]#. The NAND Tree follows the pins sequentially around the chip skipping only PCIRST#, REQ[2:0]#, and GNT[2:1]#. The first input of the NAND chain is GNT0#, and the NAND chain is routed counter-clockwise around the chip (e.g., GNT0#, PHLDA#, . . .). The only valid outputs during NAND Tree mode are GNT1# and GNT2#. GNT1# is the final stage output and GNT2# is an interim stage (A8 buffered) of the NAND Tree.

#### 6.1.1. TEST MODE OPERATION

The NAND Tree mode is entered by driving the PCIRST#, REQ[2:0]#, and PWRSD pins low while all other MTSC inputs are driven high. This causes all MTSC outputs and bi-directional pins to be floated.

To start the NAND Tree test continue to hold the PCIRST# and REQ[2:0]# pins low while the PWRSD pin and all other inputs, outputs and bi-directional pins are driven high.

To execute the NAND Tree test, beginning with GNT0# and working counter-clockwise around the chip through A8, toggle each pin in the NAND chain to a 0 singularly and observe the resulting toggles on both GNT1# and GNT2#. Each pin is toggled only once and must remain at 0 thereafter. Continue toggling the chain from A11 through MA11 and observe resulting toggles on GNT1# only. For this section of the chain GNT2# remains a constant 1. The final section of the chain, from MA11 through CLKRUN# requires special treatment. Rather than toggling PWROK (the next cell of the chain), toggle MA11 back to a 1 while leaving PWROK high. Observe a toggle on GNT1#. Lastly, the remainder of the chain from PWRSD through CLKRUN# is toggled in sequence to a 0. The GNT1# pin will toggle for each transition.

#### 6.1.2. TEST ISSUES

To avoid pin contentions between the MTSC and tester drivers for the RAS[3:0]#, CAS[7:0]#, MA[11:0], and WE# pins, the PWROK pin must never be driven low during a NAND Tree test. PWROK acts to enable the output drivers for these pins. Note that since the PWROK pin is skipped, GNT1# is toggled by the reassertion of MA11. The reassertion of MA11 before PWRSD is negated guarantees that PWRSD is tested.

