PRELIMINARY

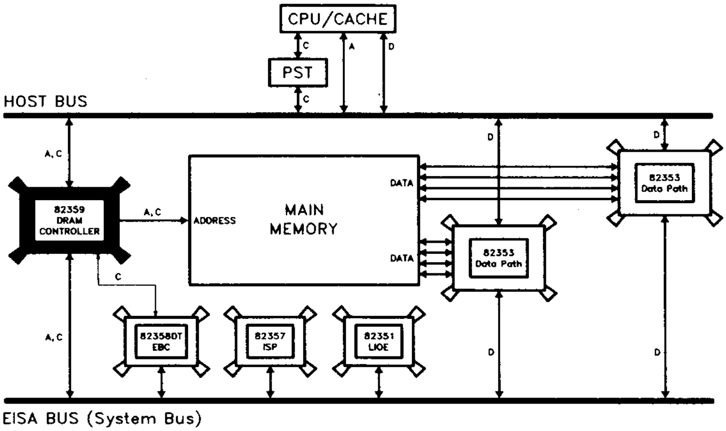
82359
DRAM CONTROLLER

- **Dual Ported Memory Controller**
  - **—Allows EISA/Host Bus Concurrency**
  - **—CPU Speed Independent**
  - **—Controls up to 256M of Motherboard DRAM**
  - **—LIM Hardware Support**
  - **—Support for Shadow/Disable/Remap/Cacheing/Write Protect of Motherboard Memory**
- **Flexible DRAM Support**
  - **—64K, 256K, 1M, 4M, 16M (4M x 4) DRAMs**
  - **—60 ns, 70 ns, 80 ns speeds**
  - **—Single or Double Density SIMMs**
  - **—Ability to Mix DRAM Sizes**
  - **—Supports 32-, 64-, or 128-Bit Wide Memory Configurations**
- **High Integration**
  - **—Integrated Posted Write Latch**
  - **—Cacheability/Write Protect Map**
  - **—Integrated Delay Lines**
  - **—Integrated Bus Drivers**
- **Integrated Delay Line**
  - **—Critical DRAM Timings Generated Internally**
  - **—DRAM Timings are Programmable with 2.61 ns Resolution**
- **Cache Support**
  - **—Support for 82385, 82395, 82485 Cache**
  - **—Built-in Snoop Filter**
- **CPU Support**
  - **—Support for Intel386™ and Intel486™ Microprocessors**
  - **—Intel486™ Burst Reads at 0 Wait State**
  - **—Posted Writes at 0 Wait State**
- **196-Pin PQFP Package**
  (See Package Specification Order Number 240800, Package Type KU)

The 82359 DRAM Controller is a highly integrated advanced memory controller capable of supporting today's Intel386 and Intel486 high performance microprocessors. Its decoupled handshake protocol gives the 82359 independence over processor type and speed, allowing the system designer to implement a variety of CPU/cache combinations.

The 82359 implements a dual ported architecture by providing two independent address paths to main memory. This allows activity on each bus to run independently of the other, giving each greater bus throughput and decreased bus latency.

The 82359 provides address control, refresh generation, critical DRAM timing generation and, by working closely with two 82353 Advanced Data Path devices, provides a highly integrated 32-bit dual ported memory controller in just three VLSI components.



290378-1

*Diagram illustrates Standard Configuration

Intel386™ and Intel486™ are trademarks of Intel Corporation.

October 1992
Order Number: 290378-005

82359 DRAM Controller

CONTENTS PAGE

82359 BLOCK DIAGRAM 1-525

1.0 INTRODUCTION 1-526

2.0 PIN DESCRIPTION QUICK REFERENCE 1-528

3.0 REGISTER QUICK REFERENCE .. 1-534

4.0 HARDWARE OVERVIEW 1-547

4.1 Host Interface 1-548

4.2 Host Address, Cycle Definition ... 1-548

4.3 Memory Ownership Protocol 1-549

4.4 Start of Host Cycles 1-550

4.5 Host Cycle Types 1-550

4.6 Deterministic Cycles 1-550

4.7 Non-Deterministic Cycles 1-551

4.8 Locked Cycles 1-551

4.9 Programmable Attributes 1-551

4.10 Snooping 1-551

5.0 SYSTEM PORT 1-552

5.1 System Interface Overview 1-553

5.2 System Address, Cycle Definition 1-553

5.3 System Control Signals 1-554

5.4 Memory Ownership Protocol 1-555

5.5 System Cycles 1-556

5.6 Deadlock Conditions 1-558

6.0 MEMORY INTERFACE 1-559

6.1 Memory Array Overview 1-559

6.2 Memory Interface Signals 1-562

6.3 DRAM Address Generation 1-564

6.4 RAS# Modes of Operation 1-567

6.5 Special Considerations 1-569

6.6 DRAM Refresh Generation 1-570

6.7 Decoupled Refresh Mode 1-570

6.8 Coupled Refresh Mode 1-571

CONTENTS PAGE

7.0 82353 ADVANCED DATA PATH INTERFACE 1-572

8.0 INTERNAL DELAY LINES 1-574

9.0 THROTTLE MECHANISMS 1-577

9.1 Introduction 1-577

9.2 Host-to-Memory Throttle and Watchdog (HMT, HMTW) 1-578

9.3 System-to-Memory Throttle (SMT) 1-578

9.4 Host-to-System Throttle and Watchdog (HST, HSTW) 1-579

10.0 SNOOP FILTER 1-579

11.0 CONCURRENCY 1-581

11.1 Concurrent Mode 1-581

11.2 Non-Concurrent Mode 1-581

12.0 LIM SUPPORT 1-581

13.0 THE INTERNAL REGISTERS 1-582

13.1 Programming the Internal Registers 1-582

13.2 Register Listing 1-585

13.3 Detailed Register Descriptions .. 1-586

14.0 DETAILED PIN DESCRIPTIONS .. 1-626

14.1 Host Port Interface 1-626

14.2 System Port Interface 1-631

14.3 Memory Interface 1-636

14.4 82353 Interface 1-638

14.5 Miscellaneous Decodes and Control Signals 1-639

1

CONTENTS

PAGE

15.0 FUNCTIONAL TIMING DIAGRAMS 1-643
15.1 Host to Memory Read—Page Hit 1-643
15.2 Host to Memory Read—Page Miss 1-644
15.3 Host to Memory Read—Row Miss 1-645
15.4 Host to Memory Burst Read (1-Way Interleave) 1-646
15.5 Host to Memory Burst Read (2-Way Interleave) 1-647
15.6 Host to Memory Burst Read (4-Way Interleave) 1-648
15.7 Host to Memory Write 1-649
15.8 Host to Memory Write—Single Byte 1-650
15.9 Host to System Single Dword Cycle 1-651
15.10 Host Burst to System Cycle ... 1-652
15.11 Accessing the 82359's Internal Registers 1-653
15.12 Snoop Cycle 1-655
15.13 Snoop Filter Example 1-656

CONTENTS

PAGE

16.0 ELECTRICAL CHARACTERISTICS 1-657
16.1 Absolute Maximums 1-657
16.2 D.C. Specifications 1-657
16.3 A.C. Specifications 1-658
16.4 Driver Characterization Data 1-682

17.0 MECHANICAL SPECIFICATIONS 1-686

18.0 THERMAL SPECIFICATIONS 1-686

19.0 NUMERICAL PIN LISTING 1-687

20.0 ALPHABETICAL PIN LISTING ... 1-688

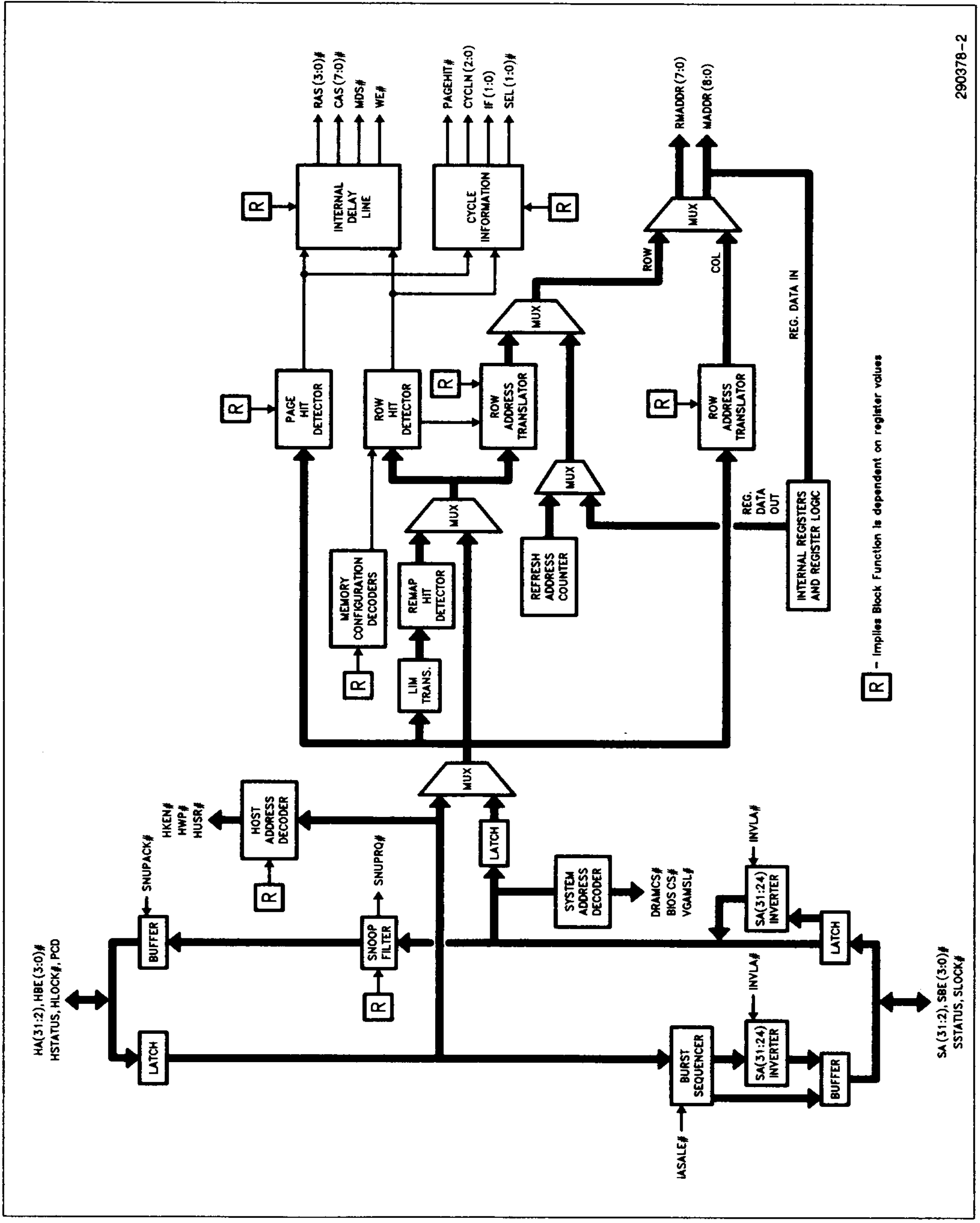
21.0 PINOUT DIAGRAM 1-689

APPENDIX A 1-690
General Register List 1-690
LIM Registers 1-692

APPENDIX B 1-693
Memory Sizing Algorithm 1-693
Revision Summary 1-695

82359 BLOCK DIAGRAM

Below is a simplified block diagram of the 82359 DRAM Controller. It is provided for illustrative purposes only.



1.0 INTRODUCTION

The 82359 DRAM Controller is a highly integrated EISA DRAM memory controller based on a dual ported memory architecture. It provides address and control signals for DRAM based main memory and it works very closely with two 82353 Advanced Data Path devices.

The 82359 may operate in one of two modes: (1) Standard Mode in which the 82359 connects directly to the EISA address bus; and (2) Buffered Mode, in which a new bus exists between the 82359 and the EISA bus and functions similar to the host bus. In this mode, the EISA bus is "buffered" from the 82359. For a full discussion of Buffered and Standard Mode, see the "82350DT System Architecture Overview" document.

The 82359 has two ports, or address gateways, to main memory; one exclusively for the host and one exclusively for EISA. This allows CPU activity to be isolated from EISA bus activity, allowing the host to run out of main memory at the same time system bus (EISA) activity is occurring. This dual ported architecture provides four routes which a cycle may follow: (1) Host to main memory; (2) Host to system slave; (3) System master to main memory and; (4) System address to host cache (for cache line invalidation).

One port, labeled "Host Port", provides a one-way path for host cycles to DRAM memory or to the system bus. It is capable of accepting a 32-bit host address and host cycle definition. From the address and cycle definition, the 82359 determines if the cycle is bound for main memory, in which case the 82359 executes a DRAM cycle, or if not to main memory, the 82359 forwards the cycle to the system bus. Although the host port is considered one-way in direction in that it is only capable of receiving host originated cycles, it does drive the host address lines when forwarding cache invalidation addresses to the host cache (if one exists).

The second port, labeled "System Port", acts as the gateway to/from the system bus. Unlike the host port, the system port is bi-directional, capable of sending as well as receiving 32-bit addresses and system bus cycle definitions. The system port accepts system bus cycles and, if the cycle is to an address contained in main memory, it executes the DRAM cycle. If the address of the system cycle is not contained in main memory, no action is taken by the 82359.

Since the 82359 was designed to support an EISA based expansion bus, it closely communicates with the 82358DT EISA Bus Controller (EBC). All host-to-system cycles are sent through the 82359 to the EBC for correct EISA/ISA cycle generation. All EISA bus activity is directly monitored and interpreted by the 82359, and the 82359 automatically acts upon EISA cycles to main memory without EISA protocol translation by the EBC.

The 82359 does not follow the typical ADS# and READY# protocol of the microprocessor. Instead, it uses a clockless protocol on both the host and system ports which isolates the CPU clock from the DRAM controller. This allows the 82359 to become CPU frequency independent.

A typical design would take the cycle definition (M/IO#, W/R#, D/C#) and ADS# of the CPU and interpret these to communicate with the 82359, telling it to start a cycle and what type of cycle is required. The 82359 decodes the address presented with the start of the cycle and returns a 3-bit code for the cycle length. From this cycle length, the protocol converter knows when to return READY# to the CPU.

The 82359 contains many programmable registers which control functions such as memory block enable/remap/shadow, DRAM timing generation, memory array population, and memory cycle length to name a few. These registers are typically programmed by the BIOS at power-up. It is through these registers that the 82359 achieves its flexibility.

Four registers are provided for memory array population information. The BIOS typically tests memory at power-up and provides DRAM SIMM size and population information to the 82359.

DRAM access times of 60, 70, or 80 ns are supported by the 82359. To facilitate the critical timings specific to each speed of the DRAM, the 82359 has programmable registers which access an internal delay line. Through these programmable timing registers, DRAM parameters such as RAS# precharge, RAS# to CAS# delay, etc. can be tailored to the DRAM's required times with 2.61 ns resolution.

Portions of the memory array may be individually disabled, remapped, write-only or read-only under programmable register control. Through the use of these registers, BIOS may be shadowed to DRAM. Also the memory map may be configured to "jump over" areas which contain other system functions (such as video, BIOS, etc.) by disabling portions of DRAM in 16k increments. Memory in the 512k–1M range may be disabled and remapped to the top of main memory in 64k blocks.

The 82359 provides four Programmable Attribute Map (PAM) registers to be used in systems which utilize caches on the host bus. Three bits of attribute are provided for each range: (1) Cache Enable, (2) Write Protect, and (3) a User-Defined bit. These registers allow software to determine the attributes for a programmable range size at a programmable starting address.

Eight LIM registers are provided for those systems which take advantage of the Lotus/Intel/Microsoft convention for expanded memory. These registers may be programmed to swap 16k pages of memory from anywhere in the lower 16M address range into and out of DOS accessibility.

The 82359 is designed to support write-through caches on the host bus. System write cycles are sent to the host cache as snoop cycles. Also, the 82359 performs "Snoop Filtering" which eliminates needless snoop cycles. Should a system write cycle occur to a location contained in the cache line which the 82359 invalidated by the previous snoop cycle, the 82359 will not broadcast the second, redundant snoop to the host. By eliminating redundant snoops, the host bus has increased bandwidth.

As EISA masters become more and more abundant, main memory accessibility becomes an increasingly important factor. With many EISA master devices installed in a system, the portion of memory bandwidth available to the CPU decreases significantly. To eliminate inefficient allocation of memory bandwidth, the 82359 has internal throttles which can be programmed to hold off memory ownership requests for a determined period of time so that others who desperately require memory bandwidth can have a greater time-slice than EISA arbitration allows. The net effect of these throttles allows the main memory ownership resource to be allocated for best system performance.

The 82359 provides two modes of DRAM refresh generation: (1) Coupled Refresh, in which the refresh timing is provided by the EISA bus, and (2) Decoupled Refresh, in which the 82359 refreshes main memory by generating the refresh request and address internally.

To facilitate the CPU frequency independence of the 82359, a new host bus protocol was devised. This protocol does not follow the synchronous ADS# and RDY# of the processor. Instead, it is asynchronous in nature in that it has no clock. This protocol is implemented by an external Programmable State Tracker (or PST) which converts the CPU's ADS# and RDY# protocol to the asynchronous protocol used by the 82359. This PST can typically be implemented in a two or three PLD solution.

Although the protocol is asynchronous, it does not detriment CPU to memory performance like other asynchronous protocols. This is achieved by the unique implementation of the protocol. The protocol defines two types of cycles; (1) the Deterministic Cycle, and; (2) the Non-deterministic Cycle. Deterministic cycles are cycles to main memory. The exact length of these cycles is known by the 82359 at the beginning of the cycle since it is aware of exactly how long that cycle to memory (page hit, page miss) will require for completion. The 82359 immediately relays that information to the host PST via a "DRAM Page Hit" indicator and a 3-bit code containing wait state information. From this, the host PST knows exactly when to send the RDY# to the CPU. Thus the RDY# is returned at the exact moment the memory cycle finishes and no synchronization penalty is incurred.

Non-deterministic cycles are host cycles to system bus slaves or locked cycles. Before these cycles can complete, the host must gain ownership of the system bus and thus, arbitration may be required. Since the 82359 does not know exactly how long the host must wait before gaining system bus ownership, or exactly how long the host-to-system cycle will require to complete (due various speeds of system slaves), the 82359 can not return an exact cycle length to the host CPU. Instead, an asynchronous signal is used to indicate the completion of the host- to-system cycle. In this case, a one CPU clock synchronization penalty is paid when returning RDY#. It is important to note that host-to-system cycles are the only cycles which pay this synchronization penalty and that the more important host-to-main memory cycles pay no synchronization penalty whatsoever.

2.0 PIN DESCRIPTION QUICK-REFERENCE

The following is a brief description of the interface pins of the 82359. For a more complete explanation of each pin, refer to the section entitled Detailed Pin Description.

Symbol	Type	Name and Function
HOST PORT INTERFACE		
HAS#	I	**HOST ADDRESS STROBE:** The falling edge of HAS# is the host cycle start indicator to the 82359 and causes the 82359 to latch HA(31:2), HBE(3:0)#, HM/IO#, HW/R#, HD/C#, HLOCK#, and PCD. HAS# rising edge indicates the end of the current host cycle and re-opens the latch.
HA(31:2)	I/O	**HOST ADDRESS:** The HA(31:2) signals provide the 82359 with its 30-bit host address. These address signals are inputs for host cycles and outputs when the 82359 drives snoop addresses to the host cache.
HBE(3:0)#	I	**HOST BYTE ENABLES:** Byte enable input bits indicate active bytes during host cycles.
HM/IO# HW/R# HD/C#	I	**HOST BUS DEFINITION:** HM/IO#, HW/R#, and HD/C# comprise the host cycle definition and are inputs to the 82359 from the host bus. These three signals define the current host bus cycle type.
HARDY	O	**HOST ASYNCHRONOUS READY:** HARDY is an asynchronous ready indicator from the 82359 to the host PST driven off of a decode of the host address and status. During non-deterministic cycles, the 82359 de-asserts HARDY indicating a "not ready" condition to the host PST. HARDY is used only for host-to-system cycles and locked host cycles.
HMREQ	O	**HOST MEMORY REQUEST:** HMREQ is asserted by the 82359 to the host PST to indicate the 82359 would like ownership of main memory. The 82359 requests ownership of main memory on behalf of Refresh or a system master.
HMACK	I	**HOST MEMORY ACKNOWLEDGE:** HMACK is an input from the host PST in response to a 82359 request of memory ownership (HMREQ). HMACK asserted indicates that the HMREQ has been honored and that the 82359 has memory ownership.
HBURST#/ CCRB2	O	**HOST BURST/CCRB2:** The HBURST#/CCRB2 is an output of the 82359 and may take on one of two functions. As HBURST, the pin functions as a decode of the host address and cycle definition to indicate to the host PST that the memory address being accessed is capable of a burst transfer. As CCRB2, this pin is a direct reflection of Cache Control Register, bit2. The HBURST#/CCRB2 pin's functionality is determined by the setting of the PCDOVERRIDE bit (Mode Register B, bit 1).
PCD	I	**PAGE CACHE DISABLE:** PCD is an input from the host to the 82359, typically driven from the i486 PCD output. The PCD input is used by the 82359 in determining the cacheability of addresses.
HLOCK#	I	**HOST LOCK:** The HLOCK# pin indicates that the current bus cycle is a locked cycle. When the current cycle is locked, the system bus will be arbitrated for and system bus ownership obtained before the 82359 runs the host cycle, irrespective of the destination of that cycle. System bus ownership will remain in the host's possession until the complete lock sequence is done. Locked cycles are always run as non-deterministic cycles.
SNUPRQ	O	**SNOOP REQUEST:** The 82359 asserts this signal to the host interface when it has a pending cache invalidation cycle. The 82359 de-asserts SNUPRQ from the falling edge of SNUPACK#.
SNUPACK#	I	**SNOOP ACKNOWLEDGE:** SNUPACK# is an input from the host PST to acknowledge the 82359's request to run a snoop cycle. The 82359 de-asserts it SNUPRQ from the falling edge of SNUPACK# and the snoop address is driven onto the host bus while SNUPACK# is asserted.

2.0 PIN DESCRIPTION QUICK-REFERENCE (Continued)

Symbol	Type	Name and Function
HOST PORT INTERFACE (Continued)		
HUSR# HWP# HKEN#	O	**HOST PROGRAMMABLE ATTRIBUTES:** HUSR#, HWP# and HKEN# are three programmable attribute bits available through the PAM registers. The attribute bits will reflect the setting of one of the Programmable Attribute Map registers if the host address falls within one of the registers programmed address ranges.
ST#	I	**START (Continue) HOST CYCLE:** If ST# is sampled inactive after a DRAM cycle has started, the cycle will be aborted. ST# is typically used by parallel caches to terminate the DRAM cycle if a cache hit is detected. Another way to look at the ST# signal is to consider it a "continue" signal, causing the 82359 to continue its current cycle as long as ST# remains asserted. **NOTE:** The ST# signals should be used to abort DRAM cycles only when the HP1 register (offset 10h) is programmed to a value of 0 through 4 (inclusive).
SYSTEM PORT INTERFACE		
SAS#	I/O	**SYSTEM ADDRESS STROBE:** SAS# indicates the beginning of an 82359 system-side cycle. SAS# becomes an output when the host is sending a cycle through the 82359 to the system bus. SAS# becomes an input when the host does not own the system bus. The falling edge of SAS# transparently latches SA(31:2), SBE(3:0)#, SD/C#, SW/R#, SM/IO#, and SLOCK#.
IASALE#	I	**INTERNAL ADDRESS TO SYSTEM ADDRESS LATCH ENABLE:** IASALE# controls the latching of host addresses which need to be driven to the system bus. It is an input typically driven by the EISA Bus Controller. IASALE# asserted causes the system address latch to become transparent and the address held in the host address latch will be driven to the system address lines. The rising edge of IASALE# closes the system address latch. During host-to-system burst cycles, the first rising edge of IASALE# causes the lead-off address to be latched and subsequent rising edges increment the system burst order counter which controls SA(5:2).
SHOLD	I	**SYSTEM HOLD:** SHOLD is an input to the 82359 which indicates that a system master or DMA is requesting ownership of the system bus. The 82359 will respond with a SHLDA when the 82359 has given ownership of the system bus to the requesting master.
SHLDA	O	**SYSTEM HOLD ACKNOWLEDGE:** SHLDA is an output from the 82359 acknowledging the SHOLD from the system and indicates that the 82359 has given-up the ownership of system bus. SHLDA will remain asserted as long as SHOLD remains asserted. When the 82359 asserts SHLDA, all 82359 system address and control signals will be tri-stated.
SBREQ/ SBREQEN	O	**SYSTEM BUS REQ/SYSTEM BUS REQ ENABLE:** The generation of SBREQ is dependent on EISA Cache Control Register, bit 6, and Mode Register B, bit 2. As SBREQEN, the function of this pin is a direct reflection of the EISA Cache Control Register, bit 6. As SBREQ, the function of this pin is to generate a request to the system arbiter on behalf of the host to get ownership of the system bus so as to run host-to-system cycles.
SARDY	I	**SYSTEM ASYNCHRONOUS READY:** SARDY is an input to the 82359 used to indicate a "not ready" condition of the system slaves when the 82359 is propagating host cycles to the system bus. The rising edge of SARDY indicates the end of the system cycle.

2.0 PIN DESCRIPTION QUICK-REFERENCE (Continued)

Symbol	Type	Name and Function
SYSTEM PORT INTERFACE (Continued)		
SBURST#	I/O	**SYSTEM BURST INDICATOR:** SBURST# is an input when system PST masters are running cycles. If SBURST# is asserted during system PST master cycles, the 82359 will run the appropriate number of memory cycles to satisfy the system line size. If SBURST# is not asserted, 82359 will run one and only one memory cycle. SBURST# becomes an output when the host is running bus cycles to the system bus and indicates to the system bus controller that the host would like to run multiple system bus cycles to satisfy the host burst.
SW/R# SM/IO# SD/C#	I/O	**SYSTEM CYCLE DEFINITION:** SD/C#, SM/IO#, and SW/R# are bi-directional system cycle definition signals and are defined the same as their host counterparts. These pins are inputs when the a system master owns the system bus. These signals become outputs when a host cycle is being forwarded to the system bus.
SBE(3:0)#	I/O	**SYSTEM BYTE ENABLES:** SBE(3:0)# are the byte enables for each one of the four bytes in the dword. The SBE#'s become outputs when host cycles are run to the system bus and are a reflection of the host port HBE#'s. When a system master owns the system bus, the SBE#'s become inputs receiving the BE#'s from the system cycle.
SA(31:2)	I/O	**SYSTEM ADDRESS:** The SA lines provide a 30-bit address port into/out of the 82359. They become outputs for host-to-system cycles and provide the cycle address to the 30-bit system address bus. SA(5:2) are also controlled by the internal system burst sequencer and provide the correct address for host burst to system cycles. The SA lines become inputs whenever a system master owns the system bus. The system address will be latched on the falling edge of SAS# for system PST masters or START#, IORC#, or IOWC# (delayed internally by 30 ns) for EISA/ISA bus masters.
SLOCK#	I/O	**SYSTEM LOCK CYCLE INDICATOR:** SLOCK# is an input to the 82359 when a system master owns the system bus and indicates that the master is running a locked cycle. SLOCK# is an output when the host owns the system bus. The assertion of SLOCK# during a system master cycle causes all throttles to be overridden.
SMREQ	O	**SYSTEM MEMORY REQUEST:** The 82359 will assert SMREQ to system PSTs when it or the host wants to own memory. The 82359 will immediately assert SMREQ whenever the system does not request memory ownership (i.e., the default state of SMREQ is asserted). SMREQ is not typically used in Standard Mode.
SMACK	I	**SYSTEM MEMORY ACKNOWLEDGE:** SMACK is controlled by the system PST and when asserted indicates that the system PST has released memory ownership to the 82359. When SMACK becomes de-asserted, the system PST master has memory ownership. SMACK should be pulled high in Standard Mode.
INVLA#	I	**INVERTED LA ADDRESS LINES:** This is a strap which indicates to the 82359 to treat the most significant byte of the system address, SA(31:24), as inverted. If INVLA# is tied low, all SA(31:24) address bits are inverted before being driven or read from the system bus. If this strap is high, SA(31:24) are treated as non-inverted.
BCLK	I	**EISA BUS CLOCK:** BCLK provides the 82359 with a reference for sampling EISA specific signals. Since the EISA bus is synchronous to BCLK, the 82359 samples EISA events synchronous to BCLK edges without regard to frequency or duty cycle.
START#	I	**EISA START# SIGNAL:** START# indicates the start of an EISA cycle. A bus master asserts START# after SA(31:2) and SM/IO# become valid, and negates START# on a rising edge of BCLK after one BCLK period.
CMD#	I	**EISA CMD# SIGNAL:** This input is the EISA CMD# signal monitored by the 82359 in directly tracking EISA master cycles. CMD# controls the bus data timings and its rising edge signals the end of the current EISA cycle.

2.0 PIN DESCRIPTION QUICK-REFERENCE (Continued)

Symbol	Type	Name and Function
SYSTEM PORT INTERFACE (Continued)		
MRDC#	I	**MEMORY READ COMMAND:** MRDC# is an ISA signal monitored by the 82359 in Non-concurrent Mode. It indicates to the 82359 that an ISA master wants to run a cycle. In Concurrent Mode, this signal is a "don't care" since only EISA signals are monitored.
MWTC#	I	**MEMORY WRITE COMMAND:** MWTC# is an ISA signal monitored by the 82359 in Non-concurrent Mode. The ISA master will assert MWTC# to indicate that the slave may latch the data bus. This is a "don't care" for Concurrent mode, since the 82359 monitors only EISA signals.
MSBURST#	I	**EISA BURST CYCLE INDICATOR:** MSBURST# is the EISA burst cycle indicator monitored by the 82359. When the 82359 samples MSBURST# asserted, the 82359's EISA burst state machine is activated and all subsequent memory cycles are run as page hits.
MEMORY INTERFACE SIGNALS		
RAS(3:0)#	O	**ROW ADDRESS STROBE:** The 82359 provides four RAS# signals for the DRAM array, one per row. RAS0#, RAS1#, RAS2# and RAS3# are connected to row 0, 1, 2 and 3 of the memory array respectively. The assertion of RAS# signals is dependent on memory array population and mode of operation.
CAS(7:0)#	O	**COLUMN ADDRESS STROBE:** The 82359 provides eight CAS# signals for the DRAM array. These eight CAS lines are broken-up into two groups of four; CAS(3:0)# for row1 and row2 to share, CAS(7:4)# for row 2 and row 3 to share. The CAS# signals are "byte based" meaning that CAS0# is connected to byte 0 of all eight dwords in the two rows, CAS1# to byte 1 of all eight dwords, etc.
WE(3:0)#	O	**MEMORY WRITE ENABLES:** The WE# signals select one of four dwords within a row during write operations and remain de-asserted for read operations. The WE#'s are also used for 82353 output enables for driving data to the DRAMs during memory writes.
MADDR(8:0)	I/O	**MEMORY ADDRESS:** The MADDR(8:0) are the nine address bits common to all rows of the memory array. Typically these address lines are outputs, providing row and column address information to the DRAM array. As a second function, eight of these bits, MADDR(7:0), become the slave port used in programming the 82359's internal registers.
RMADDR(7:0)	O	**ROW SPECIFIC MEMORY ADDRESS:** RMADDR(7:0) are row-specific DRAM address bits. In addition to the common memory address line of MADDR(8:0), the 82359 provides a second group of row-specific memory address lines. These eight bits are broken into four groups of two and each group connects to the corresponding row of the memory array, giving each row eleven total address bits.
82353 DATA PATH INTERFACE SIGNALS		
H/S#	O	**HOST/SYSTEM SELECT:** The H/S# output indicates which port currently has ownership of main memory. H/S# is driven by the 82359 to the 82353 Data Path for data routing and control.
SEL(1:0)	O	**SELECT SIGNALS:** The SEL(1:0) determine which one of the four possible dwords latched into the 82353 is requested by the current cycle. Should the cycle be a burst, SEL(1:0) points to the lead-off dword and the remaining sequence of dwords is known by the 82353 (since the i486 burst sequence is fixed, it can be determined by the lead-off address).
MDS#	O	**MEMORY DATA STROBE:** The 82359 generates MDS# to control data latching in the 82353 Data Path. During memory read cycles, MDS# falls with CAS# and after a precise delay (controlled by an internal delay line in the 82359), goes high causing the memory data to be latched into the 82353.

2.0 PIN DESCRIPTION QUICK-REFERENCE (Continued)

<table>
<tr><th>Symbol</th><th>Type</th><th>Name and Function</th></tr>
<tr><td colspan="3"><b>82353 DATA PATH INTERFACE SIGNALS</b> (Continued)</td></tr>
<tr><td>HIOE#</td><td>O</td><td><b>HOST INTERNAL OUTPUT ENABLE:</b> The 82359 uses HIOE# (along with MIOE#) to configure the 82353 Data Path for the current cycle. HIOE# is asserted during host write cycles to enable the host data port to drive the internal bus of the 82353.</td></tr>
<tr><td>MIOE#</td><td>O</td><td><b>MEMORY INTERNAL OUTPUT ENABLE:</b> The 82359 uses MIOE# (and HIOE#) to configure the 82353 Data Path for the current cycle. The 82359 will assert MIOE# and de-assert HIOE# when memory has been selected as the source of data (i.e., memory read cycles). The combination of MIOE# and HIOE# de-asserted causes the 82353's system port to drive its internal bus.
<table>
<tr><th>HIOE#</th><th>MIOE#</th><th>Data Source</th></tr>
<tr><td>0</td><td>X</td><td>Host</td></tr>
<tr><td>1</td><td>0</td><td>Memory</td></tr>
<tr><td>1</td><td>1</td><td>System</td></tr>
</table></td></tr>
<tr><td colspan="3"><b>CHIP SELECTS AND OTHER SIGNALS</b></td></tr>
<tr><td>IF(1:0)</td><td>O</td><td><b>INTERLEAVE FACTOR:</b> IF(1:0) reflect the dword interleave factor of the current row of the memory array being accessed by the 82359. The dword interleave factor is programmed into the configuration registers on a row basis when the system is booted (derived from a BIOS check of main memory).
<table>
<tr><th>IF(1:0)</th><th>Dword Interleave Factor</th></tr>
<tr><td>00</td><td>4-Way</td></tr>
<tr><td>01</td><td>2-Way</td></tr>
<tr><td>10</td><td>1-Way</td></tr>
<tr><td>11</td><td>System Cycle</td></tr>
</table></td></tr>
<tr><td>REFRESH#</td><td>I</td><td><b>SYSTEM REFRESH:</b> REFRESH# asserted indicates that a system device is running a refresh cycle. If the 82359 is programmed for Coupled Refresh, REFRESH# asserted causes the 82359 to execute a refresh cycle to main memory. If the 82359 is programmed for Decoupled Refresh, the REFRESH# input is ignored.</td></tr>
<tr><td>PER#</td><td>I</td><td><b>PARITY ERROR:</b> PER# is a parity error input to the 82359. This input is used by the 82359 to generate PERSTB# only for non-PST initiated read cycles (DMA, EISA, ISA masters).</td></tr>
<tr><td>PERSTB#</td><td>I/O</td><td><b>PARITY STROBE:</b> PERSTB# is the main memory parity error indication. It is an open-drain signal and will be driven by the 82359 when a parity error is generated by the 82353 Data Path during non-PST master (DMA, EISA, ISA masters) read cycles to main memory. Signaling of a parity error causes the contents of the Parity Error Trap Registers to be frozen.</td></tr>
<tr><td>MRDY</td><td>I/O</td><td><b>MEMORY READY:</b> MRDY is an open collector output from the 82359 to the EISA bus signal EXRDY. The 82359 will de-assert MRDY when wait states need to be inserted in system cycles. MRDY is monitored during EISA/ISA cycles by the EISA State Tracker portion of the 82359.</td></tr>
</table>

2.0 PIN DESCRIPTION QUICK-REFERENCE (Continued)

Symbol	Type	Name and Function
CHIP SELECTS AND OTHER SIGNALS (Continued)		
BIOSCS#	O	**BIOS CHIP SELECT:** BIOSCS# is an output from the 82359's system address decoder to indicate the BIOS ROM on the system bus is being accessed. BIOSCS# responds to both addresses 000E0000h to 000FFFFFh and FFFE0000h to FFFFFFFFh.
VGAMSL#	O	**VGA MEMORY SELECT:** VGAMSL# is an output from the system address decoder to indicate the current system address resides in VGA memory space. VGA memory is that which resides within the address range 000A0000h to 000BFFFFh.
DRAMCS#	O	**DRAM CHIP SELECT:** DRAMCS# is an output from the 82359 by the system address decoder to indicate that the address on the SA(31:2) lines exists in main memory.
LOCKEN#	O	**LOCK ENABLE:** This is a static output that is a direct reflection of the Lock Enable bit (bit 4 of the Cache Control Register). This pin is used to enable or disable the LOCK# signal of the 386 from reaching the cache controller. This signal is not used in i486 systems.
DEN#	O	**DATA ENABLE:** This pin is asserted in response to host-to-system cycles which program the 82359's internal registers and, if enabled by Mode Register A, bit 3, in response to a 486-initiated cache flush cycle. Specifically, DEN# is asserted during all host I/O write cycles to the Index and Data registers (typically 22h or 23h). When DEN# is driven by the 82359, the MADDR(7:0) bus carries 82359 internal register data.
SPEED(1:0)	I/O	**DRAM SPEED:** These are open drain pins accessible via the 82359's DRAM Speed Register. These bits can be used to reflect the binary code for the speed of DRAM in the memory array.
CYCLN(2:0)	O	**CYCLE LENGTH FEEDBACK:** CYCLN(2:0) reflects a 3-bit code for the required number of wait states for the current host or system deterministic cycle. These bits are driven directly from the values programmed in the CYCLN registers.
PAGEHIT#	O	**PAGE HIT:** This output is asserted by the 82359 when it detects that the current memory cycle has resulted in a DRAM page hit. This signal is available slightly before the CYCLN(2:0) signals and should be monitored by both host and system PSTs to indicated that the current cycle is the fastest memory cycle capable of occurring.
OSC	I	**REFERENCE FREQUENCY CLOCK:** This pin is driven by a 40 MHz, 50% ±10% duty cycle input used by the internal programmable delay line and for internal signal synchronization purposes.
RESET	I	**POWER-ON RESET:** RESET causes all programmable registers and state machines to be set to the initial state. This input should be activated only during the power-up sequence.
TEST#	I	**TEST:** This pin is used for testing only and should be tied to V_{CC} through a 10K resistor for normal 82359 operation.

3.0 REGISTER QUICK REFERENCE

*Indicates the bit position is undefined. These bits should be programmed as logical 0. Also, these bits should be masked when reading the register as their logic state is not guaranteed.

Default values are shown in parenthesis.

ROW0 CONFIGURATION REGISTER

Offset:	00h	
Default:	*000**01b	
Access:	W/R	
Bit Descriptions:		
1:0	DRAM Size	(256k)
6:4	Row Population	(Dword 0)

ROW1 CONFIGURATION REGISTER

Offset:	01h	
Default:	*111**01b	
Access:	W/R	
Bit Descriptions:		
1:0	DRAM Size	(256k)
6:4	Row Population	(Empty)

ROW2 CONFIGURATION REGISTER

Offset:	02h	
Default:	*111**01b	
Access:	W/R	
Bit Descriptions:		
1:0	DRAM Size	(256k)
6:4	Row Population	(Empty)

ROW3 CONFIGURATION REGISTER

Offset:	03h	
Default:	*111**01b	
Access:	W/R	
Bit Descriptions:		
1:0	DRAM Size	(256k)
6:4	Row Population	(Empty)

DRAM SPEED REGISTER

Offset:	04h	
Default:	******11b	
Access:	W/R	
Bit Description:		
1:0	DRAM Speed	(Slowest)

LINE SIZE REGISTER

Offset:	05h	
Default:	*100*100b	
Access	W/R	
Bit Descriptions:		
2:0	Host Line Size	(16 Bytes)
6:4	System Line Size	(16 Bytes)

3.0 REGISTER QUICK REFERENCE (Continued)

RAS# MODE REGISTER

Offset:	06h	
Default:	0**1**01b	
Access:	W/R	
Bit Descriptions:		
1:0	RAS# Mode	(1 Active RAS# Mode)
4	Reserved	(Must be "1")

BLOCK CACHE ENABLE REGISTER

Offset:	07h	
Default:	*****111b	
Access:	W/R	
Bit Descriptions:		
0	Video RAM Area HKEN#	(Non-cacheable)
1	Expansion ROM Area HKEN#	(Non-cacheable)
2	BIOS Area HKEN#	(Non-cacheable)

MODE REGISTER A

Offset:	08h	
Default:	00*01000b	
Access:	W/R	
Bit Descriptions:		
0	Stepping Indicator	(Read Only, A-2)
1	Host Status 386/486#	(486)
2	System Status 386/486#	(486)
3	Flush Host Cache on Flush Status	(Enabled)
4	Burst Non-Cacheable Code Prefetches	(Disabled)
7:6	Refresh Mode (1:0)	(Coupled)

MODE REGISTER B

Offset:	09h	
Default:	***11101b	
Access:	W/R	
Bit Descriptions:		
0	Restricted Burst	(Enabled-Restricted)
1	PCD Override	(Disabled)
2	IREQ/EREQ#	(Internal SBREQ)
3	128K/64K# BIOS	(128K BIOS)
4	SMT Enable	(SMT Enabled)

MODE REGISTER C

Offset:	0Ah	
Default:	00*****0b	
Access:	W/R	
Bit Descriptions:		
0	Concurrent/Non-Concurrent#	(Non-Concurrent)
6	1BCLK/2BCLK# EISA BURST CYCLE	(2 BCLK)
7	2BCLK/3BCLK# EISA SINGLE CYCLE	(3 BCLK)

3.0 REGISTER QUICK REFERENCE (Continued)

HOST TIMING REGISTER

Offset:	10h	
Default:	*1101000b	
Access:	W/R	
Bit Descriptions:		
3:0	HP10	(52.20 ns)
6:4	HP1	(18.27 ns)

HOST TO SYSTEM DELAY REGISTER

Offset:	11h	
Default:	****1010b	
Access:	W/R	
Bit Descriptions:		
3:0	HP11	(67.86 ns)

SYSTEM TIMING REGISTER

Offset:	12h	
Default:	0101*101b	
Access:	W/R	
Bit Descriptions:		
2:0	SP10	(20.88 ns)
6:4	SP1	(20.88 ns)
7	Buffered Mode	(Standard Mode)

ROW PRECHARGE TIMING REGISTER

Offset:	13h	
Default:	****1110	
Access:	W/R	
Bit Description:		
3:0	P2	(80.91 ns)

ROW TIMING REGISTER

Offset:	14h	
Default:	01011000b	
Access:	W/R	
Bit Descriptions:		
3:0	P4	(83.52 ns)
7:4	P3	(46.98 ns)

COLUMN TIMING REGISTER

Offset:	15h	
Default:	01111100b	
Access:	W/R	
Bit Descriptions:		
3:0	P7	(83.52 ns)
7:4	P5	(28.71 ns)

3.0 REGISTER QUICK REFERENCE (Continued)

CAS# LOW TIMING REGISTER

Offset:	16h	
Default:	10000100b	
Access:	W/R	
Bit Descriptions:		
3:0	P9	(31.32 ns)
7:4	P8	(62.64 ns)

CAS# TO MDS# DELAY REGISTER

Offset:	17h	
Default:	***01010b	
Access:	W/R	
Bit Description:		
3:0	P6	(57.42 ns)

CHIP IDENTIFICATION REGISTER (CIR)

Offset:	21h	
Default:	11111111b	
Access:	W/R	
Bit Description:		
7:0	Chip ID (CID)	(Invalid Index)

INDEX RELOCATION REGISTER (IRR)

Offset:	22h	
Default:	00100010b	
Access:	W/R	
Bit Description:		
7:0	IRR	(22h)

DATA RELOCATION REGISTER (DRR)

Offset:	23h	
Default:	00100011b	
Access:	W/R	
Bit Description:		
7:0	DRR	(23h)

PARITY ERROR ADDRESS REGISTER

Offset:		2Bh:28h	
Default:		(See bit descriptions)	
Access:		R	
Bit Description:			
28h-	7:2	Parity Error A(7:2)	(000000**b)
29h-	7:0	Parity Error A(15:8)	(00000000b)
2Ah-	7:0	Parity Error A(23:16)	(00000000b)
2Bh-	7:0	Parity Error A(31:24)	(00000000b)

3.0 REGISTER QUICK REFERENCE (Continued)

PARITY ERROR STATUS REGISTER

Offset:	2Ch	
Default:	000*0000b	
Access:	R	
Bit Descriptions:		
3:0	Parity Error BE(3:0)#	(0000)
5	Parity Error W/R#	(Read)
6	Parity Error D/C#	(Command)
7	Parity Error M/IO#	(IO)

CYCLE LENGTH REGISTER-READ PAGE HIT

Offset:	30h	
Default:	*111*111	
Access:	W/R	
Bit Descriptions:		
2:0	Host CYCLN	(7)
6:4	System CYCLN	(7)

CYCLE LENGTH REGISTER-READ PAGE MISS

Offset:	31h	
Default:	*111*111	
Access:	W/R	
Bit Descriptions:		
2:0	Host CYCLN	(7)
6:4	System CYCLN	(7)

CYCLE LENGTH REGISTER-READ ROW MISS

Offset:	32h	
Default:	*111*111	
Access:	W/R	
Bit Descriptions:		
2:0	Host CYCLN	(7)
6:4	System CYCLN	(7)

CYCLE LENGTH REGISTER-WRITE PAGE HIT

Offset:	33h	
Default:	*111*111	
Access:	W/R	
Descriptions:		
2:0	Host CYCLN	(7)
6:4	System CYCLN	(7)

CYCLE LENGTH REGISTER-WRITE PAGE MISS

Offset:	34h	
Default:	*111*111	
Access:	W/R	
Bit Descriptions:		
2:0	Host CYCLN	(7)
6:4	System CYCLN	(7)

3.0 REGISTER QUICK REFERENCE (Continued)

CYCLE LENGTH REGISTER-WRITE ROW MISS

Offset:	35h	
Default:	*111*111	
Access:	W/R	
Bit Descriptions:		
2:0	Host CYCLN	(7)
6:4	System CYCLN	(7)

LOWER MEMORY BLOCK ENABLE REGISTER 0

Offset:	40h	
Default:	11111111b	
Access:	W/R	
Bit Descriptions:		
0	00000h–0FFFFh	(Enabled)
1	10000h–1FFFFh	(Enabled)
2	20000h–2FFFFh	(Enabled)
3	30000h–3FFFFh	(Enabled)
4	40000h–4FFFFh	(Enabled)
5	50000h–5FFFFh	(Enabled)
6	60000h–6FFFFh	(Enabled)
7	70000h–7FFFFh	(Enabled)

LOWER MEMORY BLOCK ENABLE REGISTER 1

Offset:	41h	
Default:	******11b	
Access:	W/R	
Bit Descriptions:		
0	80000h–8FFFFh	(Enabled)
1	90000h–9FFFFh	(Enabled)

VIDEO RAM BLOCK ENABLE REGISTER 0

Offset:	42h	
Default:	00000000b	
Access:	W/R	
Bit Descriptions:		
1:0	A0000h–A3FFFh	(Remap)
3:2	A4000h–A7FFFh	(Remap)
5:4	A8000h–ABFFFh	(Remap)
7:6	AC000h–AFFFFh	(Remap)

VIDEO RAM BLOCK ENABLE REGISTER 1

Offset:	43h	
Default:	00000000b	
Access:	W/R	
Bit Descriptions:		
1:0	B0000h–B3FFFh	(Remap)
3:2	B4000h–B7FFFh	(Remap)
5:4	B8000h–BBFFFh	(Remap)
7:6	BC000h–BFFFFh	(Remap)

3.0 REGISTER QUICK REFERENCE (Continued)

EX ROM BLOCK ENABLE REGISTER 0

Offset:	44h	
Default:	00000000b	
Access:	W/R	
Bit Descriptions:		
1:0	C0000h-C3FFFh	(Remap)
3:2	C4000h-C7FFFh	(Remap)
5:4	C8000h-CBFFFh	(Remap)
7:6	CC000h-CFFFFh	(Remap)

EX ROM BLOCK ENABLE REGISTER 1

Offset:	45h	
Default:	00000000b	
Access:	W/R	
Bit Descriptions:		
1:0	D0000h-D3FFFh	(Remap)
3:2	D4000h-D7FFFh	(Remap)
5:4	D8000h-DBFFFh	(Remap)
7:6	DC000h-DFFFFh	(Remap)

BIOS AREA BLOCK ENABLE REGISTER 0

Offset:	46h	
Default:	10101010b	
Access:	W/R	
Bit Descriptions:		
1:0	E0000h-E3FFFh	(Copy)
3:2	E4000h-E7FFFh	(Copy)
5:4	E8000h-EBFFFh	(Copy)
7:6	EC000h-EFFFFh	(Copy)

BIOS AREA BLOCK ENABLE REGISTER 1

Offset:	47h	
Default:	10101010b	
Access:	W/R	
Bit Descriptions:		
1:0	F0000h-F3FFFh	(Copy)
3:2	F4000h-F7FFFh	(Copy)
5:4	F8000h-FBFFFh	(Copy)
7:6	FC000h-FFFFFh	(Copy)

3.0 REGISTER QUICK REFERENCE (Continued)

REMAP ENABLE REGISTER

Offset:	4Eh
Default:	00111100b
Access:	W/R

Bit Descriptions:

0	80000h–8FFFFh	(Not Remapped)
1	90000h–9FFFFh	(Not Remapped)
2	A0000h–AFFFFh	(Remapped)
3	B0000h–BFFFFh	(Remapped)
4	C0000h–CFFFFh	(Remapped)
5	D0000h–DFFFFh	(Remapped)
6	E0000h–EFFFFh	(Not Remapped)
7	F0000h–FFFFFh	(Not Remapped)

SPLIT ADDRESS REGISTER

Offset:	83h:84h
Default:	00000000 1***1111b
Access:	W/R

Bit Descriptions:

83h-	7:0	Split Address A(31:24)	(0h)
84h-	3:0	Split Address A(23:20)	(15 Meg)
	7	Split Enable	(Disable)

PROGRAMMABLE ATTRIBUTE MAP REGISTER 0

Offset:	53h:50h
Default:	00000000 00000000 00000*** 00001110b
Access:	W/R

Bit Descriptions:

50h-	0	HKEN#	(Cacheable)
	1	HWP#	(Not Write Protected)
	2	HUSR#	(De-asserted)
	7:3	Block Size	(2K Block)
51h-	7:3	A(15:11)	(0h)
52h-	7:0	A(23:16)	(0h)
53h-	7:0	A(31:24)	(0h)

PROGRAMMABLE ATTRIBUTE MAP REGISTER 1

Offset:	57h:54h
Default:	00000000 00000000 00000*** 00001110b
Access:	W/R

Bit Descriptions:

50h-	0	HKEN#	(Cacheable)
	1	HWP#	(Not Write Protected)
	2	HUSR#	(De-asserted)
	7:3	Block Size	(2K Block)
51h-	7:3	A(15:11)	(0h)
52h-	7:0	A(23:16)	(0h)
53h-	7:0	A(31:24)	(0h)

3.0 REGISTER QUICK REFERENCE (Continued)

PROGRAMMABLE ATTRIBUTE MAP REGISTER 2

Offset:		5Bh:58h	
Default:		00000000 00000000 00000*** 00001110b	
Access:		W/R	
Bit Descriptions:			
50h-	0	HKEN#	(Cacheable)
	1	HWP#	(Not Write Protected)
	2	HUSR#	(De-asserted)
	7:3	Block Size	(2K Block)
51h-	7:3	A(15:11)	(0h)
52h-	7:0	A(23:16)	(0h)
53h-	7:0	A(31:24)	(0h)

PROGRAMMABLE ATTRIBUTE MAP REGISTER 3

Offset:		5Fh:5Ch	
Default:		00000000 00000000 00000*** 00001110b	
Access:		W/R	
Bit Descriptions:			
50h-	0	HKEN#	(Cacheable)
	1	HWP#	(Not Write Protected)
	2	HUSR#	(De-asserted)
	7:3	Block Size	(2K Block)
51h-	7:3	A(15:11)	(0h)
52h-	7:0	A(23:16)	(0h)
53h-	7:0	A(31:24)	(0h)

CACHE CONTROL REGISTER (CCR)

Offset:		85h	
Default:		*100*1*1b	
Access:		W/R	
Bit Descriptions:			
	0	Cache Enable#	(Non-Cacheable)
	2	External Cacheability Map Write Enable#	(Disabled)
	4	Lock Enable#	(Disabled)
	5	Snoop Enable#	(Snoop Enabled)
	6	SBREQ Enable#	(Disabled)

SYSTEM MEMORY THROTTLE (SMT)

Offset:		8Bh	
Default:		11111111b	
Access:		W/R	
Bit Description:			
	7:0	SMT	(FFh)

3.0 REGISTER QUICK REFERENCE (Continued)

HOST MEMORY THROTTLE

Offset:	8Ch	
Default:	00000000b	
Access:	W/R	
Bit Description:		
7:0	HMT	(00h)

HOST MEMORY THROTTLE WATCHDOG (HMTW)

Offset:	8Dh	
Default:	11111111b	
Access:	W/R	
Bit Description:		
7:0	HMTW	(FFh)

HOST SYSTEM THROTTLE (HST)

Offset:	8Eh	
Default:	00000000b	
Access:	W/R	
Bit Description:		
7:0	HST	(00h)

HOST SYSTEM THROTTLE WATCHDOG (HSTW)

Offset:	8Fh	
Default:	11111111b	
Access:	W/R	
Bit Description:		
7:0	HSTW	(FFh)

RAM ENABLE REGISTER

Offset:	90h	
Default:	*******b	
Access:	R	
Bit Description:		
7:0	RAM Enable	(********)

RAM DISABLE REGISTER

Offset:	91h	
Default:	*******b	
Access:	R	
Bit Description:		
7:0	RAM Disable	(********)

ELAPSED TIME COUNTER (ETC)

Offset:		93h, 92h
Default:		******** *******b
Access:		R
Bit Description:		
92h-	7:0	ETC(7:0)
93h-	7:0	ETC(15:8)

3.0 REGISTER QUICK REFERENCE (Continued)

HOST MEMORY REQUEST TO OWNERSHIP (HMR)

Offset:		95h, 94h
Default:		******** *******b
Access:		R
Bit Description:		
94h-	7:0	HMR(7:0)
95h-	7:0	HMR(15:8)

SYSTEM MEMORY REQUEST TO OWNERSHIP (SMR)

Offset:		97h, 96h
Default:		******** *******b
Access:		R
Bit Description:		
96h-	7:0	SMR(7:0)
97h-	7:0	SMR(15:8)

HOST MEMORY OWNERSHIP (HMO)

Offset:		99h, 98h
Default:		******** *******b
Access:		R
Bit Description:		
98h-	7:0	HMO(7:0)
99h-	7:0	HMO(15:8)

SYSTEM BUS OWNERSHIP (SBO)

Offset:		9Bh, 9Ah
Default:		******** *******b
Access:		R
Bit Description:		
9Ah-	7:0	SBO(7:0)
9Bh-	7:0	SBO(15:8)

HOST REQUEST OF SYSTEM BUS (HRS)

Offset:		9Dh, 9Ch
Default:		******** *******b
Access:		R
Bit Description:		
9Ch-	7:0	HRS(7:0)
9Dh-	7:0	HRS(15:8)

MEMORY OWNERSHIP TRANSFER (MOT)

Offset:		9Fh, 9Eh
Default:		******** *******b
Access:		R
Bit Description:		
9Eh-	7:0	MOT(7:0)
9Fh-	7:0	MOT(15:8)

3.0 REGISTER QUICK REFERENCE (Continued)

LIM CONTROL REGISTER

Offset:		00h	
Default:		0*****00b	
Access:		W/R	
Bit Description:			
00h-	1:0	Active LIM Set	(Set 0)
	7	Global LIM Enable	(Disabled)

LIM PAGE TRANSLATION REGISTER 0

Offset:		81,80	
Default:		0*****00 00000000b	
Access:		W/R	
Bit Description:			
80h-	7:0	Translation Address A(21:14)	(00h)
81h-	1:0	Translation Address A(23:22)	(00b)
	7	LIM Page Enable	(Disabled)

LIM PAGE TRANSLATION REGISTER 1

Offset:		83,82	
Default:		0*****00 00000000b	
Access:		W/R	
Bit Description:			
82h-	7:0	Translation Address A(21:14)	(00h)
83h-	1:0	Translation Address A(23:22)	(00b)
	7	LIM Page Enable	(Disabled)

LIM PAGE TRANSLATION REGISTER 2

Offset:		85,84	
Default:		0*****00 00000000b	
Access:		W/R	
Bit Description:			
84h-	7:0	Translation Address A(21:14)	(00h)
85h-	1:0	Translation Address A(23:22)	(00b)
	7	LIM Page Enable	(Disabled)

LIM PAGE TRANSLATION REGISTER 3

Offset:		87,86	
Default:		0*****00 00000000b	
Access:		W/R	
Bit Description:			
86h-	7:0	Translation Address A(21:14)	(00h)
87h-	1:0	Translation Address A(23:22)	(00b)
	7	LIM Page Enable	(Disabled)

3.0 REGISTER QUICK REFERENCE (Continued)

LIM PAGE TRANSLATION REGISTER 4

Offset:		89,88	
Default:		0*****00 00000000b	
Access:		W/R	
Bit Description:			
88h-	7:0	Translation Address A(21:14)	(00h)
89h-	1:0	Translation Address A(23:22)	(00b)
	7	LIM Page Enable	(Disabled)

LIM PAGE TRANSLATION REGISTER 5

Offset:		8B,8A	
Default:		0*****00 00000000b	
Access:		W/R	
Bit Description:			
8Ah-	7:0	Translation Address A(21:14)	(00h)
8Bh-	1:0	Translation Address A(23:22)	(00b)
	7	LIM Page Enable	(Disabled)

LIM PAGE TRANSLATION REGISTER 6

Offset:		8D,8C	
Default:		0*****00 00000000b	
Access:		W/R	
Bit Description:			
8Ch-	7:0	Translation Address A(21:14)	(00h)
8Dh-	1:0	Translation Address A(23:22)	(00b)
	7	LIM Page Enable	(Disabled)

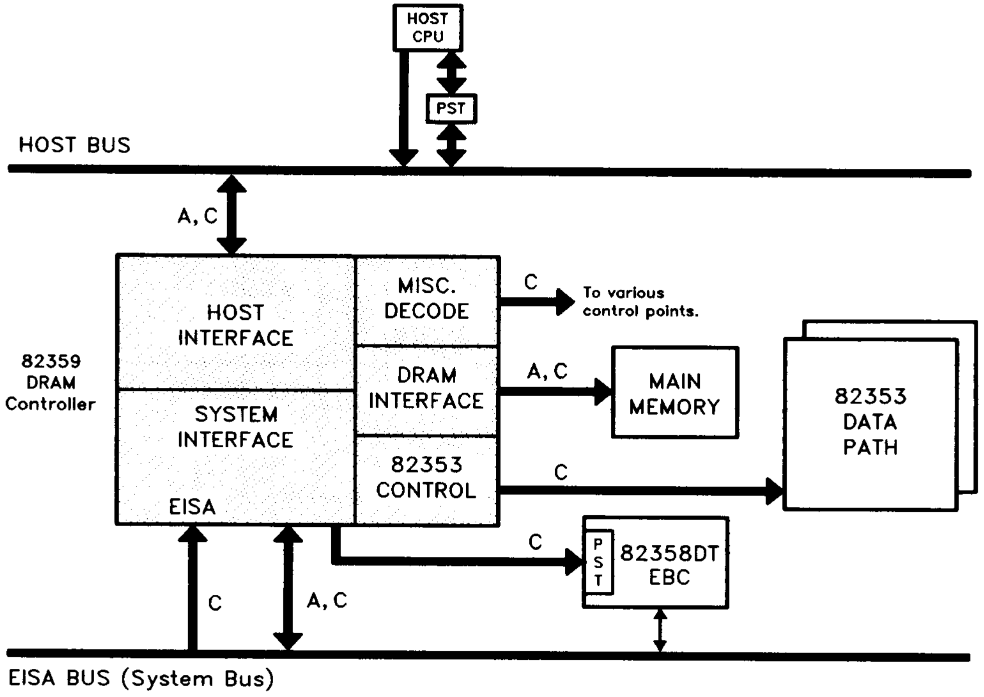
LIM PAGE TRANSLATION REGISTER 7

Offset:		8F,8E	
Default:		0*****00 00000000b	
Access:		W/R	
Bit Description:			
8Eh-	7:0	Translation Address A(21:14)	(00h)
8Fh-	1:0	Translation Address A(23:22)	(00b)
	7	LIM Page Enable	(Disabled)

4.0 HARDWARE OVERVIEW

The following section describes how the 82359 interfaces to external circuitry. The discussion will be presented in five parts: (1) the Host Interface, (2) the System Interface; (3) the Memory Interface; (4) the 82353 Data Path Interface, and; (5) the Miscellaneous Decode and Control signals. These functional partitions are illustrated in the following diagram.

82359 Functional Partition



290378–4

*Diagram illustrates Standard Configuration.

Figure 4-1. 82359 Functional Partition

4.1 Host Interface

The Host Interface is the gateway for host cycles to access main memory or system bus resources. It is capable of accepting a standard Intel 32-bit address and cycle definition (A(31:2), BE(3:0), M/IO#, W/R#, and D/C#). It also provides the signals necessary to communicate with the host PST for the request and grant of memory ownership, snoop cycles, and cache control.

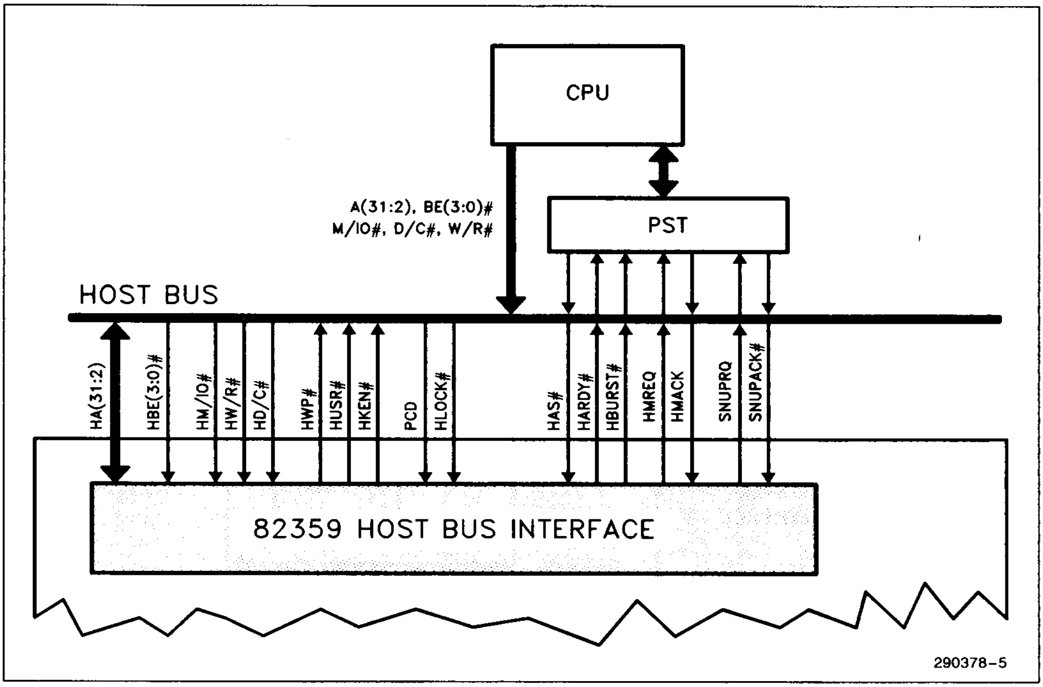


Figure 4-2. Host/PST Interface

4.2 Host Address, Cycle Definition

The host communicates its 32-bit cycle address and status on the host bus signals HA(31:2), HBE(3:0)#, HM/IO#, HD/C#, and HW/R#. The 82359 monitors these signals to determine the type and destination of the host originated cycle.

The 82359 uses an internal latch to capture the host cycle's address and status with the falling edge of HAS#. When HAS# is de-asserted, the host interface latch is transparent allowing host address and status to reach the internal decode circuitry. The falling edge of HAS# causes the latch to close and at this point the host address and status are free to change with no effect on any host cycle in progress. The rising edge of HAS# re-opens the latch and resets all host state machines.

Before an 82359 host cycle starts, the host drives its address and status on to the host bus. Since HAS# is de-asserted at this point, the address and status travel through the host interface latch to the internal circuitry. From the address and status, the 82359 immediately begins to decode the type and destination of the host cycle.