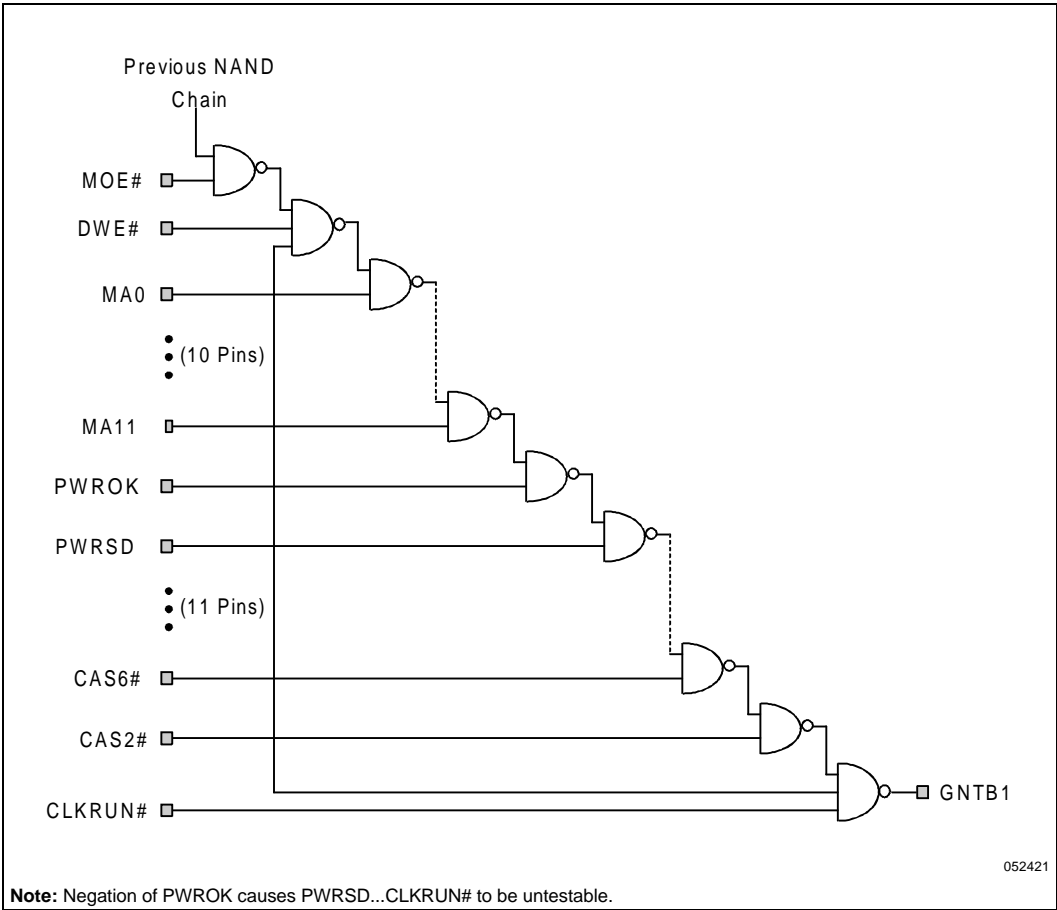


Figure 21. NAND Tree Structure (With PWROK 3V Well Isolation)



### 6.1.3. NAND TREE TIMING REQUIREMENTS

Allow 800 ns for the input signals to propagate to the NAND tree outputs (input-to-output propagation delay specification).

**Table 19. MTSC NAND Tree**

Order	Pin #	Pin Name	Notes	Order	Pin #	Pin Name	Notes
	77	PCIRST#	Must be 0 to enter NAND tree mode. This signal must remain 0 during the entire NAND tree test.	19	175	AD18	
	129	PWRSD	Must be 0 to enter NAND tree mode. This signal is driven high to initiate the NAND tree test.	20	176	AD17	
	147	REQ2#	Must be 0 to enter NAND tree mode. This signal must remain 0 during the entire NAND tree test.	21	177	AD16	
	149	REQ1#	Must be 0 to enter NAND tree mode. This signal must remain 0 during the entire NAND tree test.	22	178	C/BE2#	
	151	REQ0#	Must be 0 to enter NAND tree mode. This signal must remain 0 during the entire NAND tree test.	23	179	FRAME#	
1	150	GNT0#	Start of the NAND Tree chain.	24	180	IRDY#	
2	152	PHLDA#		25	181	TRDY#	
3	153	PHLD#		26	184	DEVSEL#	
4	154	PCLKIN		27	185	STOP#	
5	159	AD31		28	186	LOCK#	
6	160	AD30		29	187	PAR	
7	161	AD29		30	188	C/BE1#	
8	162	AD28		31	189	AD15	
9	163	AD27		32	190	AD14	
10	164	AD26		33	191	AD13	
11	165	AD25		34	192	AD12	
12	166	AD24		35	193	AD11	
13	167	C/BE3#		36	194	AD10	
14	168	AD23		37	197	AD9	
15	171	AD22		38	198	AD8	
16	172	AD21		39	199	C/BE0#	
17	173	AD20		40	200	AD7	
18	174	AD19		41	201	AD6	
				42	202	AD5	
				43	203	AD4	
				44	204	AD3	
				45	205	AD2	
				46	206	AD1	
				47	3	AD0	
				48	4	TIO0	
				49	5	TIO1	
				50	6	TIO2	
				51	7	TIO7	
				52	8	TIO6	

Order	Pin #	Pin Name	Notes
53	9	TIO5	
54	10	TIO4	
55	11	TIO3	
56	12	TWE#	
57	13	CADV#/ CA4	
58	14	CADS#/ CA3	
59	15	COE#	
60	16	CWE4#	
61	17	CWE5#	
62	18	CWE6#	
63	19	CWE7#	
64	20	CWE0#	
65	21	CWE1#	
66	22	CWE2#	
67	23	CWE3#	
68	24	CCS#	
69	25	A23	
70	28	A21	
71	29	A24	
72	30	A27	
73	31	A22	
74	32	A26	
75	33	A25	
76	34	A28	
77	35	A31	
78	36	A3	
79	37	A30	
80	38	A29	
81	39	A4	
82	40	A7	
83	41	A6	
84	42	A5	
85	43	A8	
86	44	A11	
87	45	A10	
88	46	A16	
89	47	A17	
90	48	A18	