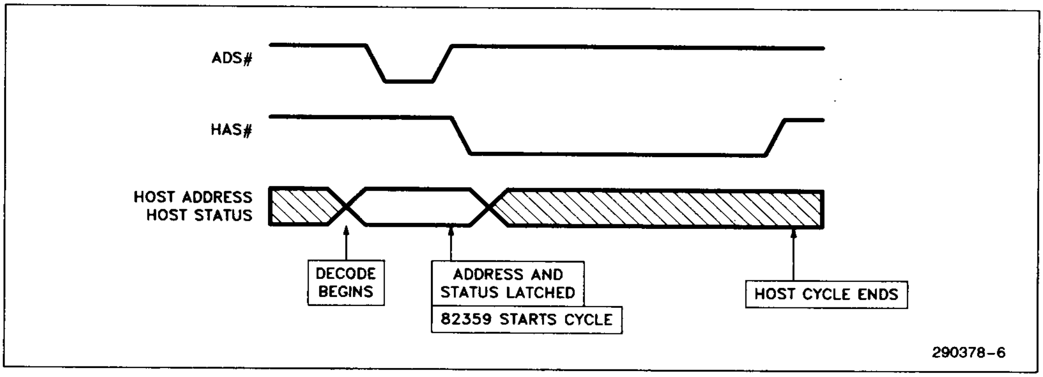


Figure 4-3. Host Address Cycle Definition

Should the host cycle be an IO access, special cycle, or the cycle's address not reside in main memory, the host interface unit determines that the destination of the cycle is to a resource which is not under its control and passes the cycle on to the system interface unit. If the cycle definition is that for a memory read or memory write to an address under the 82359's control, the 82359 will begin an access to main memory.

The host address is also used to transfer snoop addresses to the host cache. The 82359 indicates to the host PST that a snoop cycle is pending by asserting SNUPRQ. When the host PST is ready for the snoop address, it will assert SNUPACK# to the 82359. When this occurs, the 82359 drives the snoop address on to the host address bus. This is the only case where the 82359's Host Interface actively drives the host address bus. In all other cases the 82359 treats the host address as an input.

4.3 Memory Ownership Protocol

To allow the greatest control flexibility, the memory ownership arbitor is left as an external element, implemented in the host PST. This allows the system designer to implement a memory ownership algorithm which best suits the design objectives.

To implement the arbitration process between the host and the 82359 and communicate who is the current owner of main memory, two signals are used. HMREQ is driven by the 82359 to the host PST to indicate a system bus controller request for ownership, or to indicate the need of a refresh cycle. Note that the system bus does not directly arbitrate for memory ownership, but does so through the 82359.

The host PST will relinquish main memory control to the 82359 by asserting HMACK. From HMACK asserted, the 82359 knows that the host has given control to the 82359 to run either refresh or system cycles to main memory. Control remains in possession of the 82359 until it de-asserts HMREQ. The 82359 will de-assert HMREQ after the refresh, DMA or system master cycle is complete and HAS# is active (the host has a pending cycle). In essence, the 82359 is under control of the host PST and may only run system-to-memory or refresh cycles when the host PST has released main memory ownership.

4.4 Start of Host Cycles

Although a host cycle generally starts with HAS# falling, memory ownership may not be in the host's possession at that time. Should memory be owned by the 82359 (HMACK asserted) when HAS# goes low, the falling edge of HAS# acts like a "Host Cycle Request" and the 82359 does not begin the cycle until it is through with its current cycle and the PST de-asserts HMACK. From this two possible cycle start scenarios exist:

1. Memory is owned by the host at the falling edge of HAS#. The diagram below illustrates this example. In this case, the host has control of main memory as can be seen by HMACK de-asserted. Thus the falling edge of HAS# is both the host cycle request and host cycle start indicator.

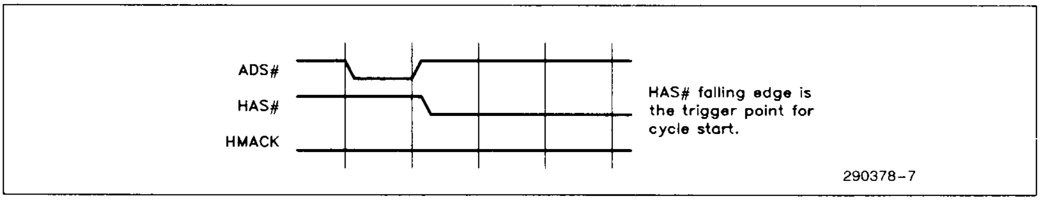


Figure 4-4. Start of Host Cycle (HAS# Initiated)

2. Memory is owned by the 82359 at the falling edge of HAS#. In this example, the host PST has previously given memory ownership to the 82359. Since the host does not own memory at the start of the host cycle, the falling edge of HAS# acts like a cycle request. The 82359 does not act on the host cycle until it is finished with its current memory cycles and it releases memory ownership back to the host with the de-assertion of HMREQ (which causes the host PST to de-assert HMACK). In this case, it is the falling edge of HMACK which acts as the cycle start trigger and causes the 82359 to begin processing the host cycle.

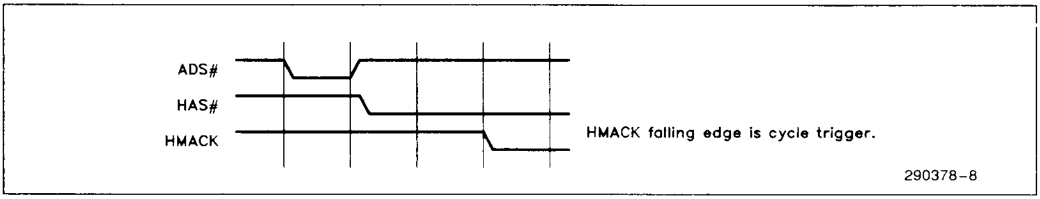


Figure 4-5. Start of Host Cycle (HMACK Initiated)

SPECIAL NOTE: The host PST must never assert HAS# and HMACK on the same clock edge.

4.5 Host Cycle Types

All host cycles fall into one of two categories; (1) Deterministic, or; (2) Non-deterministic. Which category a particular cycle may fall in to is determined by the host address and status.

4.6 Deterministic Cycles

A deterministic cycle is defined as a cycle from either the host or system bus which the 82359 can perform in a pre-determined amount of time. Examples of deterministic cycles include all accesses to main memory. For this type of cycle, the 82359 determines if the cycle is a DRAM page hit, page miss, or row miss and tells the host PST exactly how long the cycle will take to complete. For deterministic cycles, the 82359 keeps HARDY asserted and returns the programmed cycle length for the access via CYCLN(2:0) and PAGEHIT# to the host PST.

4.7 Non-Deterministic Cycles

A non-deterministic cycle is a host or system originated cycle for which the 82359 can not pre-determine the exact cycle length. An example of a non-deterministic cycle is a host access to the system bus. Since this is a dual ported memory controller, the system bus may be busy at the start of the host-to-system cycle. In this case the 82359 must go through system bus arbitration and gain bus ownership before the cycle can be executed. Secondly, once the bus is owned, the execution of the cycle takes an undetermined amount of time due to the various speed slaves which may exist on the bus.

To indicate to the host PST that the current host cycle is a non-deterministic cycle, the 82359 de-asserts HARDY. An unpredictable CYCLN(2:0) may be returned but should be ignored any time HARDY is de-asserted. The 82359 communicates the end of the non-deterministic cycle to the host PST with the rising edge of HARDY.

4.8 Locked Cycles

All locked cycles from the host are run as non-deterministic cycles. This allows the 82359 time to arbitrate for ownership of the system bus before performing the locked data transfer. Also note that all locked cycles are non-cachable and non-burstable.

4.9 Programmable Attributes

The 82359 provides three memory attribute bits which are available to the system designer. HWP#, HKEN#, and HUSR# are activated when the host address has been deemed as write protected, cache enabled, or user specified. The 82359 decodes the values of these bits directly off the host address. The programming of the address ranges to which these bits respond is done through the four Programmable Attribute Registers.

4.10 Snooping

The host interface is also responsible for the communication of snoop cycles (or cache invalidation cycles). When a system bus write occurs to main memory, a snoop cycle must be sent to the host cache to invalidate the cache line for which the address to be written exists. When the system bus write occurs, the 82359 will assert its SNUPRQ to the host PST. The host PST returns its SNUPACK# to indicate that it is ready for the snoop address. This cause the 82359 to remove its SNUPRQ and drive the snoop address onto the host address bus. The de-assertion of SNUPACK# will cause the 82359 to return the host address pins to their tri-state condition.

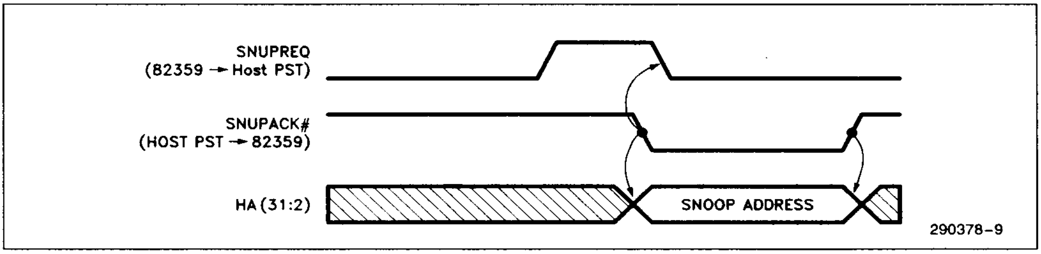
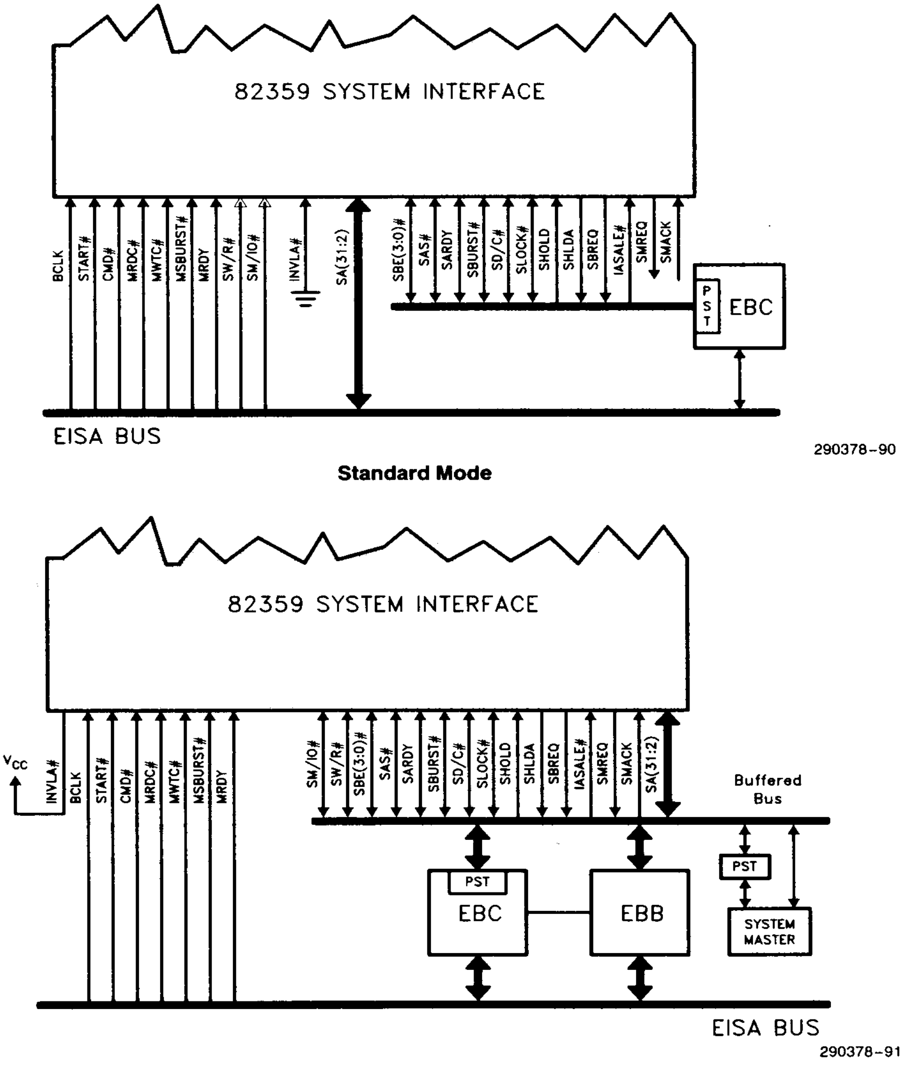


Figure 4-6. Snooping

5.0 SYSTEM PORT

The system port provides address and control signals needed for system bus cycles. Internal to the system interface is an EISA cycle tracker. This is used to monitor EISA activity and give the 82359 the information it needs to run EISA to memory cycles with the interpretation of the EBC. To communicate with the system bus, the 82359 uses an interface which is a superset of the host interface and also includes signals directly off the EISA bus for the internal EISA Cycle Tracker.



290378-90

Standard Mode

290378-91

Buffered Mode

Figure 5-1. System Port Interface

5.1 System Interface Overview

The system interface is responsible for communication between the 82359 and the system bus controller (EBC). Its responsibilities include transferring host-to-system cycles to the system bus controller, indicating the end of non-deterministic cycles, and providing a means for system PST masters to gain main memory ownership. To accomplish these tasks, the system interface uses a protocol that is very much like the host bus protocol.

The 82359 is capable of running in two different system interface modes. In Standard Mode, the 82359's address and status signals connect directly to the EISA address bus and no System PST masters are supported. In Buffered Mode, the system interface supports both the EISA bus and a Buffered Bus on the system interface. The Buffered Mode allows the 82359's address to connect directly to a "Buffered Bus" which supports system PST masters. The system address reaches the EISA bus through a set of address buffers under control of the EBC. This optional Buffered bus provides support for system PST masters and looks and operates much like the host bus, using SAS#, CYCLN(2:0), and PAGEHIT to run deterministic cycles to main memory.

5.2 System Address, Cycle Definition

The system interface uses a 30-bit address, SA(31:2), and four byte enable bits, SBE(3:0)#, for system address generation. This address port is bi-directional in that it drives the host address onto the system bus for host-to-system cycles, and receives the system address for all system originated cycles.

The system interface drives the system address for host-to-system cycles. If the system bus is not currently in use at the time of the host-to-system cycle (as indicated by the 82359 driving SHLDA low), the 82359 is free to run the cycle to the system bus without going through EISA bus arbitration. If SHLDA is asserted, the 82359 has given system bus ownership to a system master and the 82359 enters into EISA arbitration on behalf of the host by asserting SBREQ to the ISP. (SBREQ is typically connected to CPUMISS# of the ISP.) Once arbitration has taken place and the host has gained the system bus ownership, the 82359 drives the host address onto the system address. The integrated high-output drivers of the 82359 allow it to directly drive the EISA address bus with no external buffering.

The system interface receives the system address through a transparent address latch. This latch is open during system bus idle time. For EISA masters, it is closed from the falling edge of START# for standard cycles or the rising edge of BCLK for burst cycles. For ISA masters, the latch is closed from the falling edge of MRDC# or MWTC#. Lastly, for system PST masters, the falling edge of SAS# is used. For all types of system bus masters, the actual latching of the system address occurs about 30 ns after the above mentioned events.

For host-to-system cycles, the host address is latched by HAS# falling, travels through the 82359 and is sent to the internal System Burst Sequencer where SA(5:2) are latched. From here the address is sent to the system address latch and driven to system bus where the system cycle is performed. If the host-to-system cycle is a host burst, the System Address Latch holds on to SA(31:6) from the leadoff cycle and the System Burst Sequencer increments SA(5:2) in an i486 burst sequence to generate subsequent system addresses of the burst (see the functional diagrams for more details). Thus, the lead-off address of the burst is supplied by the host and system addresses for the subsequent cycles of the host burst are generated by the system burst sequencer. Since four bits of address are controlled by the sequencer, a burst of up to 16 dwords is available (although four dwords is typical) before the System Burst Sequencer recycles.

The system interface uses SM/IO#, SD/C#, and SW/R# for system cycle definition. On a host-to-system cycle, these are outputs which are a direct reflection of the host cycle definition. For system originated cycles, they become inputs from the current system master.

One point which needs consideration is that the host interface may be either 386- or i486-like and the system interface may be 386- or i486-like. This could be a problem for processor special cycles, since the i486 and 386 differ in the way they decode M/IO#, W/R#, D/C# and BE(3:0) to define the special cycle types. To resolve this problem, the 82359 is capable of translating either a 386- or i486-like host interface to a 386- or i486-like system interface. The host and system processor type is programmable through Mode Register A, bit 2.

5.3 System Control Signals

The system interface uses several control signals to communicate with the system bus controller. IASALE# is an input to the 82359 which causes the latching of system addresses into the System Address Latch during host-to-system cycles. The first IASALE# of the host-to-system cycle causes the host address to be latched. During host burst cycles, the bus controller will generate subsequent IASALE# pulses, each causing the System Burst Sequencer to generate the next address of the burst sequence as described above. The following figure illustrates this action.

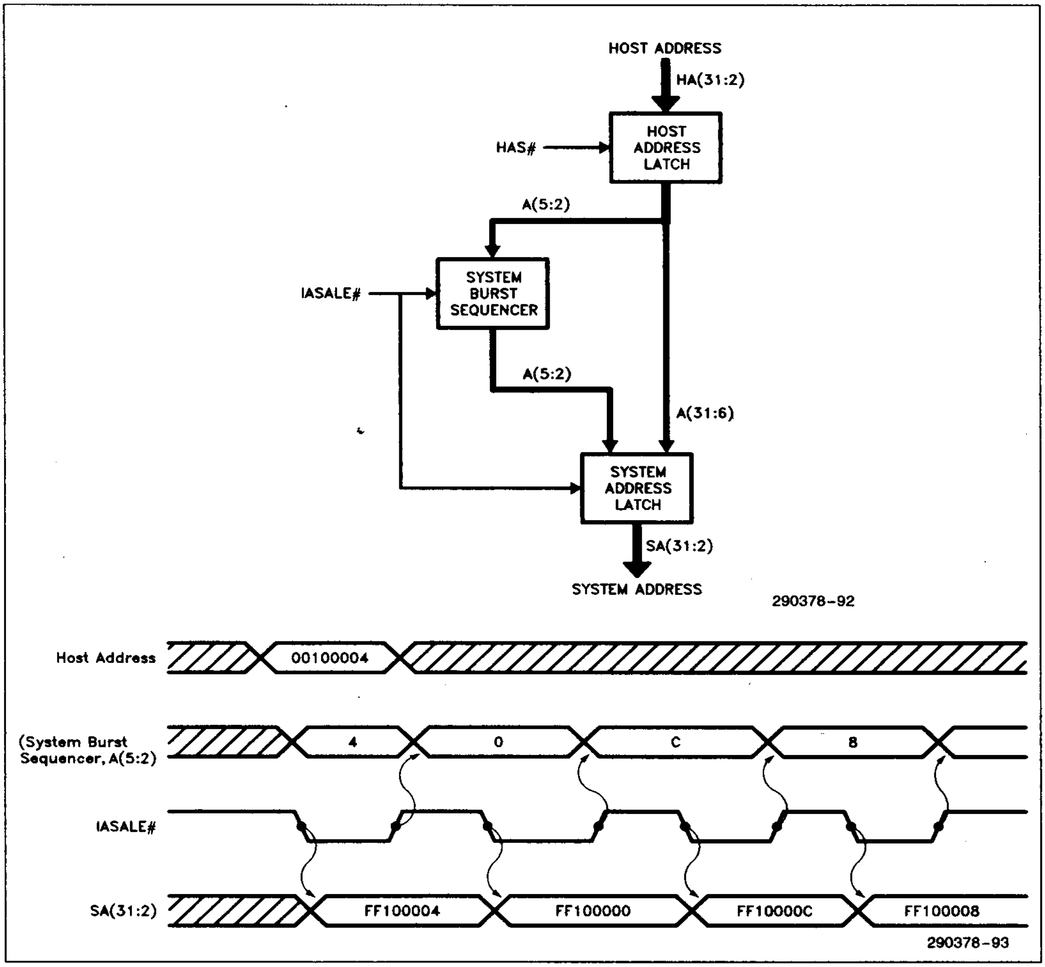


Figure 5-2. System Control Signals

The SHOLD/SHLDA signals function similar to the processor HOLD/HLDA signals. When an EISA master wishes to gain EISA bus ownership, the EBC generates an SHOLD to the 82359. If the 82359 is programmed for Concurrent Mode, the 82359 will immediately assert SHLDA back to the EBC and ISP if no host-to-system cycles are currently in progress. If a host-to-system cycle is occurring, SHLDA is asserted after the cycle completes and the host-to-system throttle has expired. In Non-concurrent Mode, SHLDA will not be returned until the 82359 has gained memory ownership from the host PST on behalf of the system bus.

5.4 Memory Ownership Protocol

The ownership of main memory occurs in different ways depending if Concurrency is enabled and if system Buffered Mode is enabled (system PST masters exist).

5.4.1 EISA MASTER CYCLES—CONCURRENT MODE

In this mode, the 82359 allows EISA masters to start cycles irrespective of memory ownership. When a master wins EISA arbitration, the EBC asserts SHOLD to the 82359. If no host-to-system cyles are occurring at that time, the 82359 immediately asserts SHLDA back to the ISP and EBC. If a host-to-system cycle is occurring, SHLDA is returned at the conclusion of that cycle (after the host-to-system throttle has expired). Once the master has taken control of the EISA bus it can begin its memory cycle.

5.4.2 EISA MASTER CYCLES—NON-CONCURRENT MODE

In Non-concurrent Mode, the 82359 will not return SHLDA to the ISP and EBC until memory ownership is obtained from the host PST by the 82359.

5.4.3 SYSTEM PST MASTERS

System PST masters implement a memory ownership protocol through SMREQ/SMACK which is very similar to HMREQ/HMACK of the host PST. The 82359 drives SMREQ to the system PST when other resources (host, decoupled refresh) require main memory ownership. The system PST returns SMACK asserted to acknowledge that main memory is not in its possession and SMREQ's default state is asserted. When a system PST master wishes to run cycles, it will assert SAS# to the 82359. The 82359 will go through the HMREQ/HMACK protocol to gain memory ownership on behalf of the system PST master. When the main memory ownership is obtained, the 82359 de-asserts SMREQ to the system PST. The system PST in turn de-asserts SMACK to indicate it has ownership. It is this falling edge of SMACK which is the start of the system PST cycle. The 82359 will re-assert the SMREQ upon sampling SMACK asserted to request memory back from the system PST master. It is up to the system PST master to be "self-throttling" in that it has the power to keep memory ownership for as long as it wishes by keeping SMACK low, although doing so may be detrimental to other system masters.

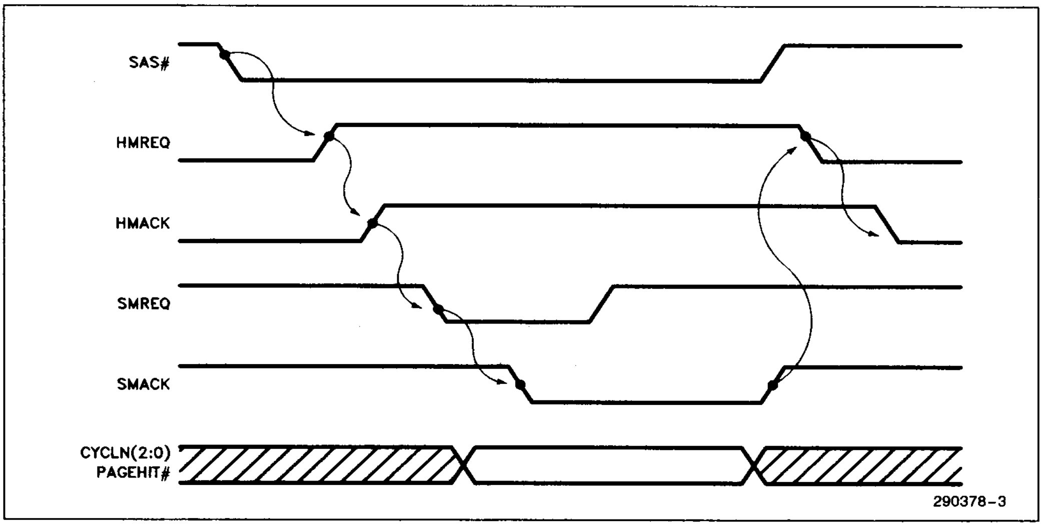


Figure 5-3. System PST Master Protocol

5.5 System Cycles

5.5.1 HOST-TO-EISA CYCLES

A host to EISA cycle begins with the host driving the address and asserting HAS#. The 82359 decodes the address and determines it is not to main memory, so the cycle is forwarded to the system bus and the 82359 de-asserts HARDY to the host PST. Before the system interface can run this cycle, it must determine if any system bus activity is occurring. This can be determined from the state of SHLDA. If SHLDA is de-asserted, no EISA device owns the bus and the cycle is run without incurring EISA arbitration. Should an EISA master request the bus from the 82359 via SHOLD while the host is using the EISA bus, the 82359 will not return the SHLDA until the host-to-system cycle finishes (after the host-to-system throttle has expired).

If SHLDA is asserted when a host-to-system cycle is requested, the 82359 must enter into EISA arbitration before the cycle can be executed. Specifically, the 82359 asserts SBREQ to the ISP's CPUMISS# which causes the current EISA owner to be preempted. The 82359 knows it has gained EISA ownership when the EBC deasserts SHOLD.

Once the 82359 has control of the system bus, whether directly or through EISA arbitration, IASALE# will be low allowing the system address to be driven onto the system bus and SAS# will be asserted to the EBC (82358DT). SAS# causes the EBC to begin the EISA cycle. The EBC will generate the appropriate EISA/ISA cycle, latch the system address with the rising edge of IASALE#, and de-assert SARDY to the 82359. The 82359 knows the system bus cycle is complete when it detects the rising edge of SARDY, which in turn causes the 82359 to generate the rising edge of HARDY to the host PST. Upon sampling HARDY asserted, the host PST removes HAS# causing SAS# to be de-asserted and ending the cycle.

If the 82359 is executing a host burst-to-system cycle, four single system bus cycles will be run in a back-to-back fashion. (Note: the host is doing the burst, not the EISA bus.) The EBC asserts IASALE# at the end of the leadoff, reopening the system address latch, and allowing the next address in the burst to be driven. SARDY rising indicates the end of the lead off cycle. Subsequent end-of-cycle indicators are generated by SDVLD to the 82353 which generates its own HARDY to the host PST (see the functional diagrams and the 82353 datasheet).

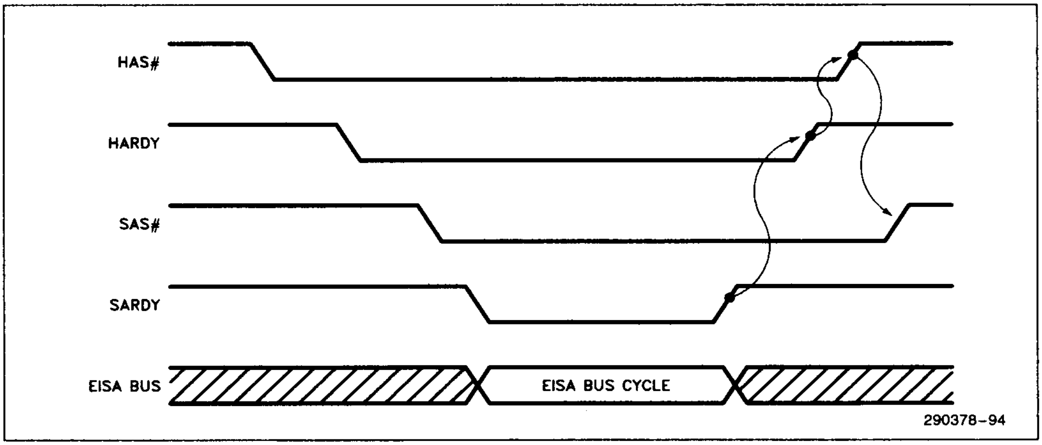


Figure 5-4. Host-to-EISA Cycle

5.5.2 EISA CYCLES

When an EISA device wants to run a cycle, it must first gain bus ownership through the ISP. The EBC and ISP will in turn assert SHOLD to the 82359. The 82359 will return SHLDA to the EBC and ISP immediately if no host-to-system cycle is occurring, or as soon as the current host-to-system cycles have completed. (In Non-concurrent Mode, the 82359 must first gain memory ownership from the host before SHLDA is returned.)

5.5.3 EISA Cycles Not to Main Memory

The master, upon sampling MACK<x> asserted, will begin its transfers by driving the cycle address and asserting START#. The 82359 begins decoding the address and if it is an address which is not contained in main memory, DRAMCS# is negated and the 82359 does not actively participate in the transfer.

The 82359 does however monitor all EISA cycle addresses so that it may send snoop cycles to the host cache for EISA memory write cycles. EISA burst memory writes may require that the 82359 drop MRDY for 1 BLCK so that it may keep snooping on pace with the burst addresses. Also the 82359 may drop MRDY during the CMD portion of the EISA cycle to allow the host to utilize the programmed throttle settings. Once the 82359 removes MRDY, the EISA cycle completes normally.

5.5.4 EISA Cycles to Main Memory

When the EISA master drives its address and the 82359 determines this address is to main memory, DRAMCS# is asserted and the 82359 will drop MRDY and begin the memory ownership protocol with the host PST. When the host PST returns HMACK, MRDY is remcved and the cycle completes as normal.

The first cycle of an EISA master is always run as a single, normal EISA cycle. At the end of the cycle, all RAS# signals are de-asserted.

For slower memory systems, not able to keep pace with EISA cycles, the 82359 allows the programmability of the number of BCLKs per EISA cycle. For non-burst EISA master cycles, the default length may be programmed for either 2 or 3 BCLKs per cycle. Should 3 BCLKs be selected, the 82359 will hold standard non-burst cycles off by inserting one wait state with the de-assertion of MRDY. For burst cycles, the cycle length may be programmed for either 1 or 2 BCLKs per cycle. Again, if 2 BCLKs is selected, masters will be held off one clock with MRDY.

5.5.5 ISA CYCLES

The sequence for ISA master cycles is very similar to EISA master cycles. The ISP must gain control of the EISA bus from the 82359 through the SHOLD/SHLDA protocol.

For ISA masters, the cycle start indicator is the falling edge of MRDC# or MWTC#. The 82359 will assert DRAMCS# if the ISA address is contained in the address range of main memory. Also, the 82359 must gain memory ownership from the host PST through the HMREQ/HMACK protocol. During this time the ISA master is held in wait states through the 82359 de-asserting MRDY (connected to EXRDY of the EISA bus which the EBC will translate to CHRDY of the ISA bus). Once memory ownership is obtained, the 82359 executes the cycle to DRAM. If the address is not to main memory, DRAMCS# is de-asserted and the ISA cycle proceeds without participation from the 82359. It will, however, generate snoop cycles to the host cache for ISA write cycles.

ISA signals MRDC# and MWTC# are monitored only in Non-concurrent Mode. In Concurrent Mode, the 82359 monitors only EISA signals and ISA signals are translated to these by the EBC.

If the system contains ISA masters which do not honor CHRDY, the 82359 must be programmed for Non-concurrent Mode. The reason is these masters run ISA cycles based on a predetermined cycle length and wait states can not be added to the cycle via CHRDY. This disallows the case where the 82359 holds off the master with CHRDY until memory ownership is obtained. These masters must be given memory ownership before they start their cycle. Running the 82359 in Non-concurrent Mode does exactly this. (See Concurrency for more information.)

5.5.6 SYSTEM PST MASTER CYCLES

System ownership of main memory in Buffered Mode occurs in two situations. For EISA masters, memory ownership follows that for Standard Mode described above. For system PST masters, SMREQ and SMACK are used in the protocol described above.

When a system PST master wants to run a cycle, it will provide its address and status, and assert SAS#. The falling edge of SAS# causes this address to be latched and, if this address is contained in the main memory address space, DRAMCS# will become asserted.

The EBC will ignore the system PST master cycles when DRAMCS# is asserted (DRAMCS# is typically tied to the EBC HLOCMEM# input) due to the cycle going to main memory rather than the EISA bus. Since this cycle requires ownership of main memory, the 82359 begins the SMREQ/SMACK ownership protocol. For system PST master cycles, SAS# acts as the cycle request and SMACK is the cycle start indicator. In system PST master cycles to main memory, the system PST uses CYCLN(2:0) and PAGEHIT# from the 82359 for determining the length of the cycle.

When DRAMCS# is not asserted, indicating that the system address is to an address other than main memory, the EBC is responsible for translating the system cycle to an EISA/ISA cycle. In this case, the EBC performs exactly as if it were a host-to-system cycle in that SARDY will be de-asserted until the EISA/ISA cycle is complete.

The 82359 performs snoops whenever system write cycles occur. Since system PST masters can potentially perform write cycles faster than the snoop protocol can complete, the 82359 will de-assert MRDY until the snoop has finished. The system PST should sample MRDY high before starting a subsequent write cycle to eliminate the possibility of snoops backing-up.

5.6 Deadlock Conditions

The way the System Interface is implemented gives rise to several deadlock conditions where both the host and system interface need a resource in the other's possession, but are unable to obtain that resource until the other cycle finishes. In these cases, both host and system cycles could permanently stall waiting for the other.

There are four cases in which these deadlocks may occur:

5.6.1 CASE #1: HOST-TO-SYSTEM CYCLE DURING A SYSTEM-TO-MAIN MEMORY CYCLE

In this case, HMACK is de-asserted so the host starts a host-to-system cycle. At the same time, for example, an EISA master begins a system-to-main memory cycle and drives the system address. Here, the host tries to preempt the EISA master but the master can not relinquish control of the system bus, since he is stalled in the middle of its cycle. On the other hand, the EISA master can not complete its cycle because it can not gain memory ownership until the host cycle is finished; thus the EISA master is stalled in wait states.

Solution: In this case, the 82359 recognizes that neither cycle can finish and takes control of main memory. The 82359 keeps HARDY de-asserted and allows the EISA master to run its cycle to main memory regardless of the HMREQ/HMACK protocol. This does not hurt the host master since it is already waiting for HARDY to return high, and the only effect to the host CPU is a longer non-deterministic cycle. This is the only case where a system master is allowed to access main memory without first obtaining memory ownership.

5.6.2 CASE #2: HOST LOCKED CYCLE TO MAIN MEMORY

The PST typically gives memory ownership to a system master in between host cycles. If a system master tries to gain memory ownership during a locked sequence of host cycles, the system master causes the lock to be invalid and it has the ability to access a memory location which the host is supposed to have exclusive access to during the locked sequence.

Solution: To alleviate the possibility of a system master interrupting a host locked sequence, the 82359 will automatically gain system bus ownership before starting a locked memory cycle. From this, it can be seen that all host locked cycles run uninterrupted as long as the host CPU owns the system bus.

5.6.3 CASE #3: SYSTEM MASTER LOCKED CYCLES

System masters may have problems running locked cycles with the throttles turned on. In this case, a system master can be running a locked cycle when the throttles allow the host to sneak cycles to main memory, thus violating a locked cycle.

Solution: When system locked cycles occur, the throttles are overridden and the host is not allowed to sneak in memory cycles. This eliminates the chance of the host accessing memory of which the system master should have exclusive ownership.

5.6.4 CASE #4: USAGE OF THE 82385 CACHE CONTROLLER

The 82385 Cache Controller is only able to snoop while in a HOLD state. Since all system bus writes must be snooped, the 82385 must be put on hold during all system master cycles.

Solution: To allow snooping to occur, the 82359 should be programmed for Non-concurrent Mode when there is an 82385 Cache Controller on the host bus.

6.0 MEMORY INTERFACE

The following sections describe how the 82359 communicates with the DRAM memory array. A brief overview of the memory array is presented followed by a discussion of the 82359 signals which generate DRAM cycles.

6.1 Memory Array Overview

The 82359 utilizes a DRAM memory structure of four rows by four columns. Each one of the 16 elements in the 4 x 4 array is considered a "memory element" of 36 bits (32 bits of data and 4 bits of parity). The maximum of four elements in each row allows for a 128-bit wide data line.

The four rows, although physically separate, are grouped logically into two groups of two. Each group of two makes one "memory bank". Row 0 and 1 are grouped together and labled Bank 0; Row 2 and 3 are grouped together and labled Bank 1.

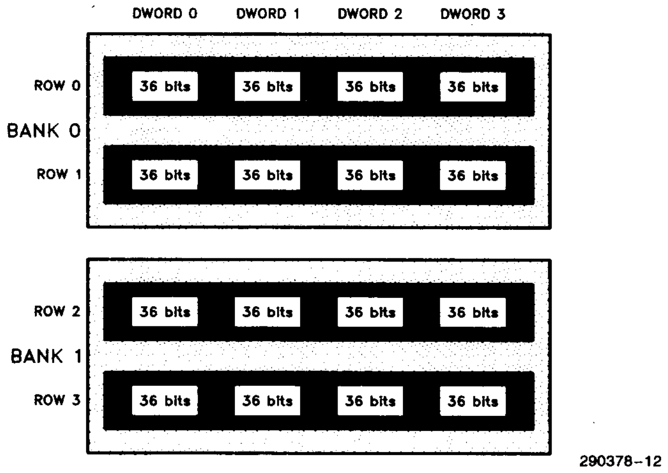


Figure 6-1. Memory Banks

Although four rows by four columns of dwords are supported, the designer may implement a subset of the array as he chooses.

The 82359 does not place any restrictions on the implementation of the 36-bit memory element. Each may consist of x1, x4, x9, or x36-bit DRAM devices as long as the address depth of each sub-element is identical (i.e., all must be 256k for example). These memory elements may be implemented with DIP, SIP, or single or double density SIMM structures.

The 82359 provides 11 address bits to the memory array allowing DRAM of up to 4 meg in address depth to be addressed. It also supports the smaller size DRAM of 64k, 256k or 1 meg. Each of the rows in the array is DRAM address depth independent in that the depth of DRAM in one row may differ from the depth of DRAM in another. For example, row zero may contain DRAMs of 1 meg, while row one may contain DRAMs of 256k. The 82359 knows the depth of the DRAM in each row through the value programmed into each row's Row Configuration Register. The ability to mix DRAMs of various address depths in the memory array provides tremendous flexibility in that the end-user is allowed to upgrade the amount of system memory in increments which best suit his or her needs without disposing of the old DRAMs which have already been installed. The row(s) containing the largest amount of memory must be physically located in the lowest numbered row(s).

The 82359 also has the capability of supporting DRAMs of various speeds (or access times). Specifically, DRAMs of 60, 70, or 80 ns can be supported. The 82359 requires that the entire array be populated with DRAMs of the same speed. The 82359 generates the various timings each speed requires through its internal delay lines. Since these delay lines are programmable through the Programmable Timing Registers, critical DRAM timings can be generated for the various speeds. (See the Internal Delay Line section for more information.) The ability to support various speed DRAMs allows the system design to be independent of what speed DRAMs populate the array. Systems requiring the highest possible performance may populate the array with a DRAM speed which gives 0 wait state memory performance. Other systems targeted for the medium performance range may use slower DRAMs at a lower system cost.

Population of the array also provides memory flexibility. Since positions for up to four dwords are provided for each row, a maximum memory line size of 128 bits is allowed. This does not always have to be the case, as memory line sizes of 64 or 32 bits are also supported. As with DRAM size, population of each row in the array is row independent and programmable through each of the row's Row Configuration Registers.

The population of the row will be referred to as the row's "Dword Interleave Factor" throughout this document. A row populated with one dword is referred to as "one-way dword interleaved"; a row two dwords is referred to as "two-way dword interleaved"; and a row with all four dwords populated is called "four-way dword interleaved".

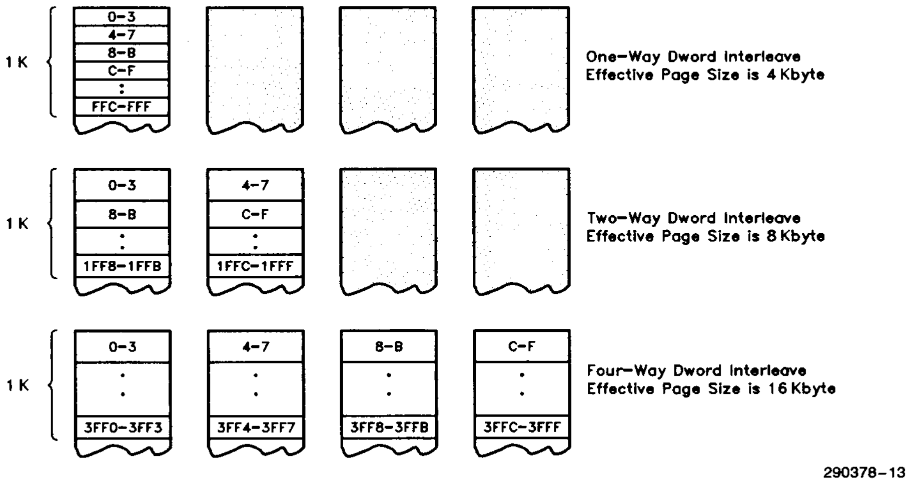
The following is a table of all valid row populations and the dword interleave factor for each.

Valid Row Population	
Row Population (Dword)	**Dword Interleave Factor**
0	1-way
1	1-way
2	1-way
3	1-way
0,1	2-way
2,3	2-way
0,1,2,3	4-way
empty	xxxxx

There are several simple rules which must be followed when populating the array.

1. All memory components in a given row must be of the same size (i.e., all 256k DRAMs).
2. The row(s) containing the largest amount of memory must be physically located in the lowest numbered row(s).
3. All DRAMs of the entire array must be of the same speed (or the 82359 must be programmed to operate at the speed of the slowest DRAM installed).
4. When using 64k DRAMs in a given row, four-way dword interleave must be used.
5. The population of any row may take on only those row configurations described above.

The 82359 maps sequential memory addresses across the row, interleaving the dwords into the populated columns. The entire row is connected to the same RAS# and thus all populated dwords of the row are accessed with each memory cycle to that row. Each time that column's RAS# is asserted, the new row address is strobed into each dword of the row. Since addresses are "interleaved" across the populated columns of the array, the population plays a role in the effective DRAM page size of that row. If the row is populated as one-way dword interleaved, the effective DRAM page size is that of the actual DRAM page size. Should the row be two- or four-way dword interleaved, the effective DRAM page size is increased by two or four times respectively. Thus by increasing the dword interleave factor, the chances for DRAM page hits increases due to a larger effective DRAM page size.



Example shows usage of 1 Meg DRAMs, 1K DRAM page size.

Figure 6-2. Memory Interleave

Any of the above populations are valid but one may be more efficient than another based on the speed of the DRAM and what type of CPU/Cache combination is acting as the host. For example, an 80386 processor with no cache does not support burst transfers, and therefore one-way dword interleaved memory for each row is considered effective since the all host cycles will be satisfied with one memory cycle.

Host devices such as the i486 processor or the 82395 cache utilize a four dword burst cycle for transfers and thus require multiple dwords per host cycle. In this case a one-way dword interleave population would require four memory cycles to satisfy the four dword burst and, should a fast host be installed and the DRAM memory is relatively slow, wait states may be incurred between dwords of the burst while each memory cycle is completing. Progressing to a two-way dword interleaved population would allow each memory cycle to fetch two dwords at a time and thus the second and fourth dwords of the burst are available at zero wait state. Once the first DRAM cycle has completed and the two dwords latched into the 82353 Data Path device, the second DRAM cycle can begin. Should relatively fast DRAMs be used and this second cycle completes before the host expects the third dword of the burst, no wait states will occur. If slower DRAMs are used, wait states may be needed until the memory cycle completes and the third and fourth dwords of the burst become latched into the 82353. If four-way dword interleaved memory is accessed by the memory cycle, all four dwords of the burst will be read from the array at the same time and sent to the host with zero wait states in between each dword transfer of the burst regardless of the speed of DRAM. Typically a 25 MHz or 33 MHz host CPU requires at least two-way dword interleaved memory to satisfy a zero wait state burst.

6.2 Memory Interface Signals

The 82359 provides all the signals needed for DRAM address and control. Four RAS# signals are provided on a one-per-row basis to support the four rows of the memory array. Eight CAS# signals are provided. These are divided into two groups of four as described below. Four WE# signals control the reading and writing of memory and, along with the CAS# lines, provide the ability to write to memory on a byte-wide basis. The 82359 provides 11 bits of DRAM address. Nine global address bits are labeled MADDR(8:0). Each row is also provided with two row-specific bits from the eight RMADDR(7:0) signals.

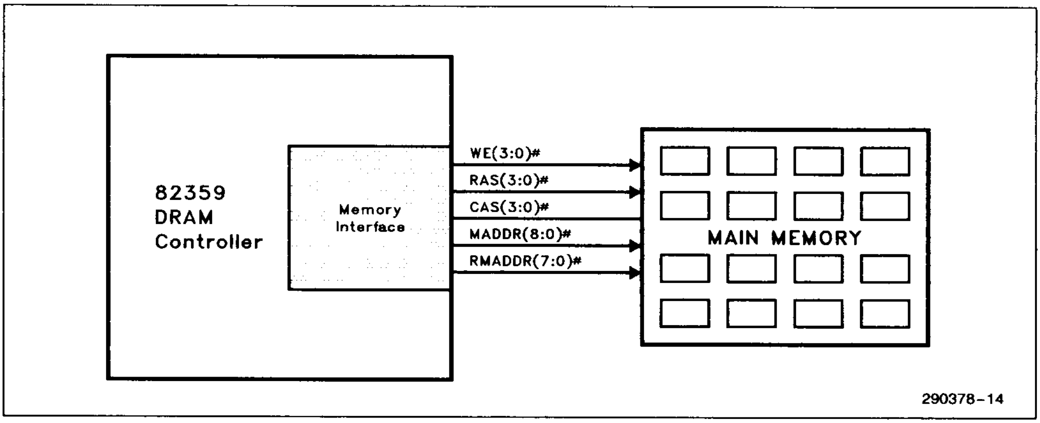


Figure 6-3. Memory Interface

6.2.1 WE# SIGNALS

Four WE# (Write Enable) signals are provided to the memory array. Each write enable signal should be connected to each corresponding dword of the memory array (i.e. WE0# to all dword0 DRAMs, WE1# to all dword1 DRAMs). Dwords in the same column share the same WE# signals.

When the 82359 performs a write to the memory array, it will assert the appropriate WE#, determined from the address of the write and memory array population. The WE# signals are also used by the 82353 Advanced Data Path for data output enables to the DRAM array.

6.2.2 RAS# SIGNALS

The 82359 generates four RAS# signals, RAS(3:0)#. Each RAS# connects to all dwords that populate its corresponding row.

Critical RAS# timing generation for the particular speed of DRAMs in the memory array is accomplished through the Internal Delay Line and Programmable Timing Registers. (See the sections entitled Internal Delay Line and Programmable Registers for more information on the generation of the RAS# timings.)

The 82359 supports four modes of RAS# operation. These modes specify which RAS# lines are left active for host or system cycles. These modes are described under the section "RAS# Modes of Operation".

6.2.3 CAS# SIGNALS

Eight CAS# signals, CAS(7:0)#, are available to the memory array. These eight CAS# lines can be broken down into two groups of four, CAS(7:4)# and CAS(3:0)#. CAS(3:0)# are shared by rows 0 and 1 of the memory array, and CAS(7:4)# are shared by rows 2 and 3. Each of the four individual CAS# lines in the group of four connects to each of the corresponding bytes in the memory array row. For example, CAS0# would connect to byte 0 of element 1, byte 0 of element 2, byte 0 of element 3, and byte 0 of element 4 in both rows 0 and row 1.

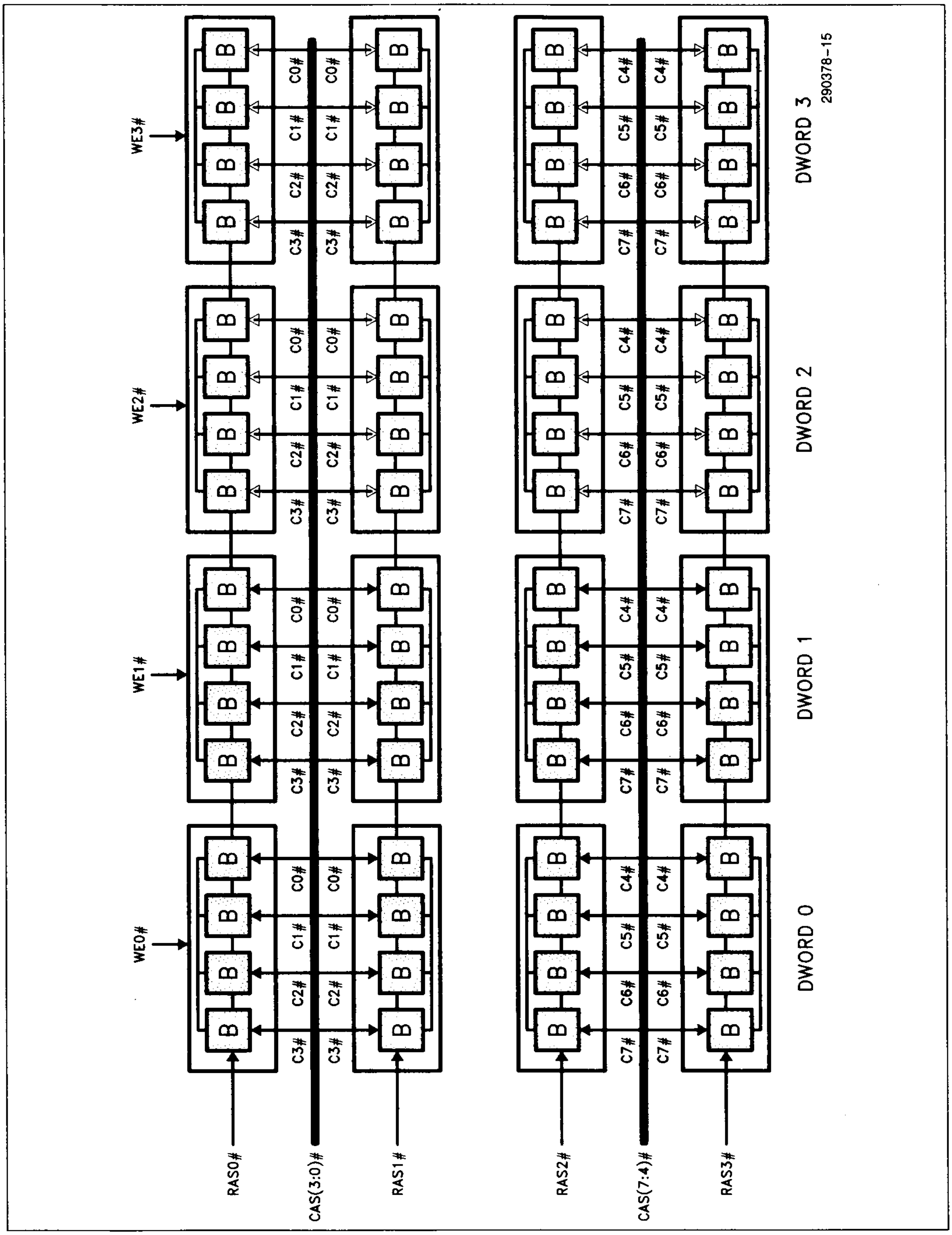


Figure 6-4. RAS# and CAS# Connection to the Memory Array

During write cycles, only the CAS# signals to the bytes being written are asserted. During read cycles, the 82359 attempts to read the entire row regardless of the BE(3:0)# value and the 82353 "throws away" what it doesn't need. Therefore, during DRAM reads, all CAS# signals of the row being read are asserted.

From the WE#, RAS#, and CAS# connection scheme, it can be seen that when reading the DRAM array, the 82359 will assert the RAS# for the row which the data resides in, asserts all CAS# lines of that row, and de-asserts all WE#'s. During write cycles, the 82359 will assert the appropriate RAS# signal for the row which the address resides in, the appropriate CAS# signals for the byte, word or dword to be written, and the appropriate WE# signal for the column in which the data is to be written.

Also notice that with the CAS# sharing between row 0 and 1, and row 2 and 3, the 82359 will never assert RAS0# and RAS1# at the same time, nor will it assert RAS2# and RAS#3 at the same time. Doing so would create data contention between DRAM of the shared rows.

6.3 DRAM Address Generation

The mux'ed row/column address to the memory array is provided by two groups of address signals. The MADDR(8:0) provide nine bits of address which connect to all memory elements in the array. There is also a group of eight row-specific address lines (RMADDR(7:0)) which are broken down into 4 groups of 2. Each group of 2 connect to a specific row in the memory array. These 2 bits provide the two low order address bits to the row. These bits are toggled for the second, third, and fourth accesses in the burst sequence or 1- and 2-way dword interleaved memory populations. The 9 MADDR(8:0) signals, along with the 2 row-specific RMADDR bits, provide the 11 bits needed to access a DRAM with up to 4M address depth.

Since single or double density SIMMs are supported, a different connection scheme is required for each type. The following diagram shows how to connect the row specific bits to the array using single density SIMMs:

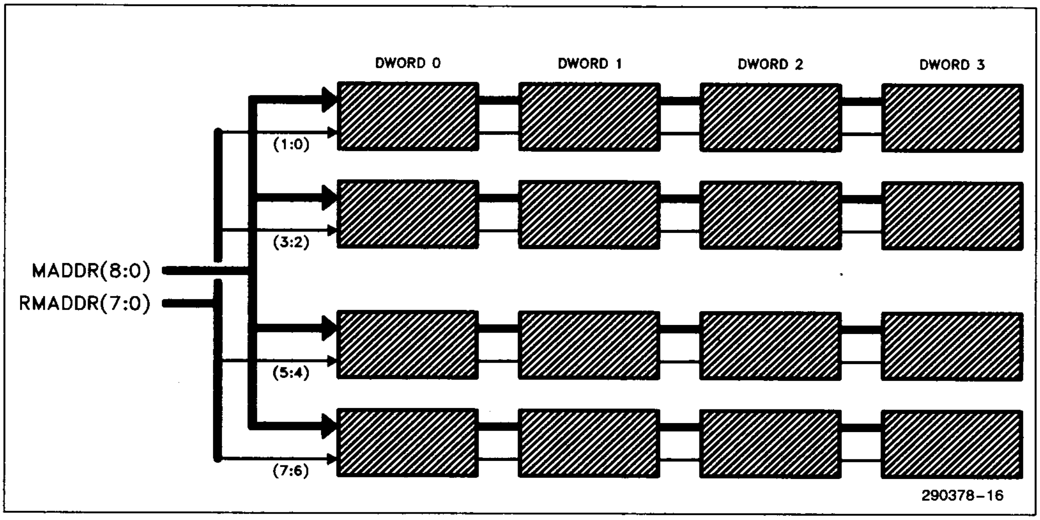


Figure 6-5. Row Specific Bit Connections—Single Density SIMMs

When using double density SIMMs, the row specific bits can't drive on a row basis since the module straddles a row pair. In this case, to maintain equivalent loading, the RMADDR(7:0) bits connect to the array as follows:

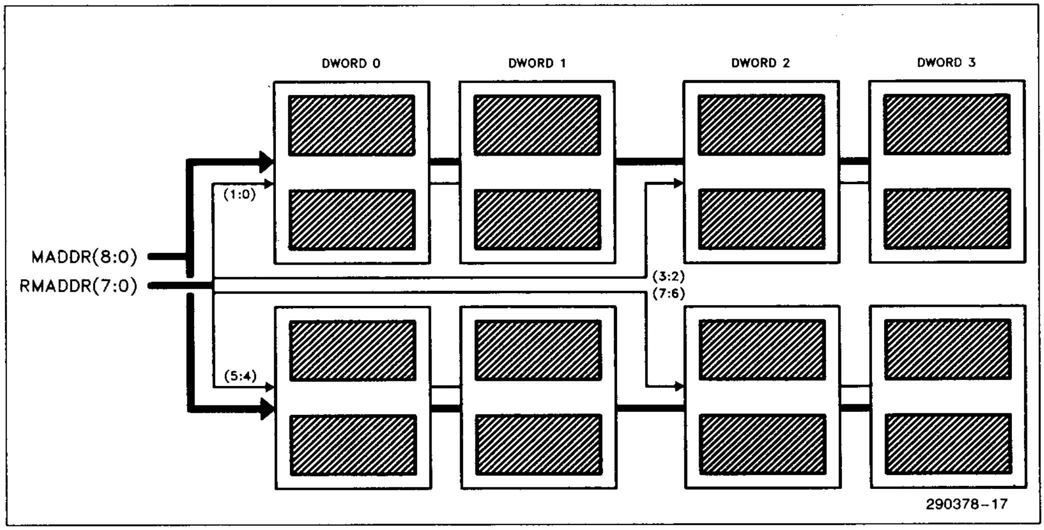


Figure 6-6. Row Specific Bit Connections—Double Density SIMMs

Connecting RMADDR(7:0) this way mandates that the row 0 and row 1 specific bits carry identical information. The same is true of the row 2 and row 3 specific bits. This implies that the population bits for a row-pair linked by a double density SIMM convey identical DRAM size and dword interleave factor information. This is not a problem however in that using a double density SIMM implies exactly this. It does however preclude a configuration where, for example, a row is populated by double density SIMMs in dwords 0 and 1, and dwords 2 and 3 are populated with single density SIMMs. This configuration is excluded since row 0 is 4-way dword interleaved while row 1 is 2-way, which will cause the information content on the row specific address bits to differ.

To accommodate sockets that can accept either single or double density SIMMs, the RMADDR(7:0) double density connection above should be used. Then, when populated with single density SIMMs, the population encoding should indicate that all the memory resides on only 1 of the rows, and that the other linked row is empty. When a row is programmed as empty, the 82359 automatically duplicates the row specific information of the linked populated row onto the row specific bits of the empty row. This maintains identical information content on the row specific bits of the linked row-pair.

During refresh cycles the least significant address bits of the refresh address are copied on all row specific address outputs: RMADDR(1:0) = RMADDR(3:2) = RMADDR(5:4) = RMADDR(7:6).

The 11-bit DRAM address is generated off of the 32-bit host or system address port. The exact translation of the 32-bit address to the 11-bit DRAM row and column address is dependent on the size of the DRAM being addressed, the dword interleave factor and the active RAS mode.

In the following table, the 32-bit host or system address to DRAM address translation is given for each of the factors described above. Recall that the 11-bit DRAM address comprises of the 9-bit MADDR(8:0) address bus and 2 bits of the row-specific address bus RMADDR(7:0). Bits 10:2 refer to the MADDR(8:0) bus and bits 1:0 refer to the row specific bits.

For example, assume the host is accessing a memory row which 2-way dword interleaved, contains 256K DRAMs. The row and column addresses seen by the DRAM is shown below:

DRAM Address Bus

	10	9	8	7	6	5	4	3	2	1	0
Row Address	X	X	HA19	HA17	HA16	HA15	HA14	HA13	HA18	HA12	HA20
Col Address	X	X	HA10	HA9	HA8	HA7	HA6	HA5	HA4	HA3	HA11

DRAM Address Decoding

										W=1		W=2		W=4	
RAS# ADD	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**	**01**	**00**	**01**	**00**
64K	X	X	X	17	16	15	14	13	12	Invalid		Invalid		19	18
256K	X	X	19	17	16	15	14	13	18	12	11	12	20	21	20
1M	X	21	19	17	16	15	14	20	18	12	13	22	13	23	22
4M	23	21	19	17	16	15	22	20	18	14	13	14	24	25	24

CAS# ADD	**10**	**09**	**08**	**07**	**06**	**05**	**04**	**03**	**02**	**01**	**00**	**01**	**00**	**01**	**00**
64K	X	X	X	09	08	07	06	05	04	Invalid		Invalid		10	11
256K	X	X	10	09	08	07	06	05	04	03	02	03	11	12	11
1M	X	11	10	09	08	07	06	05	04	03	02	12	03	12	13
4M	12	11	10	09	08	07	06	05	04	03	02	03	13	14	13

W = Dword Interleave Factor.

The user must arrange the memory array rows according to the total amount of memory. The user must place the DRAMs with the largest total memory in row 0. The user must also arrange the DRAMs in the remaining 3 rows according to their total amount of memory, placing the largest amount in the lower numbered row. For example, assume that row 0 contains a single 256K by 36-bit SIMMS and no other rows are populated. The user wants to add two 1 meg by 36-bit SIMMS to their system. The user would physically place the two 1 meg by 36-bit SIMMS in row 0, and move the 256K by 36-bit SIMMS to row 2.

Note that adding memory to the system may change the memory performance. For example, if a system is shipped with 1M total memory installed in row 0 (64K x 36 x 4 dword interleave) and all other rows are unpopulated, all memory cycles to the first meg of memory will be to a four-way dword interleaved row. Only one memory cycle is necessary to complete a burst line fill operation. If 4M of memory (1M x 36 x 1 dword interleaved) are to be added at a later date, the user must first physically move the 64K x 36 x 4 dword interleaved DRAMs from row 0 into another row, and then place the 1M x 36 x 1 dword interleaved DRAM into row 0. This will cause memory addresses 0 to 4M to map to row 0, and address 4M to 5M to map to the other row. Now all accesses to the first meg of memory will be to a one-way dword interleaved row. Now four memory cycles will be required to complete a burst line fill, therefore wait states may be necessary.

6.4 RAS# Modes of Operation

The 82359 provides four modes of functionality in which RAS# can operate, programmable through the RAS# Mode Register. These four modes are labeled "0 Active RAS# Mode", "1 Active RAS# Mode", "2 Active RAS# Mode", and "Dynamic RAS# Mode". These modes control which RAS# signals, if any, may be left active between memory cycles. Each of the modes provides different memory performance in terms of access times for each of cases of DRAM page hits, DRAM page misses, row hits, row misses, and combinations of these.

Define:

Page Hit: *A memory cycle to the same DRAM page as the preceding memory cycle. Obviously, in this case the RAS# of the row being accessed must have been left in the asserted state from the previous DRAM access and because the current access is to the same DRAM page, no row strobing is required and access time is CAS# limited. This type of memory cycle results in the fastest access time since a new DRAM row does not have to be strobed.*

Page Miss: *A DRAM Page Miss is the least desirable type of memory cycle. This type of cycle results from the current memory cycle occurring to a row who's RAS# has been left active from the previous memory cycle, but the current access is to a different DRAM page. In this case RAS# must first be de-asserted and row precharge time satisfied before a new DRAM row can be strobed. This type of cycle results in the longest access time.*

Row Miss: *A Row Miss is a memory cycle which causes a page miss due to the current cycle accessing a row who's RAS# was not left in the asserted state. In this case, part or all of the required row precharge has been satisfied during the time RAS# was high, and thus RAS# may be asserted immediately if row precharge has been met. These types of cycles are slower than DRAM page hits due to the need for the strobing of the new row, but faster that DRAM page miss cycles which must pay the entire row precharge penalty.*

Note that no matter what mode of RAS# operation is currently in use, all RAS# lines are de-asserted at the end of of ISA and EISA standard cycles (when CMD#, MRDC#, or MWTC# rises). All RAS# lines are also de-activated whenever a change in memory ownership occurs.

6.4.1 0-ACTIVE RAS# MODE

When 0-Active RAS# mode is selected, the 82359 will de-assert all RAS# signals after every host or system cycle to memory. All RAS# signals are de-asserted from a delay line tap that meets the minimum RAS# hold from CAS# specification. This mode guarantees that the following host or system cycle will always be a DRAM page miss and thus require a row to be latched into the DRAM regardless if the access is to the same DRAM page as the previous cycle.

This mode is beneficial for systems where the page hit rate is expected to be low. Many of the memory cycles which would have resulted in the worst case page miss cycles now result in the shorter row miss memory cycles. Since all RAS# signals are de-asserted after every host or system cycle, at least three of the four RAS# lines have already satisfied their RAS# precharge before the next cycle is started (the fourth RAS# line may or may not have satisfied its RAS# precharge depending on how much time separates sequential memory cycles). Should the subsequent cycle be to a different row than the previous cycle, the RAS# precharge has been satisfied so that the 82359 may asserted RAS# immediately after the row address setup time has been met.

The major disadvantage of 0-Active RAS# Mode is that if the system is expected to have a reasonable page hit rate (>50%), the access time of the row miss cycle will be longer than the average access time of page hit and page miss combination.

Also, it should be noted that the 82359 does not guarantee RAS# precharge in this mode. The PST must take responsibility for RAS# precharge by delaying back-to-back host cycles to the 82359 until the precharge time has expired if necessary. (Note: at 25 MHz and 33 MHz, precharge time does not have to be actively monitored since the fastest possible time from RAS# deactivation in one cycle to RAS# activation in a subsequent cycle is greater than the required precharge time.)

6.4.2 1-ACTIVE RAS# MODE

The second mode of RAS# operation is labeled "1 Active RAS# Mode". In this mode, the 82359 will leave the most recently used RAS# signal in the asserted state for host or system PST master cycles. Unlike the 0-Active RAS Mode, the 82359 controls RAS# precharge and the PST does not need to hold back host-to-memory cycles for this reason.