Order	Pin #	Pin Name	Notes
91	49	A19	
92	50	A20	
93	55	A9	
94	56	A12	
95	57	A14	
96	58	A13	
97	59	A15	
98	60	HLOCK#	
99	61	M/IO#	
100	62	CACHE#	
101	63	KEN#	
102	64	AHOLD	
103	65	BRDY#	
104	66	NA#	
105	67	BOFF#	
106	68	EADS#	
107	69	ADS#	
108	70	D/C#	
109	71	HITM#	
110	72	W/R#	
111	73	SMIACK#	
112	76	HCLKIN	
113	74	BE0#	
114	79	BE1#	
115	80	BE2#	
116	81	BE3#	
117	82	BE4#	
118	83	BE5#	
119	84	BE6#	
120	85	BE7#	
121	86	PLINK0	
122	87	PLINK1	
123	88	PLINK2	
124	89	PLINK3	
125	90	PLINK4	
126	91	PLINK5	
127	92	PLINK6	
128	93	PLINK7	

Order	Pin #	Pin Name	Notes
129	94	PLINK8	
130	95	PLINK9	
131	96	PLINK10	
132	97	PLINK11	
133	98	PLINK12	
134	99	PLINK13	
135	100	PLINK14	
136	101	PLINK15	
137	102	PCMD0	
138	107	PCMD1	
139	108	MSTB#	
140	109	MADV#	
141	110	HOE#	
142	111	POE#	
143	112	MOE#	
144	113	WE#	
145	114	MA0	
146	115	MA1	
147	116	MA2	
148	117	MA3	
149	118	MA4	
150	119	MA5	
151	120	MA6	
152	121	MA7	

Order	Pin #	Pin Name	Notes
153	122	MA8	
154	124	MA9	
155	125	MA10	
156	126	MA11	
157	128	PWROK	Skip
158	129	PWRSD	
159	130	RTCCLK	
160	132	RAS2#	
161	133	RAS3#	
162	134	RAS0#	
163	135	RAS1#	
164	136	CAS7#	
165	137	CAS3#	
166	138	CAS5#	
167	139	CAS1#	
168	141	CAS4#	
169	142	CAS0#	
170	143	CAS6#	
171	144	CAS2#	
172	145	CLKRUN#	
	146	GNT2#	Interim output of the NAND Tree chain.
	148	GNT1#	Final output of the NAND Tree chain.

6.2. 82438MX MTD TESTABILITY

The device pins are configured in a chain test structure terminating in a single observation point. The NAND Tree test mode is decoded from the HOE#, MOE#, POE#, and MSTB# pins and is defined as follows.

Test Mode	HOE#	MOE#	POE#	MSTB#
NAND Tree	1	1	1	1

The NAND Tree test mode tri-states all outputs and bi-directional buffers except for MD0 which is the output of the NAND Tree. The NAND Tree follows the pins sequentially around the chip, skipping only HCLK and MD0. The first chain input is HD5, and the route is counter-clockwise around the chip (e.g., HD5, HD6...).

6.2.1. NAND TREE TEST MODE OPERATION

The NAND Tree mode is entered by driving all pins included in the NAND Tree high with the exception of HCLK and MD0. HCLK must be active for at least one clock to sample HOE#, MOE#, POE# and MSTB#. Once these signals are sampled HCLK must be driven low for the duration of the NAND Tree test.

To execute the NAND Tree test, beginning at HD5, sequentially toggle each pin in the NAND chain low and observe a resulting toggle on MD0. Each pin is toggled only once and must remain low for the remainder of the NAND Tree test.

The only valid output during NAND Tree mode is MD0.

The NAND Tree mode is exited by starting HCLK with the HOE#, MOE#, and POE# pins **not equal** to '111'.

Schematic of MTD NAND Tree circuitry.