Since the last used RAS# remains active, a subsequent cycle to this row (row hit) may result in one of two DRAM cycles. Should the address of the subsequent cycle be to the same DRAM page as the previous cycle, a DRAM page hit will result. In this case the 82359 does not latch a row address into the DRAMs, but immediately latches the column. This results in a substantially faster DRAM access since neither the relatively long RAS# precharge or row address strobing is required.

The second case would be that the subsequent cycle is to the same row (row hit) but that the address is to a different DRAM page (page miss). In this case the 82359 must de-assert the active RAS# and hold it until RAS# precharge is satisfied. Then a new row address must be latched into the DRAMs. This results in a relatively long DRAM access.

The third case possible in this mode is for the subsequent cycle to be to a different row than the previous cycle. This results in a new row being latched into DRAM immediately since RAS# precharge is hidden (satisfied during the previous cycle) resulting in a faster access time than a page miss cycle.

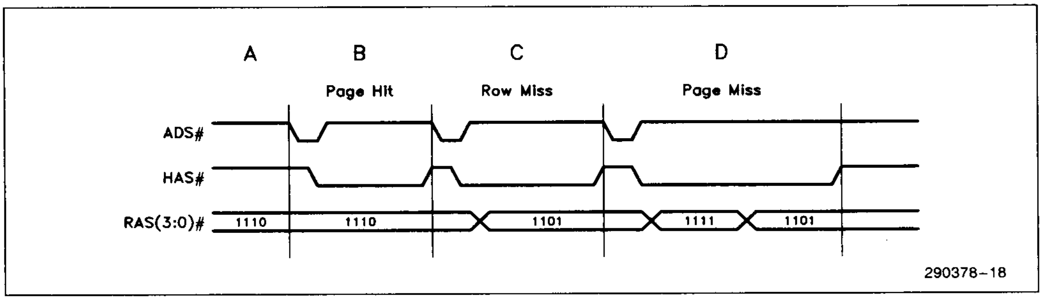


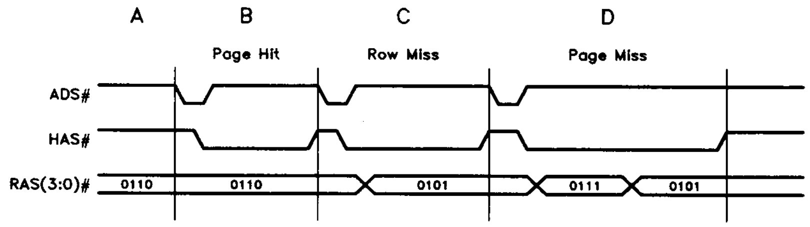
Figure 6-7. I-Active RAS# Mode

Transition	Notes
A-B	Page hit to the most recently used RAS#. RAS# was left in the asserted state at the end of cycle A allowing for a page hit cycle. The currently used RAS# remains active into the next memory cycle.
B-C	The row miss causes the previously active RAS# to be de-asserted and the current RAS# to be asserted. Its RAS# precharge has been met during the previous cycle.
C-D	The page miss cycle causes the active RAS# to become de-asserted and then asserted, latching the new row address only after RAS# precharge has been met. This results in a relatively long DRAM cycle.

6.4.3 2-ACTIVE RAS# MODE

This mode is similar to 1-Active RAS# Mode but it allows two RAS# signals to remain active at the end of a host or system cycle. One restriction is applied as to which RAS# signals may remain active. Since row 0 and 1 share the same CAS# lines, allowing RAS0# and RAS1# to be active at the same time would cause both rows of the array to drive their data lines and thus data contention would occur. A similar scenario exists for rows 2 and 3. Due to this restriction, the 82359 allows only the most recently used RAS# from rows 0 and 1, and the most recently used RAS# from rows 2 and 3 to remain active at the end of the current host or system cycle.

This mode provides a benefit over 1-Active RAS# Mode in that allowing for two RAS# signals to remain active results a better chance for a DRAM page hit.



290378-19

Figure 6-8. 2-Active RAS# Mode

Transition	Notes
All	All cycles are very similar to those of 1-Active RAS# Mode. In the diagram, it can be seen that two of the most recently used RAS# lines are left asserted (one from the row 0, row 1 pair; one from the row 2, row 3 pair).

6.4.4 DYNAMIC RAS# MODE

The final mode of RAS# operation is called "Dynamic RAS# Mode". In this mode the 82359 determines which RAS# mode of operation is best suited for the current type of memory cycles. In this mode the 82359 operates in 2-Active RAS# Mode. When three back-to-back page miss memory cycles are detected, it dynamically switches to 0-Active RAS# Mode in which precharge can be hidden. Likewise, when three back-to-back page hit cycles occur in 0-Active RAS# mode, the 82359 will dynamically switch back to 2-Active RAS# Mode. By doing so, the 82359 operates in the modes which provide the optimum memory access times for the type of memory cycles which are occurring at the present time.

6.5 Special Considerations

There is an important distinction between the "non-page mode" of 0-Active RAS# Mode and the other two "page mode" combinations of 1- and 2-Active RAS# Mode. If the 82359 is programmed for 1- or 2-Active RAS# Mode, it assumes responsibility for guaranteeing that row precharge between cycles is met. If for any reason a row's RAS# is de-asserted at the end of a cycle, a subsequent cycle to the same row will only proceed as a row miss (RAS# asserted immediately).

If the 82359 is programmed for 0-Active RAS# Mode, all host and system PST cycles are run with row miss timings (no row precharge), and it becomes the host or system PST's responsibility to guarantee row precharge between back-to-back accesses by delaying HAS#/SAS# generation if required. This allows for optimal non-page mode operation. (For example, a burst access to a 4-way dword interleaved row causes RAS# to be de-asserted after a single MDS# pulse which fetches 128 bits of data. If the typical 16-byte line size is used, the data for the entire host cycle has been read, therefore RAS# is deasserted and the row precharge for the next cycle is overlapped with the burst read.

When the 82359 is programmed for 0-Active RAS# mode, 1 BCLK EISA burst cycles to main memory do not function correctly because the RAS# signals fail to remain active throughout the entire burst sequence. In 0-Active RAS# mode, 2 BCLK EISA burst cycles do function correctly, and when programmed in other RAS# modes (i.e., 1-Active RAS#, 2-Active RAS# or Dynamic RAS# mode) 1 BCLK EISA burst cycles function correctly.

Also note that designing a system for 0-Active RAS# Mode operation does not imply that non-page mode DRAM can be used. The differences between modes which leave RAS# active and those which do not apply only to the lead off access of a burst. Multiple CAS# pulses for subsequent dwords of the burst still utilize DRAM page mode operation in that RAS# will remain asserted throughout the entire host or system cycle. This is also true for EISA bursts where CAS#-only cycles are run.

One final note is that regardless of which RAS# mode is currently selected, whenever a change in memory ownership occurs (i.e., refresh, system master, RAS# timeout), all RAS# signals are de-asserted resulting in the next cycle to main memory being a row miss.

6.6 DRAM Refresh Generation

The 82359 allows for two modes of main memory refresh; (1) Coupled Mode, and (2) Decoupled Mode. Coupled Mode refresh causes the 82359 to run main memory DRAM refresh cycles based at the same time that refresh cycles are occurring on the EISA bus. In Decoupled Mode, the 82359 refreshes the main memory DRAM array independent of the occurrence of EISA refresh cycles. The refresh mode is selected via Mode Register A, bit 6 and 7.

6.7 Decoupled Refresh Mode

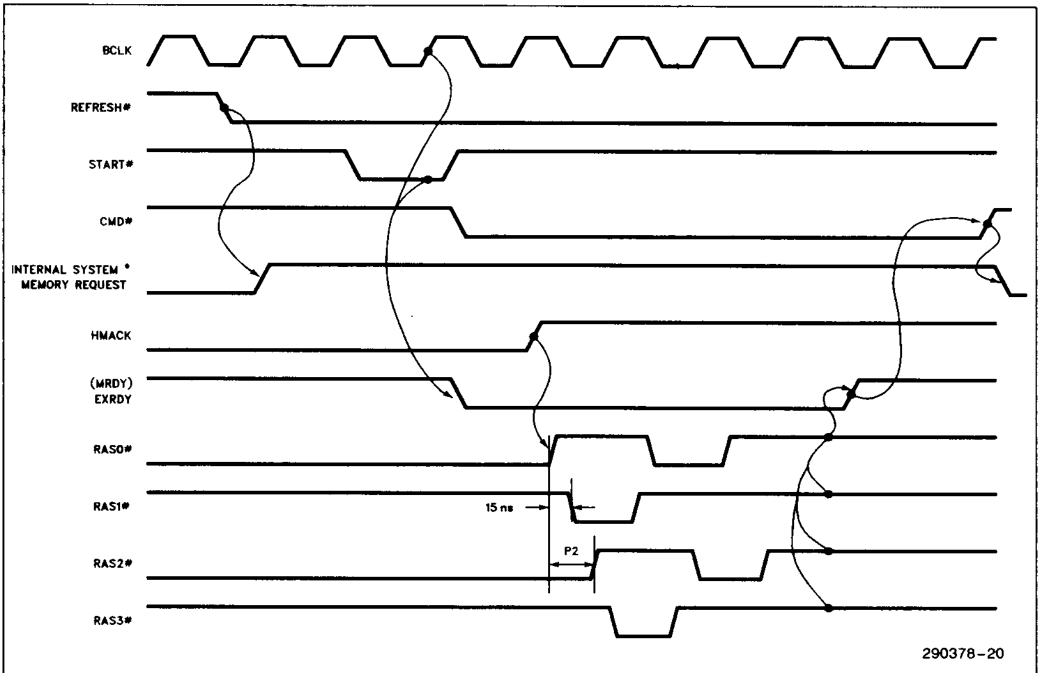
Decoupled Refresh Mode must only be selected in Concurrent mode. If the Decoupled Refresh Mode is selected, the 82359 is responsible for generating refresh cycles to main memory. In this mode, all EISA refresh cycles are ignored and REFRESH# asserted causes the system refresh cycle to be ignored. When main memory refresh is needed (as signified by the 82359's internal refresh timer), the 82359 actively causes memory to be arbitrated away from the current owner. Should the current owner be the host, the 82359 will cause a memory ownership request (HMREQ) to occur. Upon receiving the HMACK from the PST, the refresh will be executed. Should an EISA device currently own main memory at the time of the refresh request, the 82359 will wait for the current EISA cycle to finish and during the subsequent cycle assert MRDY (typically connected to EXRDY of the EISA bus) causing the EISA device to be held off in wait states while main memory refresh occurs.

Three varieties of decoupled refresh are available. Single Cycle Refresh causes the 82359's internal refresh timer to cause a refresh request once every 15 μs and causes a single refresh cycle to be executed. Two Cycle Burst Refresh causes a refresh request once every 30 μs and two refresh cycles are executed in a back-to-back fashion. Four Cycle Refresh causes refresh requests once every 60 μs and four back-to-back refresh cycles are generated. The 82359 determines which of the three varieties of decoupled refresh is desired by the setting of Mode Register A.

The default power-up state of the 82359 is Non-Concurrent mode/Coupled Refresh mode (NC/CR). To change to Concurrent mode/De-Coupled Refresh mode (C/DR) the 82359 should first be programmed into NC/DR mode, and then into C/DR mode. Changing from NC/CR mode to C/DR mode requires a minimum of four 8-bit I/O cycles. A SHOLD request during these I/O accesses may cause an arbitration problem in the 82359 and prevent the system from booting. To prevent this arbitration failure, it is recommended that refresh be temporarily disabled while programming the registers which place the 82359 into C/DR mode.

6.8 Coupled Refresh Mode

Coupled Refresh Mode should only be selected in Non-concurrent mode. In the Coupled Refresh Mode, a refresh cycle to main memory is initiated at the same time that refresh is executed on the EISA bus. In this mode the 82359 monitors the EISA bus REFRESH# signal. When REFRESH# goes active, the 82359 causes a memory ownership request of the host PST by asserting HMREQ. The refresh cycle to main memory does not begin until memory ownership has been acquired. In the meantime the EISA refresh cycle is held off with MRDY, asserted off START# sampled active. Even though EISA refresh provides the refresh address (through the ISP), the 82359 uses its refresh address generator for main memory refresh addresses. To allow enough time for all four rows of main memory to be refreshed, MRDY is negated from the sampling of BCLK active during the refresh cycle. MRDY will return to the asserted state when all RAS# signals have returned to their de-asserted state.



*As gated to HMREQ by the host memory throttles.

Figure 6-9. Coupled Refresh Cycle

Due to the large current surge associated with the rising and falling edges of RAS#, the 82359 causes these edges to be staggered by P2 ns (which will vary based on the row precharge time value programmed into the 82359). The staggered RAS# refresh sequence is run regardless of memory array population.

The 82359 follows a RAS# precharge algorithm that allows only one RAS# signal to be activated every cycle (based on the value of the P2 register). RAS# deactivation is based on the same algorithm but always leads RAS# activation for the same cycle by approximately 15 ns. The initial staggering sequence is based on whether RAS0# or RAS1# is active and RAS2# or RAS3# is active at the start of the refresh cycle. This is shown in the table below:

Start	Subsequent Cycles					
RAS (3:0)#	1	2	3	4	5	6
1010	1001	0101	0110	1010	1011	1111
10X1	1010	0110	0101	1001	1011	1111
X110	*101	1001	1010	0110	0111	1111
X1X1	*110	1010	1001	0101	0111	1111

*Indicates no change in state.

Regardless of the population of the memory array, all RAS# signals will be toggled as if the row is populated. Also regardless of which mode of refresh is used, all RAS# lines will be negated at the end of a refresh cycle. This causes the next cycle to main memory to be a row miss cycle. Since the memory ownership must be transferred from the 82359's possession to the next owner after the refresh, and the time required for this transfer is greater than the row precharge time, precharge is always hidden after a refresh cycle.

7.0 82353 ADVANCED DATA PATH INTERFACE

Although the primary function of the 82359 is to manage the operation of DRAMs, it must also supply control information to the slave 82353 Data Path devices. Since the 82359 accepts cycle requests, and from the cycle address, determines the cycle's destination, it provides the proper control information to the 82353 for the appropriate routing of the cycle's data.

The 82359 DRAM Controller provides control information to the 82353 Data Path through five signals. HIOE# and MIOE# provide data routing information to the Data Path. From these two bits, the 82353 selects one of the three Data Path interfaces (host, system, or memory) as the source of data. H/S# is used to control the Data Path's internal state machines and also indicates which set of memory read latches and memory write latches are used. The SEL(1:0) signals tell the 82353 which one of four dwords of the 128-bit memory array is the valid dword for reading or writing. The 82359 provides the rising edge of MDS# to close the 82353's memory read latches at the`appropriate time.

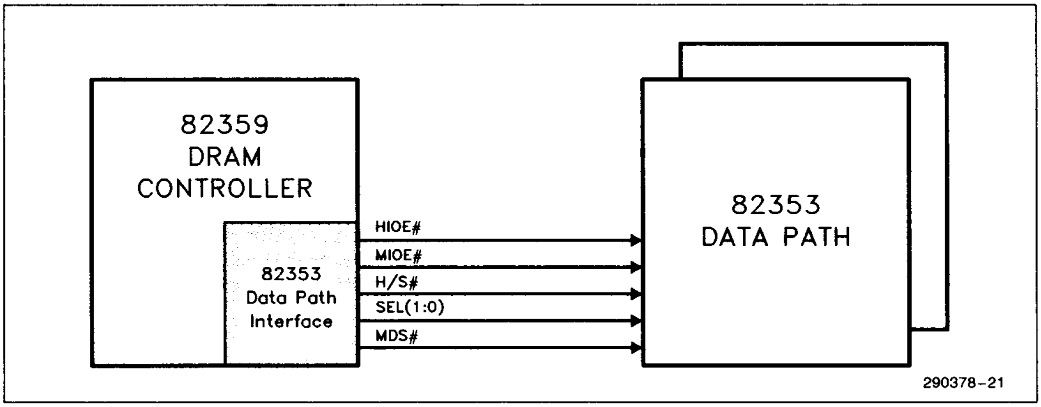


Figure 7-1. 82353 Advanced Data Path Interface

The three cycles which the 82359 must provide control information for are the following; (1) Host to main memory; (2) Host to system, and; (3) System to main memory. Cycles from system master to system slave do not involve an active participation of the 82359 and thus no data path support is needed. Also, snoop cycles sent to the host do not need data path support, so no 82353 control information is provided.

Data path routing is provided by two 82359 signals. HIOE# (Host Internal Output Enable) and MIOE# (Memory Internal Output Enable) are used to control the internal data bus of the 82353. The combination of these three signals determines which of the host, memory or system data interfaces is providing data for the current cycle. HIOE# takes priority over MIOE#, allowing the host to drive its data whenever it is asserted. When HIOE# is de-asserted, MIOE# selects between either the memory interface or system interface. Decoding of HIOE#, MIOE# is provided in the following table:

HIOE#	MIOE#	Data Source	Data Destination
0	X	Host	Memory or System (MIOE# active is meaningless since HIOE# dominates)
1	0	Memory	System or Host
1	1	System	Memory or Host

The 82359 notifies the 82353 that read data is valid with a rising edge of MDS#. This signal is a product of the 82359's internal delay line and the CAS#-to-MDS# Register setting. When the 82353 sees this edge, it latches data on its memory interface inputs into its memory read latches.

The 82359 and 82353 read the entire row of main memory during DRAM read cycles regardless of the number of dwords required by the cycle originator. Since multiple dwords are read at one time, the 82359 conveys which one of four possible dwords was requested from memory via SEL(1:0). During burst cycles where multiple dwords are to be sent, SEL(1:0) points to the lead-off dword, and from this, the 82353 mux'es the remaining dwords of the burst following the i486 predictable order burst sequence.

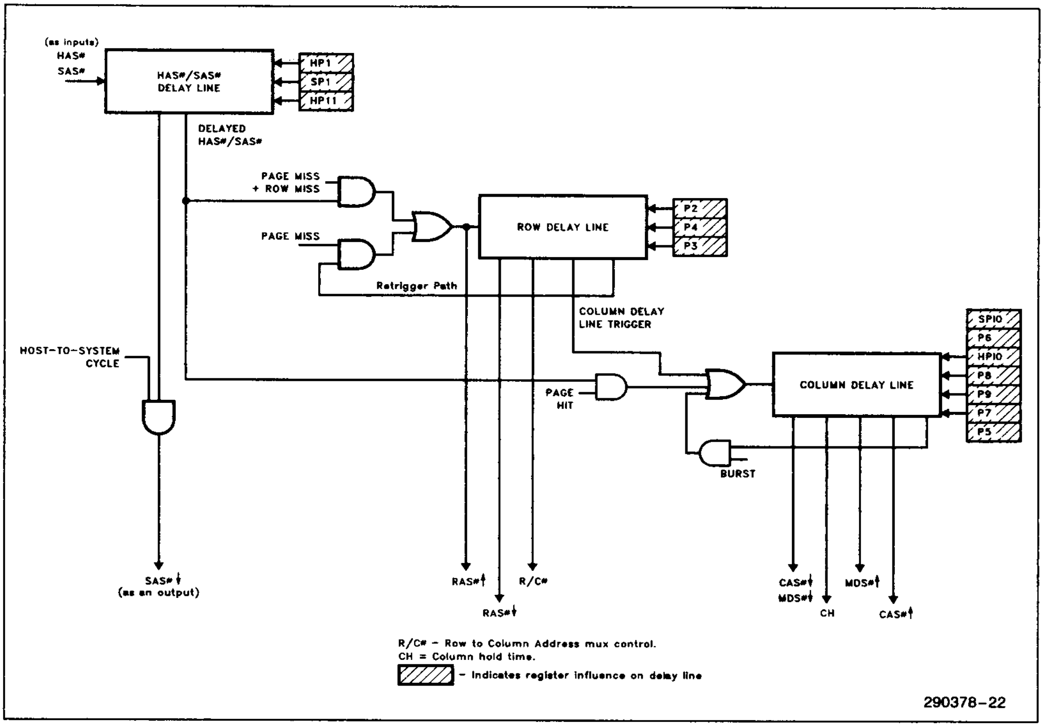
SEL(1:0)	IF(1:0)	Memory Array Dword Being Accessed
00	00 (4-Way)	0-1-2-3
00	01 (2-Way)	0-1-0-1
00	10 (1-Way)	0-0-0-0
00	11	System Access
01	00 (4-Way)	1-0-3-2
01	01 (2-Way)	1-0-1-0
01	10 (1-Way)	1-1-1-1
01	11	System Access
10	00 (2-Way)	2-3-0-1
10	01 (2-Way)	2-3-2-3
10	10 (1-Way)	2-2-2-2
10	11	System Access
11	00 (4-Way)	3-2-1-0
11	01 (2-Way)	3-2-3-2
11	10 (1-Way)	3-3-3-3
11	11	System Access

The 82359 signals which port has ownership of main memory via the H/S# signal. The 82353 uses this bit during memory reads for determining which set of read latches to use (host or system) and also uses it for state machine control as well as other functions.

8.0 INTERNAL DELAY LINES

Integrated into the 82359 are all required delay lines for generating critical DRAM timings. Through the use of these delay lines and the Programmable Timing Parameters, the 82359 is able to generate critical DRAM timings for various speed DRAMs.

This programmable delay line is derived from an Intel patented technology. It is compensated for temperature, V_{CC}, and process variations to allow programmability to 2.61 ns resolution with very high accuracy over a wide range of operating environments. The following is a functional block diagram of the Internal Delay Line structure.



R/C# – Row to Column Address mux control.
CH = Column hold time.
▨ – Indicates register influence on delay line

290378–22

Figure 8-1. Block Diagram of the Internal Delay Line

There are three distinct delay line elements, each under control of the Programmable Timing Registers. The left-most delay element is called the HAS#/SAS# Delay Line and its function is to generate a delay from the start of a host or system cycle to when the 82359 acts upon it. By causing a programmable delay, the 82359 has enough time to decode the address and determine the type of cycle it must execute. For CPU speeds of 25 MHz, no additional decode time is needed and this delay is typically programmed for its minimum 2.61 ns. For higher speed processors, delays greater than zero are required. This block is also responsible for generating the HAS# to SAS# delay for host-to-system cycles. This delay allows enough time for the host address and status to propagate through the 82359 and to satisfy setup requirements of the system bus controller.

The center delay element is the Row Delay Line. This block is responsible for generating all DRAM timings associated with strobing in the row address of the DRAM. Notice that this delay line is only triggered for row miss and page miss cycles. Also notice that this delay line has a retrigger path from one of its taps. This path qualified by a page miss cycle causes this delay line to recycle.

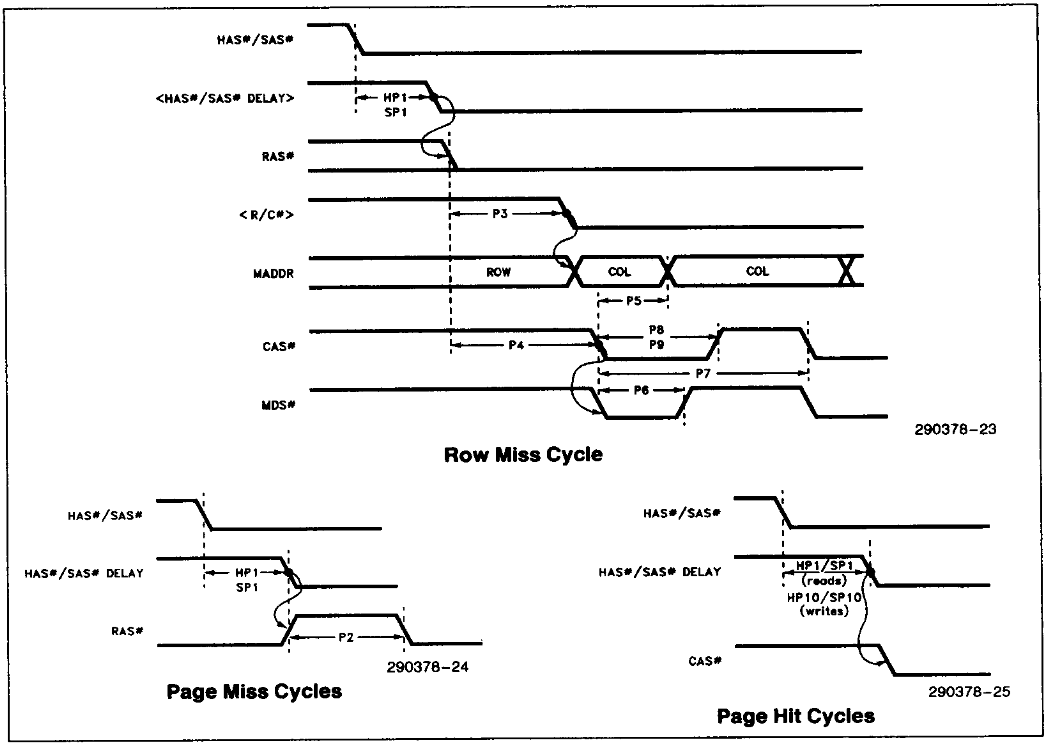
For a row miss cycle, the rising edge of RAS# has no effect since RAS# is already in the asserted state. Following this, RAS# is deasserted and then the Row/Col DRAM Address Mux is changed to the COL state. The delay line is not retriggered since the recycle path is gated for row misses.

For page miss cycles, RAS# is immediately de-asserted but RAS# falling, the Row/Col mux control signal R/C#, and the Column Delay Line Trigger are all gated out on the first pass through the delay line. This first pass is the row precharge time required in page miss cycles. The retrigger signal is qualified by the miss indicator and causes the Row Delay line to be recycled. In this second pass, RAS# falling, R/C# and the Column Delay Line trigger are allowed to reach their destinations.

This last element is labeled Column Delay Line. It is triggered off of one of three sources: (1) the Column Delay Line Trigger of the Row Delay Line for DRAM page misses or row misses; (2) directly off of the delayed cycle trigger of the HAS#/SAS# Delay Line for DRAM page hits, or, (3) from one of its own taps for subsequent cycles of a burst access. Upon being triggered, this delay element immediately asserts CAS# and MDS#. This block controls the rising edge of MDS# for data latching, the rising edge of CAS#, the column address hold time, and the CAS# cycle time for burst accesses.

Notice that all blocks of the delay structure are controlled by programmable registers. There are 13 Programmable Timing Registers internal to the 82359. Each register has a timing range associated with it which can be programmed to 2.61 ns resolution. A diagram of the functions of these registers and delay lines is shown below. Also a preview of the Programmable Timing Registers is given. For a full description of each register, see the programmable register section.

NOTE:
These descriptions assume that HMACK or SMACK is asserted indicating that the host or system master currently owns main memory. If this were not the case, HAS# or SAS# would simply indicate the request for a host or system cycle and the falling edge of HMACK or SMACK is the cycle start indicator. In this case, one would replace the HAS#/SAS# with HMACK#/SMACK# in the following diagrams and descriptions.



Row Miss Cycle

Page Miss Cycles

Page Hit Cycles

Figure 8-2. Programmable Timing Register Functions

Register	Description
HP1	**PROGRAMMABLE HOST DELAY:** Delay between the host cycle trigger (the falling edge of HAS# if HMACK is low, or the falling edge of HMACK if HAS# is low) and when the 82359 begins the memory cycle. This delay adjusts the amount of time for address and status setup, address decode, page hit/miss determination, and column address setup to CAS(7:0)#. Min: 2.61 ns Max: 18.27 ns Default: 18.27 ns
SP1	**PROGRAMMABLE SYSTEM DELAY:** Delay between the system cycle trigger (the falling edge of SAS# if SMACK is low, or the falling edge of SMACK if SAS# is low) and when the 82359 begins the memory cycle. This delay adjusts the amount of address and status setup, address decode, page hit/miss determination, and column address setup time to CAS(7:0)#. Min: 7.83 ns Max: 20.88 ns Default: 20.88 ns
P2	**PROGRAMMABLE ROW PRECHARGE:** P2 controls the amount of time RAS# is asserted for the purpose of Row precharge. Min: 44.37 ns Max: 80.91 ns Default: 80.91 ns
P3	**PROGRAMMABLE ROW HOLD:** This register controls the amount of time the row address is held valid from the time a RAS# signal is asserted. Min: 33.93 ns Max: 62.64 ns Default: 46.98 ns
P4	**PROGRAMMABLE RAS# TO CAS# DELAY:** P4 controls the amount of time between the assertion of a RAS# signal and the assertion of CAS# signal. The falling edge of CAS# causes the start of the Column Delay Line sequence. Min: 62.64 ns Max: 93.96 ns Default: 83.52 ns
P5	**PROGRAMMABLE COLUMN HOLD:** The time the column address is held from the assertion of one or more CAS# signals is determined from this register setting. Min: 10.44 ns Max: 33.93 ns Default: 28.71 ns
P6	**PROGRAMMABLE CAS# TO MDS# RISING:** This register controls the amount of time between the falling edge of CAS# and MDS# to the rising edge of MDS#. Data must be valid and setup to the 82353 for latching with the rising edge of MDS#. Min: 31.32 ns Max: 62.64 ns Default: 57.42 ns
P7	**PROGRAMMABLE CAS# RECYCLE TIME:** P7 controls the amount of time from the initial falling edge of one or more CAS# signals to the next falling edge of CAS# signals for burst cycles. This value causes the Column Delay Line to recycle, resulting in the next CAS#/MDS# cycle. This register also directly affects the amount of precharge time of CAS# for burst cycles. Min: 52.20 ns Max: 83.52 ns Default: 83.52 ns
P8	**PROGRAMMABLE CAS# LOW TIME FOR READS:** This register controls the amount of CAS# low time for read cycles. Min: 41.76 ns Max: 67.86 ns Default: 62.64 ns
P9	**PROGRAMMABLE CAS# LOW TIME FOR WRITES:** This register controls the amount of CAS(7:0)# low time for write cycles. Min: 20.88 ns Max: 46.98 ns Default: 31.32 ns
HP10	**PROGRAMMABLE HOST WRITE DATA SETUP:** The programming of this register controls the time from the internal host cycle trigger to the falling edge of CAS# for page hit write cycles. This allows enough data setup time to CAS# for host-to-memory page hit write cycles. Min: 31.32 ns Max: 52.20 ns Default: 52.20 ns
SP10	**PROGRAMMABLE SYSTEM WRITE DATA SETUP:** The programming of this register controls the time from the internal system cycle trigger to the falling edge of CAS# for system-to-memory page hit write cycles. This allows enough data setup time for system-to-memory page hit write cycles. Min: 7.83 ns Max: 20.88 ns Default: 20.88 ns
HP11	**PROGRAMMABLE HOST TO SYSTEM DELAY:** Host cycle trigger to SAS# active. This delay is used to adjust the amount of time required for host-to-system cycles to propagate through the 82359 and for system address and status setup to SAS#. Min: 41.76 ns Max: 67.86 ns Default: 67.86 ns

9.0 THROTTLE MECHANISMS

9.1 Introduction

The 82359 has three throttle elements to control memory ownership sharing. These are labeled Host-to-Memory Throttle (HMT), System-to-Memory Throttle (SMT), and Host-to-System Throttle (HST). Associated with two of the throttles (HMT, HST) are "watchdog" timers (HMTW, HSTW) which prevent the main memory resource from being wasted.

The throttles and watchdogs are devices which allow the designer to tailor the performance of the system to achieve the desired minimum host and system master performance. The throttles are features which allow a device which obtains ownership of a resource (main memory or system bus) to keep the ownership for a guaranteed minimum amount of time. Watchdogs are features which monitor the usage of that resource during the time period the throttle is in effect. If the watchdog sees that resource go idle for a programmable period of time, it will override the throttle, take away resource ownership and give it to a device which can use it. Both throttles and watchdogs are controlled through programmable registers.

The function of the throttle controls the opening and closing of a "window" during which no requests for resource ownership will be honored. The throttles operate as 8-bit countdown timers based on OSC/2 (50 ns period). They begin counting down from the point at which resource ownership is obtained during which time the request window remains closed. When the count decrements to 0, the window is opened, allowing requests for ownership from other devices to be honored. At this point, their throttle is activated and they retain ownership for the amount of time programmed into their throttle register. Below is an example of throttle operation in which the host initially has ownership.

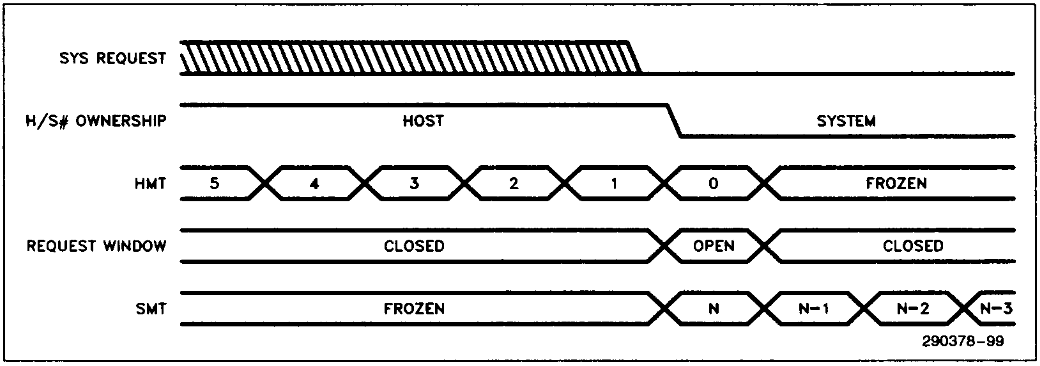


Figure 9-1. Throttle Operation

In the above example, there was no host request pending during the window, the host retains memory ownership and proceeds uninterrupted. Then, after another HMT countdown period, the window is re-opened and the system ownership request sampled again. Also, if the current cycle is locked, the opening of the window is overridden since the lock cycle must remain autonomous. It can be seen that if the system ownership request just misses an open window, or if a locked cycle occurs during an open window, the requesting device's next opportunity to be serviced is one throttle period away.

The use of the watchdogs provides control over the throttles from causing unnecessary waste. For example, the HMTW precludes the situation where the host takes memory ownership away from a system master, complete a cache line fill, continues processing out of the cache and allows the memory resource to go unused for the remaining duration of the throttle. Use of the HMTW would have caused the resource to be returned to the system master after a programmed amount of time (typically much shorter than the HMT). If the watchdog had not been used in the above case, a severe latency penalty would have been paid by the system master for the host's cache line fill since the system master was held off for the entire HMT period, even though the resource was idle. The watchdog registers have no effect if no other device is requesting resource ownership. The diagram below illustrates the function of the watchdogs.

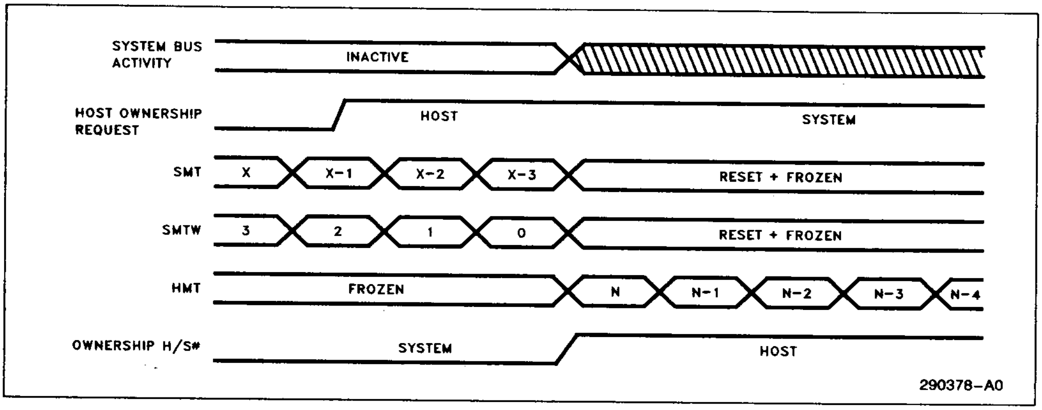


Figure 9-2. Watchdog Register Function

9.2 Host-to-Memory Throttle and Watchdog (HMT, HMTW)

The Host-to-Memory Throttle (HMT) guarantees host bandwidth to main memory. Once the host has actively taken ownership of memory from a system master (by the expiration of the SMT), the HMT provides a guaranteed time-slice for which the host will be able to hold on to ownership without having to relinquish it back to the system. During the time the host is using memory through the HMT, the EISA master's cycles are held-off by the 82359 de-asserting MRDY (EXRDY) until the host is done. (Note: the HMT is only active in Concurrent mode where memory ownership was forcefully taken from the EISA master, i.e., the SMT expires. It does not have any effect when the host has memory ownership, nor does it affect any system PST master devices since they are self-throttling and can not be held-off.)

As soon as memory ownership is taken away from the system master, an internal counter is loaded from the programmed value of the HMT. This counter is decremented by one until reaching a zero value, at which time the throttle is said to have expired and ownership may be lost. During the time the HMT is in effect, the 82359 actally holds-off memory requests (HMREQ) from reaching the host PST. Once the HMT expires, any pending HMREQ's which had been held-off are allowed to reach the host PST. (At this point, it is up to the host PST to honor this request for memory ownership.)

Along with the HMT is its counterpart, the Host-to-Memory Throttle Watchdog (HMTW). This "watchdog" provides control over the HMT. Once the throttle has started, the host must use the resource or lose it. The resource must not sit idle. Specifically, the programmed value of the HMTW Register is loaded into a counter at the end of all host cycles while the host throttle is in effect. This counter begins decrementing (using the OSC/2 timebase) as long as the no host cycle is initiated. Should the counter reach zero before the HMT expires, the HMTW will override the HMT and return memory ownership to the system. If a new host cycle is started before the expiration of the HMTW, the HMTW will be reset to the programmed value and once again begin counting down from the next occurrence of idle time on the host bus. Note that the HMTW is **not** reloaded in event of an aborted host cycle (ST# negated due to cache read hit).

9.3 System-to-Memory Throttle (SMT)

The SMT regulates EISA master ownership of memory. (System PST masters are unaffected by the SMT since they are self regulating.) The SMT register defines a number of system time increments (based on the OSC input divided by 2) during which the system master owns the memory. The SMT is active in Concurrent mode only.

As an example of SMT operation, assume the SMT value is programmed for 20. This results in an EISA master being guaranteed exclusive ownership of main memory for 1 µs. If an EISA master arbitrates for the memory and subsequently finishes using it and releases ownership in less than 1 µs, the SMT will never expire and the master will run uninterrupted.

If, on the other hand, the master still owns the memory after the programmed 1 μs value, the SMT opens the request window to see if a host-to-memory cycle is pending (HAS# asserted with HMACK asserted) or if an internal refresh request is pending. If so, it takes the memory away from the system master and gives it to the requesting device.

Although the term "taken away" is used, its meaning may be misleading. When memory is actively taken away from the EISA/ISA master, it is done so transparently from the EISA/ISA master's point of view. The 82359 accomplishes "taking memory" from the EISA/ISA master by negating MRDY (if the subsequent cycle is to a main memory address), causing the master to be stalled in wait states waiting for the current cycle to finish. From the master's point of view, it can not tell that memory had been taken away and only notices a long access time.

If the SMT expires during an EISA burst cycle, the throttle is overridden and the burst continues uninterrupted.

Note that a larger window value optimizes bus master efficiency via fewer interruptions and longer stretches of snoop line hits. A smaller window value reduces cache miss latency on the host bus.

9.4 Host-to-System Throttle and Watchdog (HST, HSTW)

The Host-to-System Throttle provides the host a guaranteed minimum system bus bandwidth independent of Concurrent/Non-concurrent mode. This throttle is active in both Concurrent and Non-concurrent mode. This is done by the 82359 internally masking off SHOLD requests from the system arbiter for the throttle's programmed time.

There are two programmable elements; the HST and its watchdog, the HSTW. These function very similar to the Host-to-Memory Throttle and Watchdog. The HST is triggered only once upon SHOLD falling. The HSTW, on the other hand, is reloaded every time the host accesses the system bus. Thus, as long as the host uses the system bus at least once every HSTW period, the HSTW pulse can never expire (except when the HST expires). If the host fails to use the system bus for an HSTW interval, the HST pulse is overridden and SHOLD unmasked.

Note that in concurrent mode, the HSTW is reloaded any time the CPU access the system bus. Host-to-main memory cycles (which do not reach the system port) do not impact HSTW. However in non-concurrent mode, the HSTW is reloaded upon any host access to main memory or the system bus, since these resources are considered one resource. Note that the HSTW is not reloaded in the event of an aborted host cycle (ST# negated due to cache read hit).

The HST throttle takes on an especially critical role when concurrency is turned off. In this case, the memory and system bus are considered one resource and thus the HST not only guarantees host-to-system bandwidth, but also host to memory bandwidth.

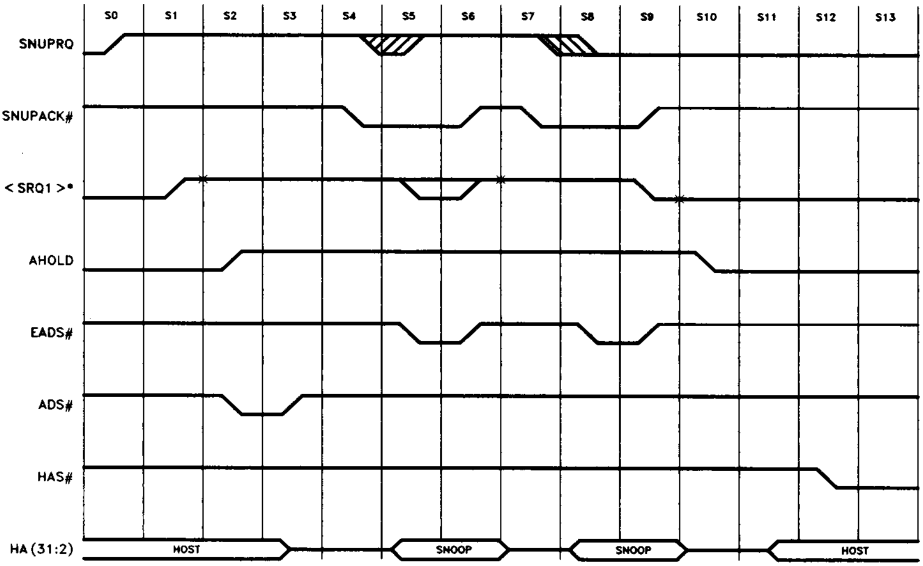
10.0 SNOOP FILTER

The 82359 increases host bus bandwidth by eliminating redundant cache invalidation cycles to cache lines which have been marked "dirty". Since the host address lines are used for transferring snoop addresses, elimination of these redundant snoop cycles allows the host bus to be used for host to memory or host to system cycles with fewer interruptions.

To eliminate redundant snoops, the 82359 uses a latch to hold the address of the previous snoop. This latch is loaded with the system address at the end of every system to main memory write cycle. When the next snoop address comes along, it is compared with the address held in the latch. This comparison, along with the Cache Line Size register, allows the 82359 to determine if the current snoop will be to the cache line invalidated in the previous snoop cycle. If this is the case, the snoop is not broadcast to the host. If the comparison is determined to be to a different cache line than the previous snoop, the 82359 requests a snoop cycle via the SNUPRQ, SNUPACK# protocol. The snoop filter is cleared whenever a change in memory ownership or bus ownership occurs.

Should either SNUPRQ be high or SNUPACK# be low (indicating that a pending snoop cycle is present) at the start of a system PST master memory write cycle, MRDY is driven low until SNUPACK# rises, thus causing this memory write to be stretched until the previous snoop is complete. If the memory write is an EISA burst cycle, MRDY is dropped low in the subsequent write cycle for only one BCLK, regardless of the rising edge of SNUPACK#.

When designing the external PST logic that generates invalidations cycles to the host cache, care should be taken to minimize the SNUPRQ to SNUPACK# latency. Specifically, an invalidation cycle should be given priority over the start of a new host initiated memory cycle. If the SNUPACK# latency is too long (greater than 180 ns with no accummulated snoops), there is a risk that the host cache could lose the snoop address during EISA or ISA write cycles. For the same reason, it is also important for the external PST snoop logic to process a snoop cycle at an equal or faster rate than the incoming memory write cycles being snooped. Although the 82359 will add one wait state for EISA burst write cycles during a snoop miss, under extreme circumstances where each consecutive memory write cycle is a miss to the snoop filter and the host CPU is running with a relatively slow clock, a snoop address could be lost if the snoop rate is not handled adequately. The figure shown below is a waveform example for an i486 host PST design demonstrating how the snoop logic can effectively handle these issues.



*Synchronized SNUPRQ

Figure 10-1. Host PST Snoop Cycle Operation

In state S0 the 82359 activates SNUPRQ in response to a memory write cycle on the system bus. SNUPRQ is synchronized and sampled by the host PST logic which results in the activation of AHOLD in state S2 and SNUPACK# in state S4. Note also, that in this example the processor happens to activate ADS# at the same time AHOLD is asserted. In this case, the snoop logic should give priority to AHOLD over the assertion of HAS# to cut down on the latency of SNUPRQ to SNUPACK#. EADS# is asserted in S5 after the snoop address meets the setup requirements for the cache invalidation cycle. In order to maintain a high snoop cycle throughput, AHOLD is asserted for 1 more cycle after EADS# is negated in S6 to allow for a second faster "Demand mode" type cycle. If synchronized SNUPRQ continues to be sampled active as in S6, then AHOLD will remain asserted until the next snoop cycle is complete. AHOLD is then negated in S10 after no more snoop requests are pending. The host PST then allows the ADS# that was stalled back in S2 to initiate a memory cycle by activating HAS# in S12.

11.0 CONCURRENCY

The 82359 may run in one of two modes: (1) Concurrent Mode, and (2) Non-concurrent Mode. The mode is selected by programming Mode Register C, bit 0 appropriately.

11.1 Concurrent Mode

In Concurrent Mode, the system bus and main memory are considered two separate resources. In this way ownership of one is completely independent of ownership of the other. For example, an EISA master may own the EISA bus and execute EISA cycles without ownership of main memory. At the same time, the host may own memory and may execute cycles to it regardless of any EISA activity.

Secondly, while in Concurrent Mode, the system interface of the 82359 monitors only the EISA signals and disregards the ISA signals. Instead, it relies on the 82358DT EBC to do ISA-to-EISA cycle translations.

11.2 Non-Concurrent Mode

In Non-concurrent Mode, the system bus and main memory become one resource. This requires that system master cycles run only after the system master has gained ownership of the system bus AND main memory. Likewise, host cycles must gain ownership of the system bus regardless of the cycle's destination. This mode does not allow system and host bus concurrency.

In Non-concurrent Mode, both ISA and EISA signals are visible (MRDC#, MWTC# and START#, CMD#, MSBURST#, EXRDY). The 82359 will monitor START# as well as MRDC#/MWTC# and locks out either one depending on what it sees first. For example, if the 82359 sees MRDC# before START#, it will use MRDC# to run the cycles and will not pay attention to START# (i.e., the 82359 runs as an ISA slave and does not request cycle extension). If it sees START# before a MRDC# or MWTC#, it ignores the MRDC#/MWTC# and runs as an EISA slave.

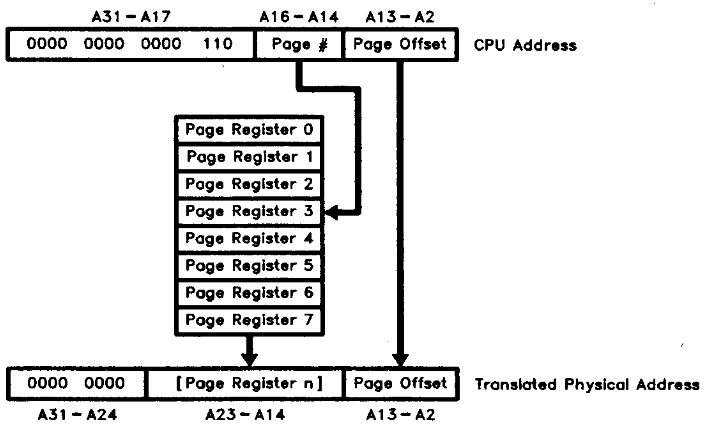
One application of Non-concurrent Mode is support for ISA masters that do not utilize CHRDY for ISA cycle extension. This group of masters runs memory cycles based on predetermined standard cycle lengths asynchronous to BCLK. Recall that the 82359 looks at the address of an EISA or ISA cycle after the cycle has begun and then, from this address, determines if the cycle is to main memory. If so, memory ownership is requested from the host PST via the HMREQ/HMACK protocol. Until memory ownership is acquired, the system master is held off with EXRDY or CHRDY. Since a few ISA masters do not understand CHRDY, it is not possible for the 82359 to hold these masters off while the memory ownership request is taking place. Therefore to support these kinds of masters on the system bus, Non-concurrent Mode is used, causing the 82359 to gain memory ownership before the system master is granted system bus ownership. This guarantees that 82359 can complete the memory cycle during the expected cycle length. Therefore to guarantee support of these masters, the user must program the 82359 to run under Non-concurrent Mode so that an ISA master will not initiate a cycle unless it owns both the system bus and main memory.

A second situation in which a Non-concurrent system is required is a design which incorporates the 82385 Cache Controller. Since the 82385 can only perform snoops while in HOLD, a conflict may arise. If the host starts a host-to-system cycle while a system master is writing to main memory (and thus the 82359 forwarding snoop cycles to the host), a deadlock may occur: the host is held off waiting for system bus ownership and the system is held off waiting for the snoop cycle to finish before proceeding. To eliminate the possibility of this occurring, Non-concurrent Mode should be used in 82385 systems. The AHOLD feature of the i486 and 82395 Cache Controller allows snoops to proceed while not in the HOLD state, and thus eliminates this problem altogether.

12.0 LIM SUPPORT

The 82359 implements LIM in hardware. Eight LIM Page Translation Registers are provided for support of up eight 16K LIM pages. An additional register is included to globally enable or disable LIM translation for those systems which may support LIM through software.

The diagram below illustrates the LIM translation process. The host address bits 16:14 select one of eight LIM translation registers. All eight LIM registers are assigned to the expansion ROM space 768K–896K (C0000h–DFFFFh), each one having its own unique 16K address range. Each translation register holds a 10-bit page number, each page being 16K. The value of these ten bits plus the lower A13:2 from the host address point to a 16K page in the 0 to 16M address range. (The upper address A31:24 are loaded with zeros, limiting the LIM translation registers to point in the lower 16M of main memory.)



290378–97

Figure 12-1. LIM Translation Process

When the host would like to access a 16K page from non-DOS memory (1M and above), the host programs a LIM translation register to point to the desired page. Now, when the host accesses the DOS accessible memory assigned to the programmed register, the translation will occur, mapping the desired page into the area assigned to the LIM register. The programming and mapping of LIM is typically handled by an EMM software driver.

Each LIM register may be individually enabled or disabled through software. By using all eight LIM registers, eight different 16K pages of non-DOS main memory may be accessed.

Two limitations are placed on the LIM translated memory. First, all memory translated by the LIM process must exist in the 0–16M range. Memory above 16M is not accessible through the LIM registers. Secondly, only memory controlled by the 82359 is available to the LIM translation process. Memory which may exist on add-in cards can not be translated.

One other limitation exists which is associated with 4-way dword interleaved memory using DRAMs with 4M address depth. Due to the way the 82359 uses host address bits in the translation, LIM blocks become 32K in size. Therefore the even LIM registers should be programmed for the lower half of aligned 32K blocks and the odd LIM registers programmed for the upper half address of aligned 32K blocks. The enforcement of all these rules is left to the EMM software driver.

13.0 THE INTERNAL REGISTERS

13.1 Programming the Internal Registers

The following section describes the way in which the 82359's internal registers are accessed. Both hardware and software issues are presented.

13.1.1 SOFTWARE CONSIDERATIONS

Each of the 82359's internal registers is given an 8-bit address, or "offset", through which register accesses may be made. The internal registers may *only* be accessed from the Host Bus (the 82359 physically precludes accesses from system masters). Register accessing is done via an indexing scheme through two I/O ports called the Index Register and Data Register, which by default, reside at I/O ports 22H and 23H respectively. In this scheme, I/O port 22H points to the desired internal register and I/O port 23H is the gateway (either read or write) to that register. The content of the Index Register is disabled after each 23H access, thus register programming must always follow a 22H/23H I/O access. All of the internal registers are accessible via this indexing scheme.

In order to expand the number of registers which use this indexing scheme from 256 and to allow multiple IC's within a system to utilize the same indexing scheme, an additional level of hierarchy has been added when accessing the internal registers. Each IC in the system with internal registers will have a unique ID which has to be written into the Chip ID Register (CIR) before accessing any of the internal registers of the device. The 82359 has been assigned two Chip ID's, 01H and A0H. All the internal registers except for LIM have been assigned a CIR value 01H. The LIM registers have been assigned a CIR of A0H.

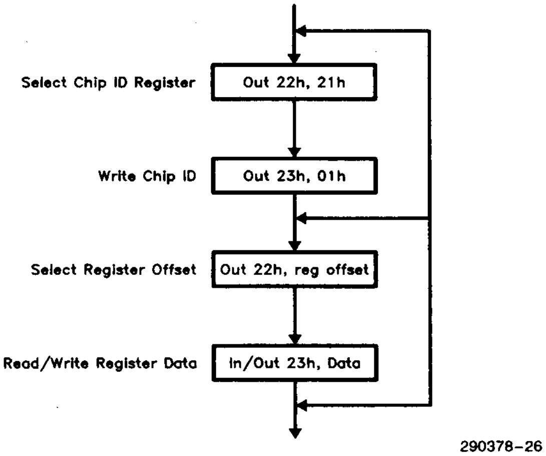


Figure 13-1. Register Indexing

Note that since a 22H access is always required before a 23H access, it is necessary to mask interrupts for the duration of the 22H/23H sequence. NMI routines are not allowed to access the 82359 through 22H/23H unless all routines that use 22H/23H also mask NMI's. If software that uses 22H/23H does not mask NMI's and an NMI routine must use 22H/23H, then this condition should be treated as non-recoverable.

Writing to any internal register will cause the 82359 DEN# signal (externally qualified with SW/R#) to flush the host cache.

Also note that if an interrupt service routine is going to modify the CIR to point to a different device, it must save and then restore the previous value.

The internal registers of the 82359 should only be programmed when all of the RAS# lines are inactive. Several of the 82359 registers affect the address translation logic within the 82359 (the DRAM population registers, offsets 00h to 03h; the remap control registers, offsets 4Eh, 83h and 84h and all of the LIM registers). Programming the above registers will not cause the RAS# signals to go inactive. If any of these registers are changed while RAS# remains active, and the next cycle is a memory access to an active page, the 82359 will run the cycle as a page hit cycle which assumes the old memory configuration. The following procedure should be used when programming any of the internal 82359 registers:

a. program the 82359 into 0 active RAS# mode (RAS# mode register, offset 06h)

b. run a dummy memory read cycle to drive the RAS# signals inactive

c. program the appropriate registers within the 82359

d. program the 82359 back to 1 or 2 active RAS# mode

The HKEN#, HWP# and HUSR# are not guaranteed to be valid when accessing the 82359's internal registers. These three signals may oscillate or glitch during access to the internal registers. The HKEN#, HWP# and HUSR# signals should be qualified by the host M/IO# and host W/R# signals if these outputs are used during host IO cycles.

The 82359 does not allow refresh cycles to occur when the system accesses the 82359's internal registers and when the 82359 is programmed for de-coupled refresh mode. The DRAM refresh latency time may be exceeded if multiple back-to-back accesses to the internal registers are executed without any other cycles between them.

Do not write software which will execute more then five accesses to the internal 82359 registers in a row. If more than five accesses must be done in a row, place a dummy I/O cycle between the back-to-back accesses to the 82359 internal registers.

NOTE: Many of the 82359's internal registers contain bits which are undefined. These are indicated by an "*" or a shaded bit position in register diagrams. Software which programs the internal registers must program these bits to "0". When reading the register, these undefined bits should be masked out since their state is undetermined.

13.1.2 HARDWARE CONSIDERATIONS

The data bus used in transferring register contents into and out of the 82359 is the lower 8 of 9 bits of the DRAM address lines, MADDR(8:0). Should the 82359 see a host originated I/O cycle to port 22H or 23H, it will assert its DEN# signal and forward the cycle to the system bus. During reads of the internal registers, DEN# is only asserted when the CIR register matches the 82359's chip ID (01H or 0AH).

Even though the cycle is accessing internal registers of the 82359, the cycle is sent to the system bus as any normal I/O cycle. From here the cycle makes the full round of EISA bus to X-bus to MADDR bus before the data reaches the 82359. When programming the internal registers in non-EISA applications, the CMD# input must still be driven low as specified in the "A.C. Specifications" section. In addition, IASALE# must be low at the assertion of CMD# to latch a proper address. Other system-side signals (such as START# and BCLK) are not required for correct programming of the internal registers in non-EISA applications.

The assertion of the DEN# signal is typically used to enable a '245 buffer which allows the I/O data to be transferred between the X-bus (peripheral bus) and the MADDR lines. Should the I/O cycle be a read of an internal register and the CIR contents matches the 82359's chip ID, the 82359 will drive the data on the MADDR lines. If the cycle is a write to an internal register, the MADDR lines are tri-stated, allowing the external '245 to drive the write data.

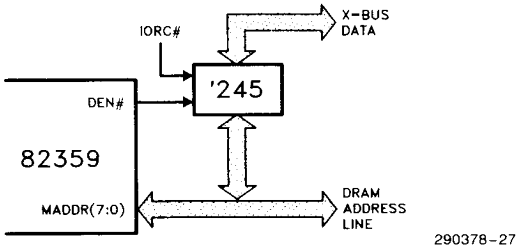


Figure 13-2. DEN#

Since many of the internal registers define the configuration of the system, the DEN# signal should be used to flush the host cache so as to keep memory mapping consistent with cache mapping/contents.

13.2 Register Listing

Configuration Registers

00h	Memory Row 0
01h	Memory Row 1
02h	Memory Row 2
03h	Memory Row 3
04h	DRAM Speed Register
05h	Line Size Register
06h	RAS# Mode Register
07h	Block Cache Enable
08h	Mode Register A
09h	Mode Register B
0Ah	Mode Register C
85h	Cache Control Register

Programmable Timing Registers

10h	Host Timing
11h	Host to System Timing
12h	System Timing
13h	Row Precharge
14h	Row Timing
15h	Column Timing
16h	CAS# Pulse Width
17h	CAS# to MDS# Delay

Index Addressing Registers

21h	Chip ID Register
22h	Virtual Index Register
23h	Virtual Data Register

Parity Error Trap Registers

28h–2Ch	Parity Error Trap Register

Cycle Length Feedback Registers

30h	CYCLN Read Page Hit
31h	CYCLN Read Page Miss
32h	CYCLN Read Row Miss
33h	CYCLN Write Page Hit
34h	CYCLN Write Page Miss
35h	CYCLN Write Row Miss

Block Enable Registers

40h–41h	Lower Memory
42h–43h	Video RAM
44h–45h	Expansion ROM
46h–47h	BIOS

Memory Remap Registers

83h–84h	Split Address Register
4Eh	Remap Enable Register

Programmable Attribute Map Registers

50h–53h	PAM Register 0
54h–57h	PAM Register 1
58h–5Bh	PAM Register 2
5Ch–5Fh	PAM Register 3

Throttles and Watchdogs

8Bh	System Memory Throttle, SMT
8Ch	Host Memory Throttle, HMT
8Dh	Host Memory Watchdog, HMTW
8Eh	Host System Throttle, HST
8Fh	Host System Watchdog, HSTW

Resource Allocation Monitors

90h	RAM Enable
91h	RAM Disable
92h–93h	Elapsed Time, ETC
94h–95h	Host to Memory, HMR
96h–97h	System to Memory, SMR
98h–99h	Host Memory Ownership, HMO
9Ah–9Bh	System Bus Ownership, SBO
9Ch–9Dh	Host Request of Systems, HRS
9Eh–9Fh	Memory Ownership Transfer, MOT

LIM Registers

00h	LIM Control Register
80h–81h	LIM Page 0
82h–83h	LIM Page 1
84h–85h	LIM Page 2
86h–87h	LIM Page 3
88h–89h	LIM Page 4
8Ah–8Bh	LIM Page 5
8Ch–8Dh	LIM Page 6
8Eh–8Fh	LIM Page 7

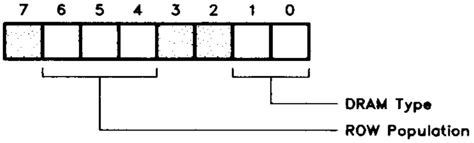
13.3 Detailed Register Descriptions

MEMORY CONFIGURATION REGISTERS

Register Name	Offset	Default Value	Access
Row 0	00h	*000**01b	W/R
Row 1	01h	*111**01b	W/R
Row 2	02h	*111**01b	W/R
Row 3	03h	*111**01b	W/R

Register Description

The Row Registers are programmed by the BIOS at power-up and tell the 82359 the size and population (dword interleave factor) of each of the four rows in the memory array.



290378-28

Field Descriptions

DRAM Type: This field specifies the size of the DRAM per row in terms of the address depth. The memory array supports 64K, 256K, 1M, and 4M DRAM address depths. All memory within the same row must have the save address depth, although differing rows may have different DRAM address depths. The width (x1, x9, x36 bits) of the particular DRAM being used to implement the row is of no consequence. For example, 256K x 1 bit DRAMs are specified the same as 256K x 9 bit DRAMs. The following table shows the correspondence of the "DRAM Type" field to the size of the DRAM.

Bits 1 0	DRAM ADDRESS DEPTH
0 0	64K
0 1	256K (default Row 0, 1, 2, 3)
1 0	1M
1 1	4M

Row Population: This field determines which dwords are populated in the corresponding row (i.e., the row's dword interleave factor). Each dword is 32 bits wide plus 4 parity bits. Each row can be populated with a maximum of 4 dwords for a total width of 144 bits. Bank population combinations are limited to those decoded in the following table:

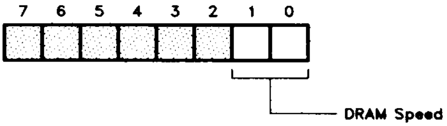
Bits 6 5 4	Bank Population
0 0 0	0 (default Row 0)
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	0, 1
1 0 1	2, 3
1 1 0	0, 1, 2, 3
1 1 1	Empty (default Row 1, 2, 3)

DRAM SPEED REGISTER

Register Name	Offset	Default Values	Access
SPEED	04h	******11b	W/R

Register Description

The Speed Register is driven onto the SPEED(1:0) pins and reflects the access time of DRAM in main memory. This is an open-collector register typically used by the PSTs to determine the required wait-state count according to the speed of DRAM in main memory.



290378-29

Field Descriptions

DRAM Speed: These bits connect to a quasi-bidirectional port of the DRAM controller (see the SPEED(1:0) pin description for more details). Writing a "1" to any of the two bits in the register causes its corresponding pin of the SPEED(1:0) output to float. Writing a "0" causes that pin to be pulled down to low.

A read of the port actually looks at the SPEED(1:0) pins themselves, which are a wired-"OR" of the Speed Register output and of system sources that can pull the SPEED(1:0) pins low, such as the SIMMs. Effectively, if either bit in the register is written to a "1", that bit will be the reflection of the external SPEED(1:0) pin.

Typically these bits are interpreted as follows:

Bits 1 0	DRAM Speed
0 0	fastest
0 1	↓
1 0	↓
1 1	slowest (default)

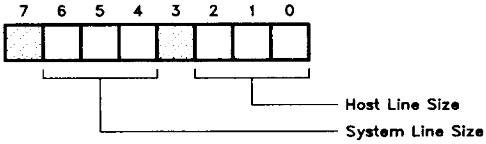
This port is intended to support DRAM SIMMs that provide straps to ground to identify the speed of the DRAM. The use of the SPEED information is typically used in conjunction with the PST interface for determining the number of wait states required for Host or System burst cycles. The state of the SPEED register has no effect on the internal functionality of the 82359 and the designer is free to modify the definition of the register and output pins as required.

LINE SIZE REGISTER

Register Name	Offset	Default Values	Access
Line Size	05h	*100*100b	W/R

Register Description

The Line Size Register (along with the dword interleave factor) is used by the 82359 to determine how many memory cycles are required to fulfill the host or system access. This register also plays a role in determining when to filter snoop cycles.



290378-30

Field Descriptions

Host Line Size: The Host Line Size is used in conjunction with the dword interleave factor to determine the number of data fetches required from the main memory for host initiated read cycles. For example, a burstable access to 1-way interleaved memory and the 82359 programmed for a 4 dword line size requires that four memory cycles be run.

Also, the Host Line Size reflects the width of the host cache line for the purpose of snoop filtering. With the Host Line Size set to the minimum (1 dword), all system memory write cycles are forwarded to the host CPU bus as cache invalidation cycles. With a programmed Host Line Size greater than one dword, back-to-back system memory write cycles to the same cache line will not be propagated to the host bus, and as a result, no cache invalidation cycles will occur.