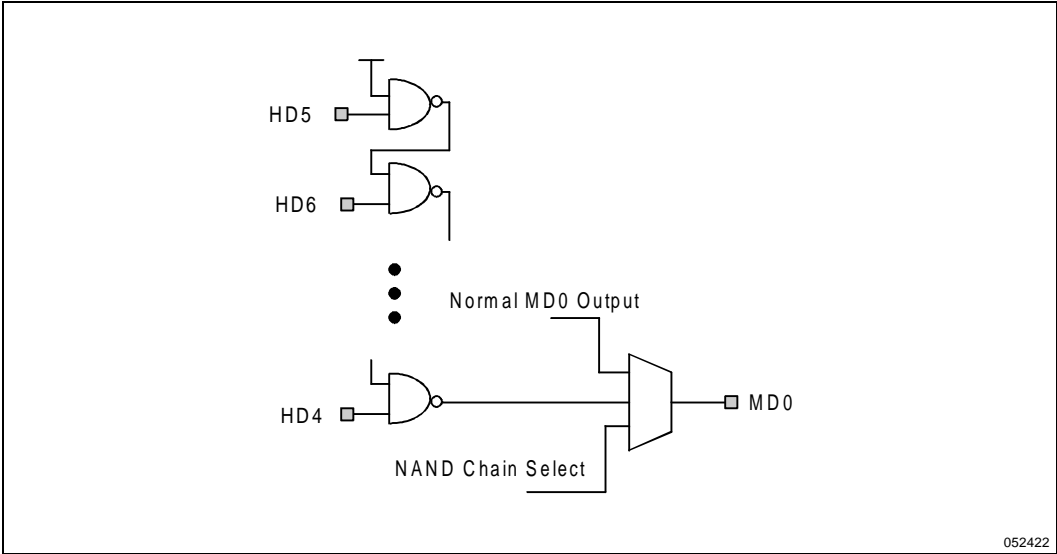


Figure 22. 82378MX NAND Tree Chain

## NAND Tree Timing Requirements

Allow 500 ns for each input signal transition to propagate to the MD0 output (input-to-output propagation delay specification).

**Table 20. MTD NAND Tree**

Pin #	Pin Name	Notes	Pin #	Pin Name	Notes
77	MOE#	Must be 1 to enter NAND Tree mode. This pin is included in the NAND Tree mode input pin sequence.	14	HD16	
78	POE#	Must be 1 to enter NAND Tree mode. This pin is included in the NAND Tree mode input pin sequence.	15	HD17	
81	HOE#	Must be 1 to enter NAND Tree mode. This pin is included in the NAND Tree mode input pin sequence.	16	HD18	
83	MSTB#	Must be 1 to enter NAND Tree mode. This pin is included in the NAND Tree mode input pin sequence.	17	HD19	
30	HCLK	HCLK must clock at least once to sample MOE#, POE#, HOE#, and MSTB# to select NAND Tree mode. Then to enter NAND Tree mode HCLK must remain 0. To exit NAND Tree mode HCLK must be started.	18	HD20	
3	HD5	First signal in the NAND Tree chain.	20	HD21	
4	HD6		21	HD22	
5	HD7		22	HD23	
6	HD8		23	HD24	
7	HD9		25	HD25	
8	HD10		26	HD26	
9	HD11		27	HD27	
10	HD12		32	HD28	
11	HD13		33	HD29	
12	HD14		34	HD30	
13	HD15		35	HD31	
			38	MD24	
			39	MD8	
			40	MD16	
			42	MD25	
			43	MD9	
			44	MD17	
			45	MD1	
			46	MD26	
			47	MD10	
			50	MD18	
			51	MD2	

Pin #	Pin Name	Notes
54	MD27	
55	MD11	
56	MD19	
57	MD3	
58	MD28	
59	MD12	
60	MD20	
61	MD4	
62	MD21	
63	MD5	
66	MD29	
67	MD13	
68	MD22	
69	MD6	
70	MD30	
71	MD14	
72	MD23	
73	MD7	
74	MD31	
75	MD15	
77	MOE#	Must be 1 to enter NAND Tree mode. This pin is included in the NAND Tree mode input pin sequence.
78	POE#	Must be 1 to enter NAND Tree mode. This pin is included in the NAND Tree mode input pin sequence.

Pin #	Pin Name	Notes
81	HOE#	Must be 1 to enter NAND Tree mode. This pin is included in the NAND Tree mode input pin sequence.
82	MADV#	
83	MSTB#	Must be 1 to enter NAND Tree mode. This pin is included in the NAND Tree mode input pin sequence.
84	PCMD1	
85	PCMD0	
88	PLINK7	
89	PLINK6	
90	PLINK5	
91	PLINK4	
92	PLINK3	
93	PLINK2	
94	PLINK1	
95	PLINK0	
96	HD0	
97	HD1	
98	HD2	
99	HD3	
100	HD4	Final signal in the NAND Tree chain.
41	MD0	Output of the NAND Tree chain.