Bits 6:4 or 2:0	Line Size
0 0 0	reserved
0 0 1	reserved
0 1 0	4 bytes
0 1 1	8 bytes
1 0 0	16 bytes (default)
1 0 1	32 bytes
1 1 0	64 bytes
1 1 1	reserved

An important note when programming the 82359 for line sizes greater than 16 bytes: the 82359 will perform a maximum of four consecutive accesses during a burst cycle. This means that a successful 32-byte burst transfer can only be run from 2-way or 4-way interleaved memory. Similarly, a 64-byte burst access can only be run from 4-way memory. The system design must ensure that the proper memory width is in place if the line size is programmed for greater than 16 bytes.

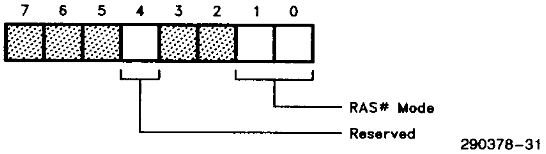
System Line Size: The System Line Size, along with the row's dword interleave factor, is used during system PST initiated read cycles of main memory for determining the number of data fetches required to complete the system access. The System Line Size has no effect on snoop filtering. The decode of the System Line Size is the same as Host Line Size.

RAS# MODE REGISTER

Register Name	Offset	Default Values	Access
RAS# Mode Register	06h	***1**01b	W/R

Register Description

The RAS# Mode Register determines the characteristics the 82359 will follow as to leaving RAS# lines active after the main memory access is complete.



Field Descriptions

RAS# Mode: This field controls the behavior of the RAS# signals for the purposes of RAS Mode control. These bits are decoded as follows:

Bit 1 0	RAS# Mode		
0 0	0 Active RAS# Mode	(Non-Page Mode)	
0 1	1 Active RAS# Mode	(Page Mode)	(default)
1 0	2 Active RAS# Mode	(Page Mode)	
1 1	Dynamic RAS# Mode	(Page Mode)	

0 Active RAS# Mode: RAS# goes inactive at the end of every host or system cycle. (For burst type accesses, RAS# remains asserted until the end of the burst cycle.)

1 Active RAS# Mode: Only the RAS# for the current row being accessed will remain active at the end of the memory cycle. The RAS# signals for the rows not accessed are forced back inactive (if previously active). This mode allows for one memory row to remain active between memory cycles.

2 Active RAS# Mode: Two RAS# signals, the one most recently used from the combination of row 0 and row 1, and the one most recent from the combination of row 2 and row 3, are allowed to stay active between DRAM cycles. This mode allows 2 rows to remain active between memory cycles.

Dynamic RAS# Mode: Dynamic RAS# Mode allows the 82359 to control when RAS# should be left active. By monitoring the number of sequential DRAM page hits and sequential page misses, the 82359 will automatically switch to the mode which best suits the type of DRAM accesses being made at that time. Initially the 82359 uses 2 Active RAS# Mode until three sequential DRAM page miss cycles occur, in which case the 82359 switches to 0 Active RAS# to hide the recurring row precharge time. When the 82359 detects three sequential page hit cycles, it automatically switches back to 2 Active RAS# Mode to take advantage of the faster page hit access times.

Reserved Bit: This is a reserved bit. This bit should be programmed to a "1" to insure correct functionality of the 82359.

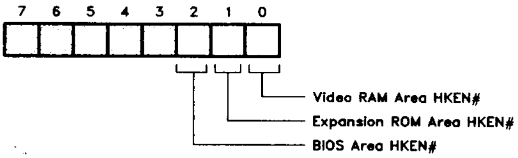
BLOCK CACHE ENABLE REGISTER

Register Name	Offset	Default Values	Access
Block Cache Enable	07h	*****111b	W/R

Register Description

The Block Cache Enable register is used for determining the cacheability of BIOS, video, and Expansion BIOS address space. Setting a bit to a "1" defines the corresponding address range to non-cacheable status. Should a host address fall within one of the ranges, the HKEN# output will be de-asserted and remain de-asserted until a new host address is latched (with the falling edge of HAS#).

These and other areas of memory can also be declared as non-cacheable through the use of the programmable attribute map. In case of an overlap of cacheability map information, HKEN# will be negated (equals "1") if it is shown to be negated in any section. Also, should cacheing be disabled through the Cache Control Register (CCR), HKEN# will remain de-asserted regardless of the setting of this register.



290378-32

Field Descriptions

Video RAM Area HKEN#: Defines the area 640k–768k (A0000h–BFFFFh) as a non-cacheable range.

Expansion ROM Area HKEN#: Defines the area 768k–896k (C0000–DFFFFh) as a non-cacheable range.

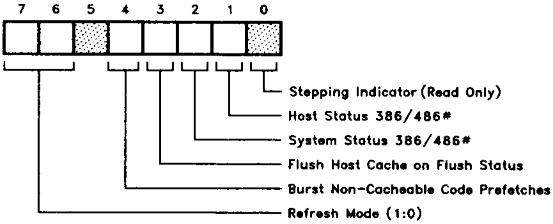
BIOS Area HKEN#: Defines the area 896k–1024k (E0000–FFFFFh) as a non-cacheable range.

MODE REGISTER A

Register Name	Offset	Default Values	Access
Mode Register A	08h	00*01000b	W/R

Register Description

Mode Register A is used to set 82359 performance parameters such as refresh mode, host status, system status, flush activation, and burstable code prefetches.


7 6 5 4 3 2 1 0
Stepping Indicator (Read Only)
Host Status 386/486#
System Status 386/486#
Flush Host Cache on Flush Status
Burst Non-Cacheable Code Prefetches
Refresh Mode (1:0)

290378-33

Field Descriptions

Stepping Indicator: A Read-Only bit which allows the BIOS to detect which stepping of the 82359 is installed in the system and determine whether to enable/disable concurrent mode of operation during POST. A "1" on this bit indicates the presence of an A-1 device, a "0" indicates that an A-2 device is installed.

Host Status: This bit determines if 386 or i486 cycle definitions will be used for host cycles. If set to "1", the 82359 will interpret HM/IO#, HW/R#, HD/C# and HBE(3:0)# in the manner defined by the 80386 processor. If set to "0", the host will use the i486 cycle definition. (See the pin descriptions of HM/IO#, HW/R# and HD/C# for more details on cycle definitions.)

System Status: This bit determines if 386 or i486 cycle definitions will be used for system cycles. If set to "1", the 82359 will interpret SM/IO#, SW/R#, SD/C# and SBE(3:0)# in the manner defined by the 80386 processor. If set to "0", the host will use the i486 definition. (See the pin descriptions of HM/IO#, HW/R# and HD/C# for more details on cycle definitions.)

Flush Host Cache: Setting this bit to a "1" causes the 82359 to assert the DEN# output pin upon detecting an i486 cache flush special cycle. Should this bit be a "0", the 82359 will ignore flush cycles and DEN# will only be asserted for 82359 internal register accesses.

Typically, any time DEN# becomes asserted, the host cache should be flushed due to the possibility of a change in the system configuration. Therefore, by asserting DEN# for the i486 cache flush cycle, the flush will be indirectly sent to the host cache. (See the DEN# pin description for more details.)

Burst Non-Cacheable Code Prefetches: This bit enables the bursting of code prefetches, regardless of cacheability. Burst always occur in lengths specified by the Host Line Size Register and thus this bit should be cleared if the code prefetch line size is smaller than the cache line size, or if the host CPU does not burst code prefetches.

Refresh Mode: Bits 7 and 6 control the 82359's refresh mode according to the table below:

Bits 7 6	Refresh Mode
0 0	Coupled
0 1	Decoupled-1 cycle
1 0	Decoupled Burst-2 cycle
1 1	Decoupled Burst-4 cycle

Coupled refresh implies a system master initiated refresh (as indicated on the REFRESH# input). The refresh timings and the manner in which refresh is arbitrated look similar to a system initiated main memory read.

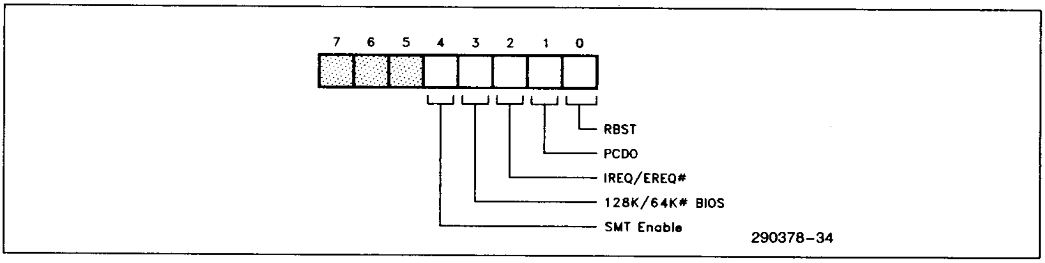
Decoupled refresh implies the 82359 handles main memory refresh internally, generating refresh requests off its own time base. In this case, the 82359 actually supports a state where it owns the memory. Decoupled refresh comes in three sub-modes: (1) Single Cycle Mode in which one refresh cycle occurs every 15 μs; (2) Two Cycle Burst Mode where one burst of two back-to-back refresh cycles are generated, and; (3) Four Cycle Burst where four back-to-back refreshes occur once every 60 μs. Decoupled Refresh Mode must be selected when using ISA masters in a system programmed for concurrent mode operation.

MODE REGISTER B

Register Name	Offset	Default Values	Access
Mode Register B	09h	***11101b	W/R

Register Description

Mode Register B is used for setting 82359 performance parameters having to do with cache, bursting, and BIOS size.



290378-34

Field Descriptions

RBST: Restricted Burst. The ability of the host to execute i486-like burst cycles for host-to-system accesses above 1 Meg is controlled by the setting of this bit. Setting this bit to a "1" causes host-to-system cycles above 1 Meg to be non-burstable regardless of cacheability. If this bit is cleared to "0", the burst restriction for addresses above 1 Meg is removed. (Host accesses to system addresses below 1 Meg are always non-burstable, regardless of the setting of this bit.) This bit should be cleared to "0" in an 82385 system before the cache is enabled.

PCDO: PCD Override. When set to "1", the 82359 PCD input overrides the internally generated burst indicator, allowing instead for the PCD input to directly control whether or not an access is burstable. When this bit equals "1" and bit 0 of the CCR indicates the cache is enabled, a low level on PCD will cause the 82359 to burst independent of the internal cacheability map. When cleared to "0", burstability is determined based on whether the 82359 decodes a cache line fill. In an 82385 system, this bit should be set to "1" and the 82359 PCD input driven by the 82385 MISS# output. (This bit should be set before the cache is enabled.)

IREQ/EREQ#: Internal System Bus Request/External System Bus Request. When set to "1", the SBREQ output pin is internally masked via bit 6 of the CCR (Cache Control Register) before being driven off the SBREQ pin. When cleared to "0", bit 6 of the CCR is directly reflected onto the SBREQ pin. (SBREQ now behaves as SBREQEN. SBREQEN is active high, and thus is actually the inverse of CCR, bit 6.) This bit should be cleared to "0" in an 82385 system before system bus requests are enabled via bit 6 of the CCR.

128K/64K# BIOS: When set to "1", the BIOS is assumed to be 128 kbytes. When "0", the BIOS is 64 kbytes.

The BIOS address maps are:

128K:	000E0000h–000FFFFFh and FFFE0000h–FFFFFFFFh
64K:	000F0000h–000FFFFFh and FFFF0000h–FFFFFFFFh

The effect of this bit is the determination of the address map to which the BIOSCS# output will respond. In case of a 64K BIOS, expansion bus devices may reside within 000E0000h–000EFFFFh. Also, main memory that may have otherwise been used to shadow BIOS in this range is now free to be either re-mapped, or to serve to shadow expansion ROMs in this range.

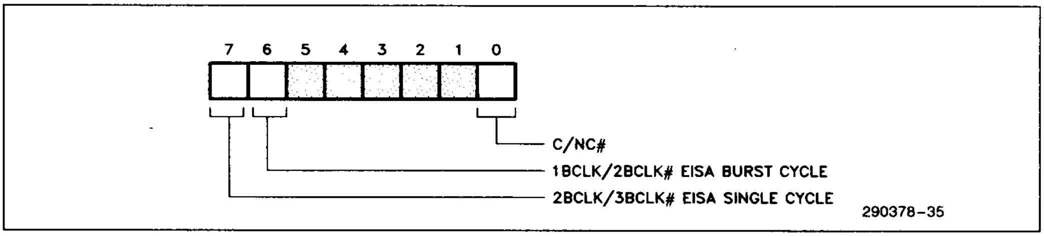
SMT Enable: This bit either enables or disables the SMT (System Memory Throttle). When set to "1" the SMT is enabled. Programming this bit to "0" prevents the SMT from expiring in concurrent mode. The default condition after power-on is a "1" (SMT enabled).

MODE REGISTER C

Register Name	Offset	Default Values	Access
Mode Register C	0Ah	00*****0b	W/R

Register Description

Mode Register C controls Concurrency/Non-concurrency, EISA burst speed, and EISA single cycle speed.



290378-35

Field Descriptions

C/NC#: CONCURRENT/NON-CONCURRENT# OPERATION. This bit globally enables or disables main memory/system bus concurrency. If operating concurrently, a system master can own the system bus independent of memory ownership, as well as the host owning main memory without owning the system bus.

If operating non-concurrently, the memory and system bus collapse into one resource. Therefore, both the memory and system bus must be owned by the host before the cycle can be run, regardless of the destination of the cycle, and system masters must own main memory before system bus ownership is obtained.

Concurrency must be turned off in systems which utilize the 82385 cache and in systems with ISA masters that do not honor slave initiated cycle extension (CHRDY).

1BCLK/2BCLK# EISA BURST CYCLE: 1 BCLK/2 BCLK# EISA BURST CYCLE. This bit determines whether the fastest EISA burst supported by the 82359/main memory subsystem is one or two BCLKs. Typically, this bit is set to "1" for 80 ns or faster DRAMs. For slower DRAMs, this bit can be cleared to "0" if timings don't support a full 8.333 MHz EISA bus with 1 BCLK bursts. However, a designer may choose a more efficient alternative such as slowing the bus slightly to 8.0 MHz or adding an external circuit to stretch BCLK. (Note that the 82359's asychronous host-to-system interface allows the exact EISA bus speed to optimally match EISA masters to DRAM parameters, with no impact on host-to-memory performance.)

2BCLK/3BCLK# EISA SINGLE CYCLE: This bit decides whether the fastest EISA single cycle runs with 0 or 1 EISA wait states. For EISA, 0 wait states implies a 2 BCLK cycle (nominally 240 ns), while 1 wait state implies a 3 BCLK cycle (nominally 360 ns). The 82359 typically runs 0 wait state for 80 ns or faster DRAMs. For slower DRAMs, this bit can cause a wait state to be added, or alternatively the EISA BCLK speed may be externally adjusted for a more efficient cycle.

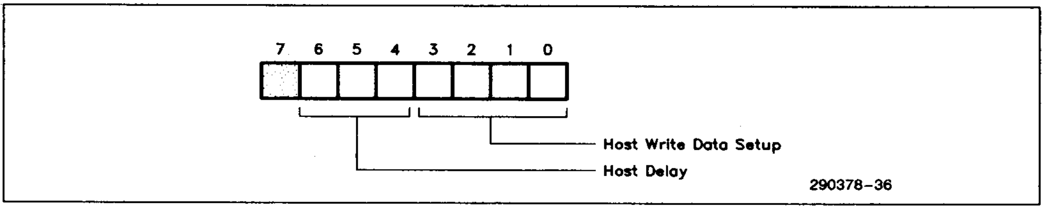
The 3 BCLK EISA Single Cycle option must be used when the 82359 is programmed for Concurrent mode/De-Coupled Refresh. The 2 BCLK EISA Single Cycle option may only be used in Non-Concurrent mode/Coupled Refresh mode.

HOST TIMING REGISTER

Register Name	Offset	Default Values	Access
Host Timing Register	10h	*1101000b	W/R

Register Description

The Host Timing Register controls address, status and data set-up time to the 82359's cycle start trigger for host accesses.



290378–36

Field Description

Host Write Data Set-Up (HP10): This 4-bit field is used to provide a delay between the host cycle start indicator (HAS# falling edge and HMACK de-asserted, or HAS# asserted and the falling edge of HMACK) and the point at which the 82359 asserts its CAS# lines during DRAM page hit write cycles. By providing this delay, the DRAM data set-up time can be controlled. The default value corresponds to 52.20 ns.

Bits 3 2 1 0	Delay
0 0 0 0	31.32 ns
0 0 0 1	33.93 ns
:	:
1 0 0 0	52.20 ns (default)

Host Delay (HP1): This 3-bit field is used to provide a delay between the host cycle start indicator (HAS# falling edge and HMACK de-asserted, or HAS# asserted and the falling edge of HMACK) and the point at which the 82359 actually begins to act on the host cycle. By internally delaying the start of the cycle, the 82359 has sufficient time to decode the address, generate attributes, determine DRAM page hit/miss status, and provide enough column address set-up to CAS(7:0)#. The ST# signal should be used to abort DRAM cycles only when the HP1 register is programmed to a value of 0 through 4 (inclusive). This delay becomes necessary as CPU speed increases. The default value of this field corresponds to 18.27 ns.

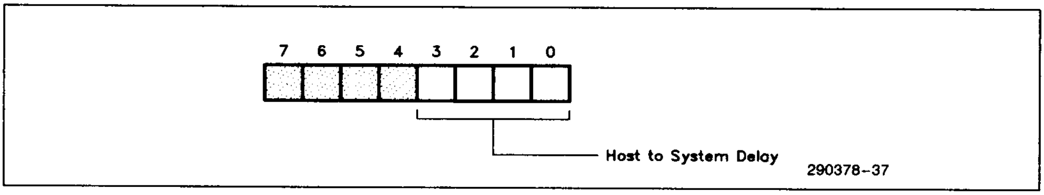
Bits 6 5 4	Delay
0 0 0	2.61 ns
0 0 1	5.22 ns
:	:
1 1 0	18.27 ns (default)

HOST TO SYSTEM DELAY REGISTER

Register Name	Offset	Default Values	ccess
Host to System Delay	11h	****1010b	W/R

Register Description

The Host to System Delay Register controls the host cycle trigger to delay to SAS# activation for host to system cycles.



290378–37

Field Description

Host to System Delay (HP11): This 4-bit field provides a delay between the host cycle trigger to SAS# activation for host cycles bound for the system bus. This delay provides a programmable set-up time for the system address and status before the system bus cycle is actually initiated. This register field is default to 67.86 ns.

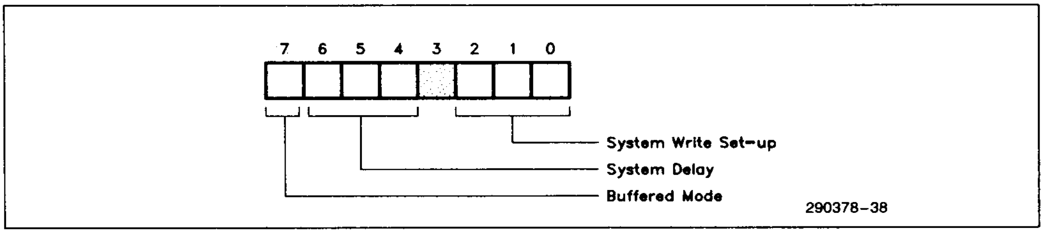
Bits 3 2 1 0	Delay
0 0 0 0	41.76 ns
0 0 0 1	44.37 ns
:	:
1 0 1 0	67.86 ns (default)

SYSTEM TIMING REGISTER

Register Name	Offset	Default Values	Access
System Timing Delay	12h	0101*101b	W/R

Register Description

The System Timing Register controls address, status, and data set-up times for system bus cycles to main memory.



290378-38

Field Description

System Write Set-Up (SP10): This field provides data set-up time to the DRAMs for system write cycles which cause a DRAM page hit. The delay is inserted between the system cycle start (SAS# falling edge and SMACK low, or SMACK falling edge and SAS# low) and the time at which the 82359 actually will begin the write sequence. This delay provides the DRAM with system data set-up to CAS# activation for DRAM write page hits. This field is default to 20.88 ns.

Bits 2 1 0	Delay
0 0 0	7.83 ns
0 0 1	10.44 ns
:	:
1 0 1	20.88 ns (default)

System Delay (SP1): This 3-bit field is used to provide a delay between the system cycle start indicator (SAS# falling and SMACK de-asserted, or SAS# de-asserted and the falling edge of SMACK) and the point at which the 82359 actually begins the system cycle. This delay adjusts the amount of address and status setup for the purpose of column address setup to CAS(7:0)#. The default value is 20.88 ns.

Bits 6 5 4	Delay
0 0 0	7.83 ns
0 0 1	10.44 ns
:	:
1 0 1	20.88 ns (default)

Buffered Mode: Setting this bit to a "0" causes the 82359 to follow A.C. timing specifications provided for "Standard Mode". Setting this bit to a "1" causes the 82359 to follow the A.C. timings specified for "Buffered Mode". Specifically, the Buffered Mode timings offer somewhat relaxed timings between BCLK and 82359 activity. This relaxation helps to compensate for the propagation delay encountered due to buffering of EISA address lines from the 82359 (and EISA data lines from the 82353).

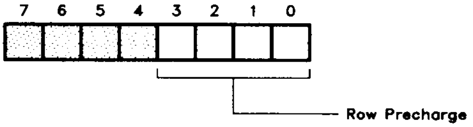
Bit 7	Mode
0	Standard Mode (default)
1	Buffered Mode

ROW PRECHARGE TIMING REGISTER

Register Name	Offset	Default Values	Access
Row Precharge	13h	****1110	W/R

Register Description

The register Row Precharge allows the programming of DRAM row precharge time for various speed DRAMs.



290378-39

Field Description

Row Precharge (P2): The Row Precharge Register should be programmed with a value which guarantees that the DRAM row precharge requirement is met. The actual row precharge time as seen by the DRAMs is specified by t_{63}. This field is default to 80.91 ns.

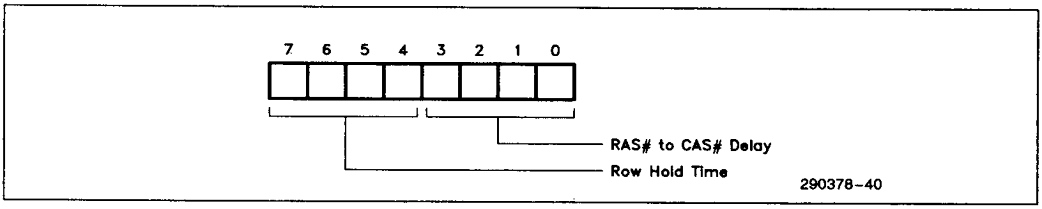
Bits 3 2 1 0	Delay
0 0 0 0	44.37 ns
0 0 0 1	46.98 ns
:	:
1 1 1 0	80.91 ns (default)

ROW TIMING REGISTER

Register Name	Offset	Default Values	Access
Row Timing	14h	01011000b	W/R

Register Description

The Row Timing Register controls timings related to DRAM row address specifications. Specifically row address hold time and RAS# to CAS# delay can be controlled through this register.



Field Description

RAS# to CAS# Delay (P4): This field should be programmed with the value of RAS# to CAS# delay required by the speed of DRAM in main memory. Actual RAS# to CAS# delay external to the 82359 is given by t_{62}. Default value for this field is 83.52 ns.

Bits 3 2 1 0	Delay
0 0 0 0	62.64 ns
0 0 0 1	65.25 ns
:	:
1 0 0 0	83.52 ns (default)
:	:
1 1 0 0	93.96 ns

Row Address Hold Time (P3): This field should be programmed for the row address hold time required by the particular speed DRAM in main memory. Actual row address hold time external to the 82359 is given by t_{71}. This register field is default to 46.98 ns.

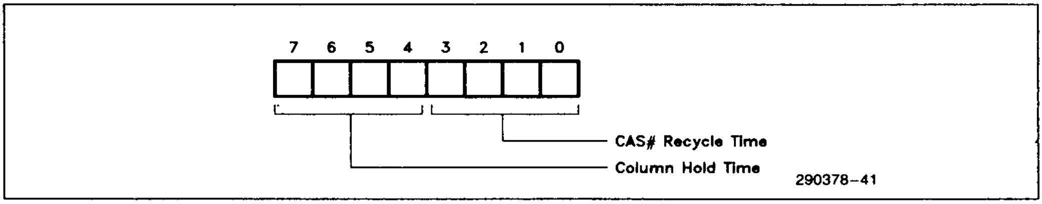
Bits 7 6 5 4	Delay
0 0 0 0	33.93 ns
0 0 0 1	36.54 ns
:	:
0 1 0 1	46.98 ns (default)
:	:
1 0 0 1	62.64 ns

COLUMN TIMING REGISTER

Register Name	Offset	Default Values	Access
Column Timing	15h	01111100b	W/R

Register Description

The programming of the Column Timing Register controls the CAS# characteristics. Specifically, the CAS# cycle time and column address hold are controlled.



Field Description

CAS# Recycle Time (P7): The programming of this field directly controls the amount of time from the initial falling edge of CAS(7:0)# to the next falling edge of CAS(7:0)# for burst cycles. It also indirectly controls the amount of CAS# precharge time. This field is default to 83.52 ns.

Bits 3 2 1 0	Delay
0 0 0 0	52.20 ns
0 0 0 1	54.81 ns
:	:
1 1 0 0	83.52 ns (default)

Column Hold Time (P5): This field should be programmed to support the DRAM's required column address hold from CAS# falling. This register also will effect valid column address setup for the next cycle. Default value is 28.71 ns.

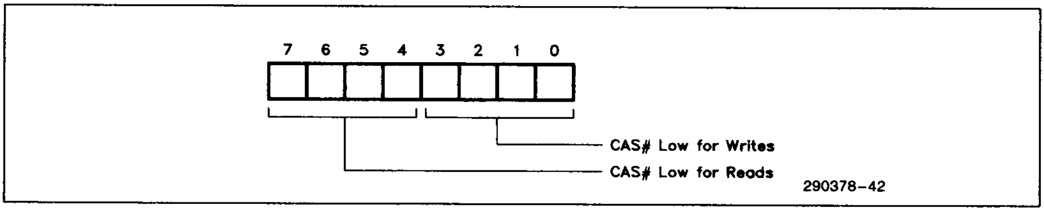
Bits 7 6 5 4	Delay
0 0 0 0	10.44 ns
0 0 0 1	13.05 ns
:	:
0 1 1 1	28.71 ns (default)
:	:
1 0 0 1	33.93 ns

CAS# LOW TIMING REGISTER

Register Name	Offset	Default Values	Access
CAS# Low Timing	16h	10000100b	W/R

Register Description

This register controls the amount of time that CAS(7:0)# is active. A 4-bit field is provided for both DRAM read and DRAM write cycles.



Field Description

CAS# Low for Reads (P8): The amount of CAS(7:0)# activation time for a DRAM read access is directly controlled by the value of this field. The default value is 62.64 ns.

Bits 7654	Delay
0000	41.76 ns
0001	44.37 ns
:	:
1000	62.64 ns (default)
:	:
1010	67.86 ns

CAS# Low for Writes (P9): The amount of CAS(7:0)# activation time for a DRAM write cycle is directly controlled by the value of this field. The default value of this field is 31.32 ns.

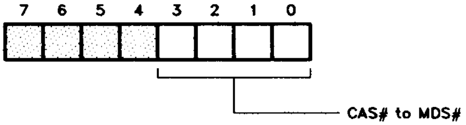
Bits 3210	Delay
0000	20.88 ns
0001	23.49 ns
:	:
0100	31.32 ns (default)
:	:
1010	46.98 ns

CAS# TO MDS# DELAY REGISTER

Register Name	Offset	Default Values	Access
CAS# to MDS#	17h	****1010b	W/R

Register Description

This register controls the delay of MDS# from the falling edge of CAS#. This delay can be thought of as the CAS access time of the DRAM since the rising edge of MDS# latches the memory read data into the 82353 Data Path device.



290378-43

Field Description

CAS# to MDS# (P6): The programmed value of this register controls the time delay between the event that caused CAS(7:0)# to be activated and the event that causes the rising edge of MDS# (Memory Data Strobe). Data should be valid for latching into the 82353 Data Path on the rising edge of MDS#. The default value of this register is 57.42 ns.

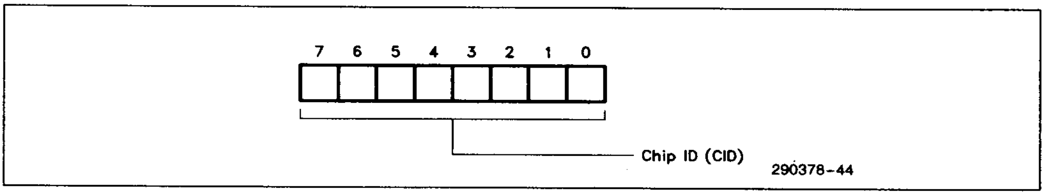
Bits 3 2 1 0	Delay
0 0 0 0	31.32 ns
0 0 0 1	33.93 ns
:	:
1 0 1 0	57.42 ns (default)
:	:
1 1 0 0	62.64 ns

CHIP IDENTIFICATION REGISTER (CIR)

Register Name	Offset	Default Values	Access
CIR	21h	11111111b	W/R

Register Description

The CIR holds the Chip ID number of the device currently selected to respond to the 22h, 23h scheme of accessing internal registers. Writing to register 21h is honored by every device following this indexing scheme, regardless of what chip is currently selected.



290378-44

Field Descriptions

CIR: This register is used to select specific IC's that should respond to I/O ports 22h and 23h indexed accessing scheme. Upon reset, the CIR will have a value of FFh and thus all IC's will be disabled for access via the indexing scheme (although writes to the CIR register are honored by all registered chips).

Chip ID	Device Name
00 h	reserved
01 h	82359 General Registers
02 h	LIO.E Internal Registers
03 h	reserved
04 h	.
.	.
A0 h	82359 LIM Registers
A1 h	reserved
.	.
FF h	none (default)

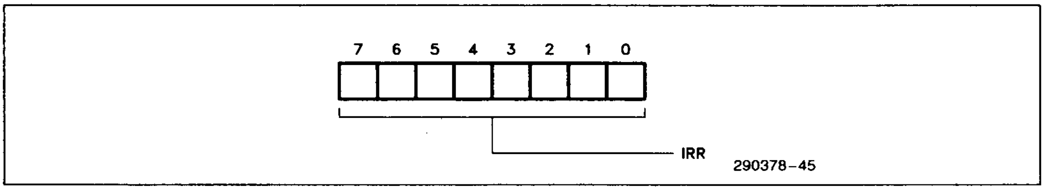
All the 82359's general registers can be accessed once a Chip ID of 01h has been written into the CIR register (except for the 82359's internal LIM registers which require a special Chip ID of A0h).

INDEX RELOCATION REGISTER (IRR)

Register Name	Offset	Default Values	Access
IRR	22h	22h	W/R

Register Description

The IRR register allows the I/O port address used for the indexing schemes register offset to be moved to a different I/O location.



Field Descriptions

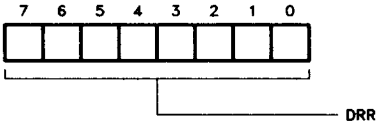
IRR: The I/O port that is used when writing the **index number** of an internal register for the indexing scheme is programmable through this register. Upon reset, this register defaults to 22h. By changing the contents of the Index Relocation Register, the physical address of the I/O port involved in specifying the register's offset can be moved anywhere in the I/O range 00–FFh.

DATA RELOCATION REGISTER (DRR)

Register Name	Offset	Default Values	Access
DRR	23h	23h	W/R

Register Description

The DRR register allows the I/O port address used for passing data to/from the 82359's internal registers to be moved to a different location in the I/O space.



290378-46

Field Descriptions

DRR: The I/O port that should be used when accessing the **data** of an internal register through the indexing scheme is programmable through this register. Upon reset, this register defaults to 23h. By changing the contents of the Data Relocation Register, the physical address of the I/O port involved in the transferring data can be moved anywhere in the I/O range 00–FFh.

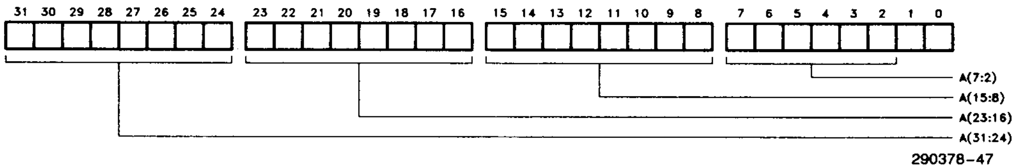
PARITY ERROR TRAP REGISTERS

Register Name	Offset	Default Values	Access
Parity Error A (7:2)	28h	xxxxxx**b	R
Parity Error A (15:8)	29h	xxxxxxxxb	R
Parity Error A (23:16)	2Ah	xxxxxxxxb	R
Parity Error A (31:24)	2Bh	xxxxxxxxb	R
Parity Error Status	2Ch	xxx*xxxxb	R

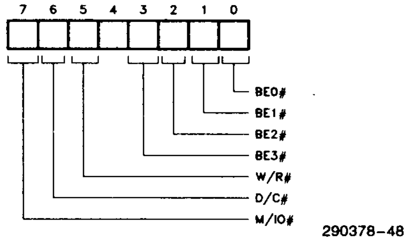
NOTE:
"x" indicates an undefined state at power up.

Register Description

The Parity Error Trap Registers are read only registers which hold the current cycle's address and status information. Should a parity error be signaled to the 82359 (through PER# or PERSTB#), these registers capture the current address and status, allowing the software error recovery routine to examine them for the location which caused the error.



Parity Error Address Register



Parity Error Status

Field Descriptions

Parity Error Address Registers: These bytes contain the address latched when a main memory parity error has been detected. The address latched is either the host or system address, depending on which port owns the memory when the error occurs. Normally, these registers are reloaded at the beginning of every memory cycle. When a parity error is detected, the registers contents are locked and are not allowed to be reloaded until each register has been read.

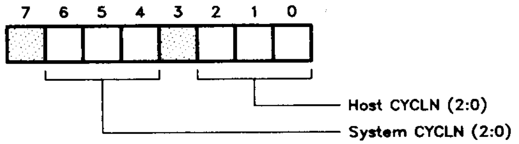
Parity Error Status Register: This byte contains the byte enables and the bus status which was present when the parity error occurred. This information is latched until the Parity Error Registers are read by the host. Specifically, the read order must be offset 28h, 29h, 2Ah, 2Bh, and finally 2Ch. A read from 2Ch automatically unlocks all Parity Error Registers.

CYCLE LENGTH REGISTERS

Register Name	Offset	Default Values	Access
Read Page Hit Cycle Length	30h	*111*111b	W/R
Read Page Miss Cycle Length	31h	*111*111b	W/R
Read Row Miss Cycle Length	32h	*111*111b	W/R
Write Page Hit Cycle Length	33h	*111*111b	W/R
Write Page Miss Cycle Length	34h	*111*111b	W/R
Write Row Miss Cycle Length	35h	*111*111b	W/R

Register Description

The Cycle Length Registers are used to relay to the PST the relative amount of time required by the 82359 to complete a deterministic host or system access.



290378-49

Field Descriptions

Host Cycle Length: This field represents a relative number of host PST clocks required for the current main memory cycle (the lead off bus cycle if a burst access). Any host or system access to main memory may result in one of six types of DRAM cycles. Each type of cycle has a 3-bit field associated with it which holds the relative amount of time required by the 82359 to complete the memory access. When the host or system bus cycle type has been decoded by the 82359, the corresponding cycle length feedback field is driven onto the CYCLN(2:0) pins. Typically, the PST will use this information to determine when to return RDY# or BRDY# to the host.

Bits 2:0 or 6:4	# of Clocks
0 0 0	n
0 0 1	n + 1
:	:
1 1 1	n + 7 (default)
n = some constant	

N typically represents the number of clocks in the fastest possible cycle, so 3 bits provide a "dynamic range" from fastest to slowest of 8 clocks. Note that the values in these registers have no impact on 82359 behavior, so a system design is free to modify the field interpretation as required.

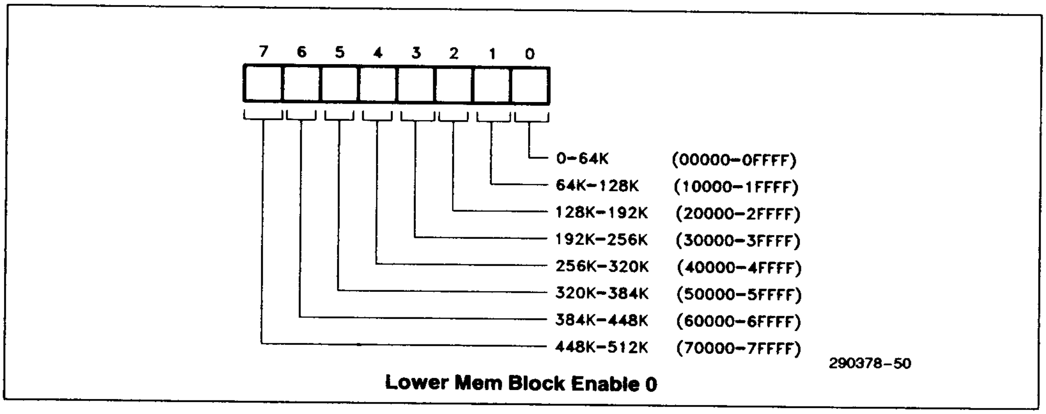
System Cycle Length: These register values are driven on to CYCLN(2:0) for system PST originated cycles in the same way the host cycle length is driven.

LOWER MEMORY BLOCK ENABLE REGISTERS

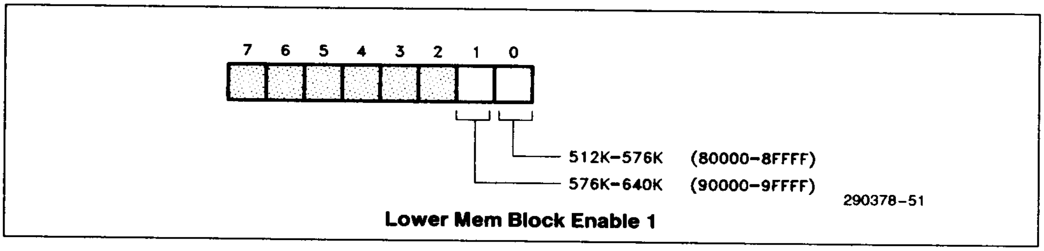
Register Name	Offset	Default Values	Access
Lower Mem Block Enable 0	40h	11111111b	W/R
Lower Mem Block Enable 1	41h	******11b	W/R

Register Description

Portions of main memory from 0K to 640 Kbytes can be enabled or disabled/remapped in 64K blocks through the use of these registers.



Lower Mem Block Enable 0



Lower Mem Block Enable 1

Field Descriptions

Block Enable Bits: Setting a bit to a "0" will disable the corresponding memory range. Setting a bit to a "1" enables the block.

Note that if data is stored in memory prior to disabling, the data remains intact such that if at a later time the memory is re-enabled, the data is still valid. This allows paged expansion memory boards to implement overlays if desired. Note that this behavior applies to all memory from 0-1M that can be toggled between the remap and linear states; i.e., data stored to memory in the linear state is maintained even if the memory is put into the remap state, *assuming of course, the memory is not actually remapped and enabled elsewhere.* (Note that although every block in the 0-64K range may be disabled, only the 512K-576K and 576K-640K blocks are actually remappable.)

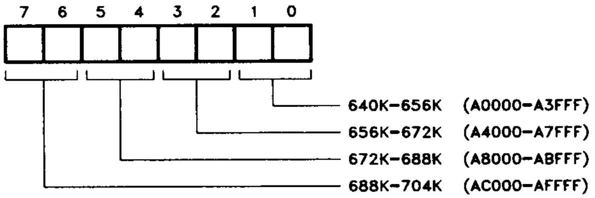
Bit	State
0	Disable
1	Enable

VIDEO RAM AREA BLOCK ENABLE REGISTERS

Register Name	Offset	Default Values	Access
VRAM BLOCK ENABLE0	42h	00000000b	W/R
VRAM BLOCK ENABLE1	43h	00000000b	W/R

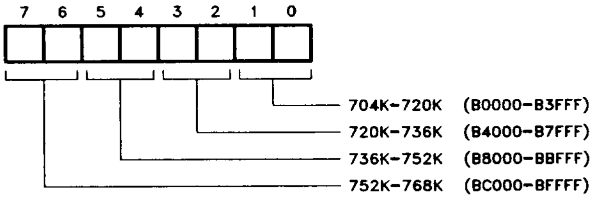
Register Description

Portions of main memory from 640K–768K bytes can be independently write/read enabled, write-only, read-only, or disabled/remapped in 16K blocks through the use of these bits.



290378–52

VRAM Block Enable0



290378–53

VRAM Block Enable1

Field Descriptions

Block Enable Bits: During the "copy state", appropriate sections of the main memory are programmed as write only. During the "shadow state", appropriate sections are programmed as read only. During the "remap state", appropriate sections are disabled, and in the "linear state", both writing and reading are enabled. The 2-bit code for each of the states follows:

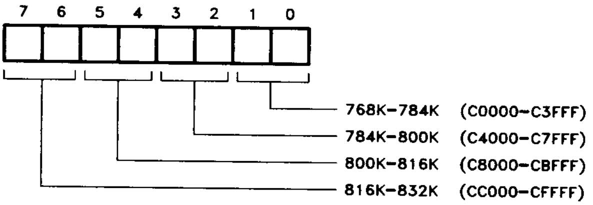
Bit Code	Memory Enable Function	Memory State
0 0	Disable Memory	REMAP (default)
0 1	Read Enable Only	SHADOW
1 0	Write Enable Only	COPY
1 1	Read/Write Enable	LINEAR

EXPANSION ROM AREA BLOCK ENABLE REGISTERS

Register Name	Offset	Default Values	Access
EX ROM BLOCK ENABLE0	44h	00000000b	W/R
EX ROM BLOCK ENABLE1	45h	00000000b	W/R

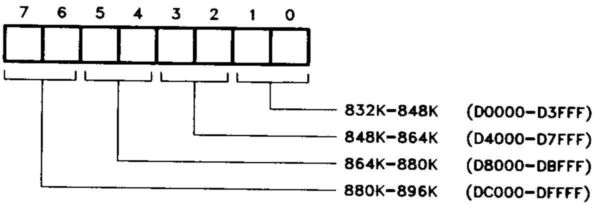
Register Description

Portions of main memory from 768K–896K bytes can be independently write/read enabled, write only, read-only, or disabled/remapped in 16K blocks through the use of these bits.



290378–54

EX ROM Block Enable0



290378–55

EX ROM Block Enable1

Field Descriptions

Expansion ROM Area Block Enable Bits: During the "copy state", appropriate sections of the main memory are programmed as write only. During the "shadow state", appropriate sections are programmed as read only. During the "remap state", appropriate sections are disabled, and in the "linear state", both writing and reading are enabled. The 2-bit code for each of the states follows:

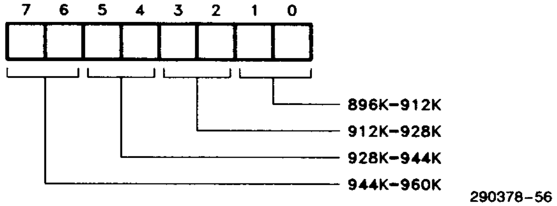
Bit Code	Memory Enable Function	Memory State
0 0	Disable Memory	REMAP (default)
0 1	Read Enable Only	SHADOW
1 0	Write Enable Only	COPY
1 1	Read/Write Enable	LINEAR

BIOS AREA BLOCK ENABLE REGISTERS

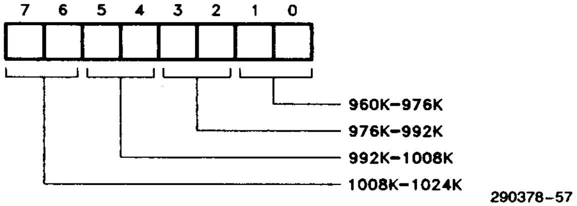
Register Name	Offset	Default Values	Access
BIOS BLOCK ENABLE0	46h	10101010b	W/R
BIOS BLOCK ENABLE1	47h	10101010b	W/R

Register Description

Portions of the main memory from 896k–1M byte can be independently write/read enabled, write only, read only, or disabled/remapped in 16k blocks through the use of these bits.



BIOS Block Enable0



BIOS Block Enable1

Field Descriptions

BIOS Area Block Enable Bits: During the "copy state", appropriate sections of the main memory are programmed as write only. During the "shadow state", appropriate sections are programmed as read only. During the "remap state", appropriate sections are disabled, and in the "linear state", both writing and reading are enabled. The 2-bit code for each of the states follows:

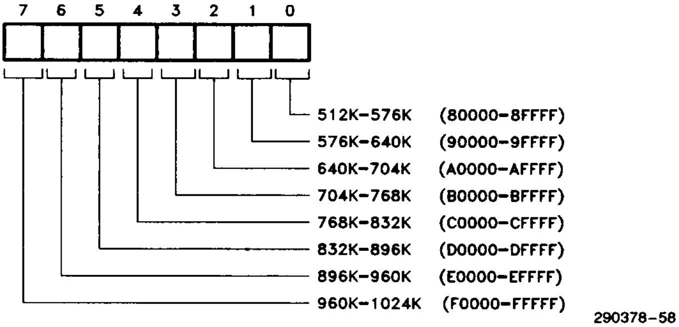
Bit Code	Memory Enable Function	Memory State
0 0	Disable Memory	REMAP (default)
0 1	Read Enable Only	SHADOW
1 0	Write Enable Only	COPY
1 1	Read/Write Enable	LINEAR

REMAP ENABLE REGISTER

Register Name	Offset	Default Values	Access
Remap Enable	4Eh	00111100b	W/R

Register Description

The user can select portions of memory in the 512K-1M range in 64K blocks to be remapped to other parts of memory through the use of these bits.



Field Descriptions

Remap Enable Bits: Setting a bit to a "0" will disable remapping of the corresponding block; setting a bit to "1" will remap the corresponding block to a new address pointed to by the Split Address Register. The portions of the memory to be remapped must initially be disabled through the use of the various block enable registers. (Note: the 82359 does not check to insure all memory remapped via this register is in the remap state. If this register attempts to remap memory not in the remap state, results are unpredictable.) All the 64K blocks that are enabled for remapping are assembled and remapped as a single contiguous block located at the address specified by the Split Address Register.

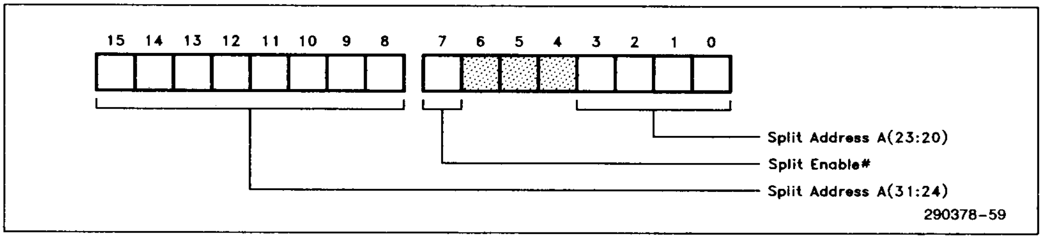
Bit	Remap State
0	Disable Remapping
1	Enable Remapping

SPLIT ADDRESS REGISTER

Register Name	Offset	Default Values	Access
Split Address Register (MSB, LSB)	83h, 84h	00000000 1***1111b	W/R

Register Description

The Split Address Register is used as the pointer to the address to which memory to be remapped is to reside. To remap a portion of main memory, it must first be disabled and placed in the remap state through its Block Enable Register.



Field Descriptions

Split Address: A31–A24 (register offset 83h), and A23–A20 (register offset 84h) specify the starting address for remappable portions of main memory. Through the use of these registers, memory specified as remappable through the Block Enable registers can be placed at any 1M boundary in the 4G address space. The value of the Split Address field has no meaning if memory remapping is not enabled by the ENSPLIT# bit of this register.

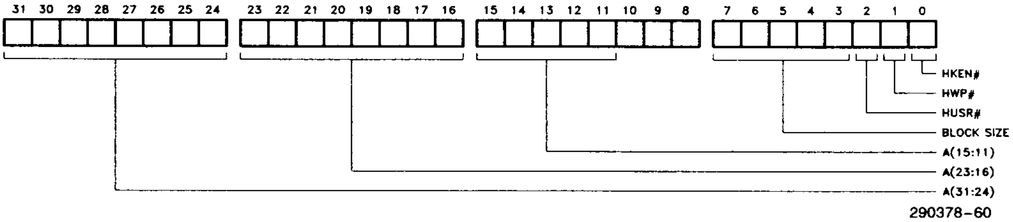
ENSPLIT#: This bit acts as a master remap enable bit. If this bit is a "0", then, based on the Remap Enable Register, memory is remapped starting on 1 Megabyte address boundaries as defined by programmed value of the Split Address field. If ENSPLIT# is programmed as a "1", all disabled memory remains disabled and no remapping takes place. It is the user's responsibility to not assign multiple memory resources to the same address range.

PROGRAMMABLE ATTRIBUTE MAP REGISTERS (PAM REGISTER)

Register Name	Offset	Default Value	Access
PAM0	53h:50h	00000000 00000000 00000*** 00001110b	W/R
PAM1	57h:54h	00000000 00000000 00000*** 00001110b	W/R
PAM2	5Bh:58h	00000000 00000000 00000*** 00001110b	W/R
PAM3	5Fh:5Ch	00000000 00000000 00000*** 00001110b	W/R

Register Description

The Programmable Attribute Map Registers (PAM Registers) provide four programmable address decode segments, each with a field of 3 attribute bits. These address regions are defined through a programmable base address and a block size for each of the segments. Should a host address lie within any block defined by a PAM, each of the three attribute bits of the register are reflected on the corresponding output pins (HKEN#, HWP#, HUSR#) of the 82359. Other than driving the output pins, the state of the PAM attributes has no effect on the internal workings of the 82359. Note that the 82359 will perform a write to a write protected area if asked to do so, and it is the external hardware's responsibility not to cause a write to a range marked as HWP# = 0.



290378-60

Also note that the starting address of a PAM block is a function of the block size (i.e., a block of size N may start on any Nk boundary.

Field Descriptions

Base Address A31–A11: This field specifies the base address for each the 4 segments. The validity of these bits is a function of the programmed block size, as shown in the table below.

Block Size: The Block Size field specifies the size of each segment. Along with the base address field, memory blocks can be decoded anywhere in the 4 gigabyte address space with a size between 2 kbytes and 4 gigabytes. The following table lists the various block sizes:

Bit 76543	Block Size	Valid Address Bits
00000	reserved	reserved
00001	2K	A(31:11) (default)
00010	4K	A(31:12)
00011	8K	A(31:13)
00100	16K	A(31:14)
00101	32K	A(31:15)
00110	64K	A(31:16)
00111	128K	A(31:17)
:	:	:
:	:	:
10010	256M	A(31:28)
10011	512M	A(31:29)
10100	1G	A(31:30)
10101	3G	A31
10110	4G	none
10111	reserved	reserved
:	:	:
11111	reserved	reserved

HKEN#: The default state of memory is cacheable (HKEN# = "0"). Writing a "1" to the HKEN# attribute bit causes the associated address range to be non-cacheable. Since PAM address ranges can overlap with each other, and also with ranges specified by other cacheable/non-cacheable mechanisms, the 82359 resolves any attribute conflicts according to the following rule: If any one mechanism marks a particular range as non-cacheable, then that range is non-cacheable. HKEN# will remain de-asserted regardless of the PAM register settings if the host cache is disabled via the Cache Control Register (offset 85h), bit 0.

HWP#: The default state of memory is non-write protected (HWP = "1"). Writing a "0" to the HWP# bit causes the associated range to be write protected. In case of an attribute conflict due to range overlap, the rule is: If any one PAM marks a range as write protected, then the range is write protected.

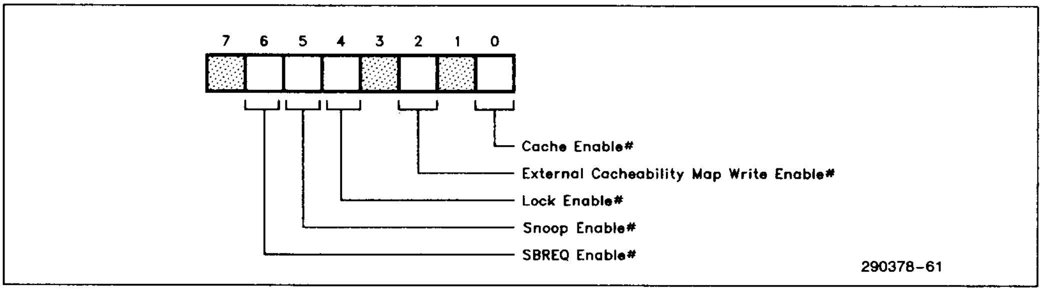
HUSR#: The default state of memory is non-user attributed (HUSR# = "1"). Writing a "0" to the HUSR# bit causes the range to be user-attributed. In case of an attribute conflict due to range overlap, if any one PAM marks a range as user-attributed, the HUSR# output will be asserted.

CACHE CONTROL REGISTER (CCR)

Register Name	Offset	Default Values	Access
CCR	85h	*100*1*1b	W/R

Register Description

The CCR register is used to control various host cache parameters.



Field Descriptions

Cache Enable#: If this bit is a "0", the cache, and thus the HKEN# function will be enabled. A "1" will force HKEN# inactive, regardless of the PAM settings.

External Cacheability Map Write Enable#: If the PCDOVERRIDE bit (Mode Register B, bit 1) is set to "1", this bit is directly reflected onto the dual function pin HBURST#/CCRB2, and acts as a control for writing to an external cacheability map SRAM. Used in this setting, writing a "0" to this bit will cause the pin to go low, and allow a write operation to occur to the external cacheable address map SRAM device implemented in some designs.

If the PCDOVERRIDE bit is "0", the External Cacheability Map Write Enable bit simply becomes a RD/WR bit with no other 82359 functionality.

Lock Enable#: This bit is directly reflected on the LOCKEN# output pin. When this bit is set to a "1", LOCKEN# causes the CPU LOCK# signal to be gated (externally) from reaching the cache controller. If this bit is a "0", LOCKEN# allows the LOCK# signal from the CPU to reach the cache.

Snoop Enable#: When set to "0" this bit will allow the 82359 to initiate snoop cycles. If set to a "1" snooping by the 82359 is disabled. This bit is provided to ease the implementation of designs using write back cache which must snoop both read and write cycles. Default value after reset is "0".

SBREQ Enable#: This bit allows the host CPU to participate in arbitrating for system bus ownership. When this bit is set to a "0", generation of system bus request SBREQ is enabled. (A system bus request is generated when the 82359 is requesting ownership of the system bus on behalf of the CPU and the system bus is currently busy.)

When this bit is a "1", system bus requests on behalf of the CPU are disabled. The CPU may not arbitrate for system bus ownership, but gets control only when no other system bus activity is occurring. The actual system bus request can be enabled/disabled either internally or externally, depending on the state of Mode Register B, bit 2.

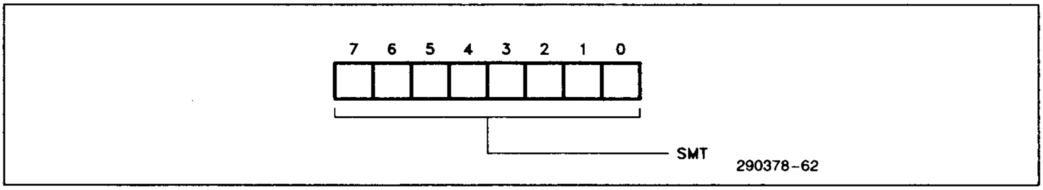
SYSTEM MEMORY THROTTLE (SMT)

Register Name	Offset	Default Values	Access
SMT	8Bh	11111111b	W/R

Register Description

When a system master owns the memory, the SMT defines the number of 50 ns periods (OSC/2) allowed to occur before sampling/honoring a host request. The SMT, in conjunction with the Host Memory Throttle (HMT), controls the time-sharing of main memory ownership between host and system masters.

The SMT Register default value 11111111b indicates an active state where system master access time to main memory is controlled.



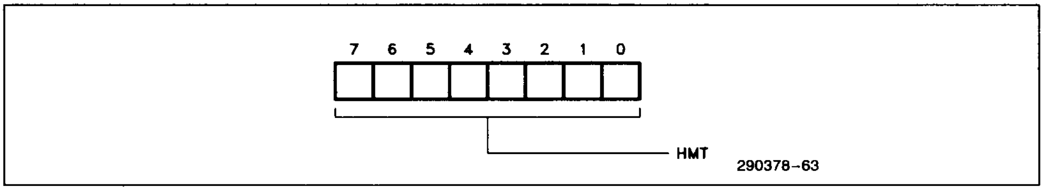
HOST MEMORY THROTTLE (HMT)

Register Name	Offset	Default Values	Access
HMT	8Ch	00000000b	W/R

Register Description

The HMT, in conjunction with the SMT, controls time-sharing of the main memory between the host and system ports. Once the host gains memory ownership, the HMT guarantees that the host can keep ownership for the number of 50 ns periods programmed into the HMT register. This is accomplished by the 82359 holding back any HMREQ signals from the host PST. Once the HMT expires, all pending or future HMREQs will be sent to the host bus as soon as the 82359 receives them.

The actual time increment used by the HMT is based on the 40 MHz OSC input divided by 2, resulting in a 50 ns period; i.e., actual length in time is the HMT count value times 50 ns.

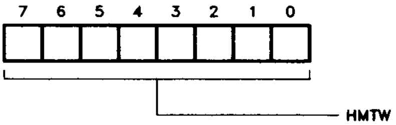


HOST MEMORY THROTTLE WATCHDOG (HMTW)

Register Name	Offset	Default Values	Access
HMTW	8Dh	11111111b	W/R

Register Description

The HMTW and HMT work closely together to control host utilization of main memory. The HMTW begins its countdown once the host stops utilizing main memory while the HMT has not expired. If the host has been given the memory for an amount of time specified by the HMT, but then fails to use it for an interval specified by the HMTW and a system device request ownership, the HMT will be overridden and memory ownership given to the requesting system device. HMTW essentially defines a "use it or lose it" mechanism to prevent an idling CPU from wasting memory bandwidth.



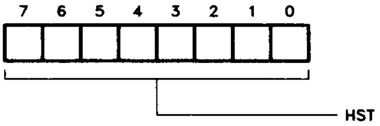
290378-64

HOST SYSTEM THROTTLE (HST)

Register Name	Offset	Default Values	Access
HST	8Eh	00000000b	W/R

Register Description

The HST guarantees the host CPU a minimum amount of system bus bandwidth. The HST takes effect upon the host obtaining system bus ownership. Once the system bus ownership is acquired by the host, subsequent requests for system bus ownership by the external system bus arbiter (EISA ISP) through SHOLD will not receive a SHLDA until the HST times out.



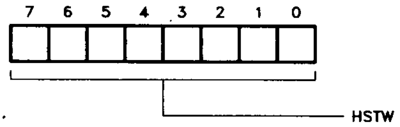
290378-65

HOST SYSTEM THROTTLE WATCHDOG (HSTW)

Register Name	Offset	Default Values	Access
HSTW	8Fh	11111111b	W/R

Register Description

The HSTW prevents the host from wasting system bus bandwidth. Once the host has obtained ownership of the system bus and the HST has begun to countdown (if enabled), the 82359 will not honor any subsequent system request from system masters until the HST times out. If, however, after the subsequent system request occurs, and the Host CPU fails to make use of the system bus in the interval specified by the HSTW (HSTW value × 50 ns), the HST is overridden and the system request is processed immediately. This "use it or lose it" mechanism prevents an idling CPU, or a CPU whose requirements have already been met (single read or write), from wasting the remaining portion of its HMT guaranteed time and thus waste system bus bandwidth.



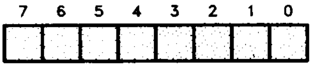
290378-66

RAM ENABLE REGISTER

Register Name	Offset	Default Values	Access
RAM Enable	90h	********b	R

Register Description

This register acts as a global Resource Allocation Monitor (RAM) enable. When this register is read, the RAM registers ETC, HMR, SMR, HMO, SBO, HRS, and MOT become active and their contents will increment when the appropriate conditions exist for each. The contents of the RAM Enable Register is a "don't care" since it is the access to the RAM Enable Register which enables the RAM registers, not the RAM Enable Register's value.



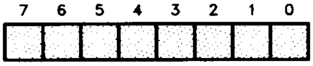
290378-67

RAM DISABLE REGISTER

Register Name	Offset	Default Values	Access
RAM Disabled	91h	********b	R

Register Description

The RAM Disable Register acts as a global Resource Allocation Monitor (RAM) disable. When this register is read, the RAM registers ETC, HMR, SMR, HMO, SBO, HRS, and MOT become disabled and their contents become frozen. At this point, the RAM Registers will no longer increment. The content of this register is a "don't care" since it is the access to the RAM Enable Register which enables the RAM registers, not the register's value.



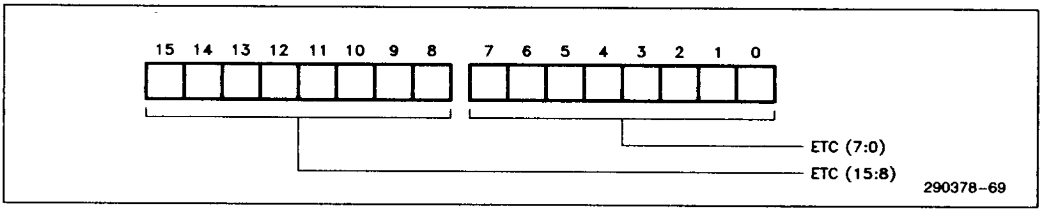
290378-68

ELAPSED TIME COUNTER (ETC)

Register Name	Offset	Default Values	Access
ETC	93h,92h	******** ********b	R

Register Description

The ETC (Elapsed Time Counter) is a read only register which provides information as to the number of cycles (based on the 20 MHz OSC/2 timebase) which have passed since the last reading of the RAM registers. The ETC consists of two 8-bit registers which base their value off of a 16-bit counter. This register is enabled as long as the RAM Count Enable bit is set to "1" and the 16-bit counter has not yet reached its maximum of FFFFh. Should the counter reach its maximum FFFFh value, the register contents are frozen until the RAM registers are read by software. Once the register is read, it resets itself to 0h.



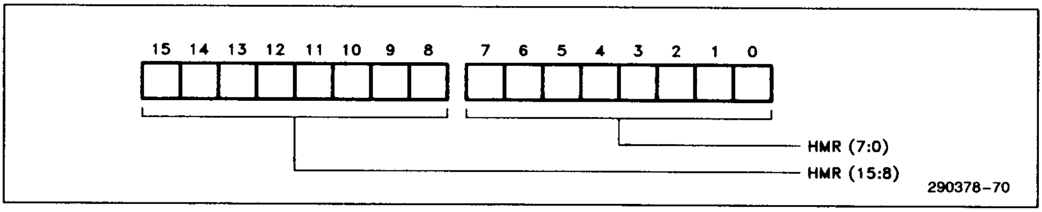
290378-69

HOST MEMORY REQUEST TO OWNERSHIP (HMR)

Register Name	Offset	Default Values	Access
HMR	95h,94h	******** ********b	R

Register Description

The HMR consists of two 8-bit registers which derive their value from a 16-bit counter based on OSC/2 (20 MHz). This counter is incremented when HAS# is asserted, the host address is for main memory, and HMACK is asserted (the conditions which occur when the host does not own memory but is requesting ownership). When any of the above factors is not true, the counter is halted but retains its current value. Like the other RAM registers, this counter is frozen once it reaches its maximum value of FFFFh and is reset when read.



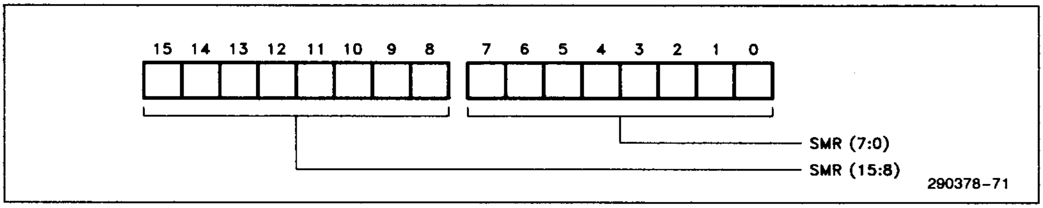
290378-70

SYSTEM MEMORY REQUEST TO OWNERSHIP (SMR)

Register Name	Offset	Default Values	Access
SMR	97h,96h	******** ********b	R

Register Description

The SMR consists of two 8-bit registers whose value is obtained from the 16-bit SMR counter. This counter gets its timebase from the EISA OSC/2. The counter is enabled for system PST masters when SAS# and DRAMCS# are asserted, and SMACK is high. For EISA/ISA masters the counter is enabled during the period defined by the start of a cycle (START# for EISA, MRDC#/MWTC# for ISA) with DRAMCS# asserted to the assertion of HMACK or the end of the current REFRESH cycle. This counter is frozen once it reaches its maximum value of FFFFh and is reset when read.

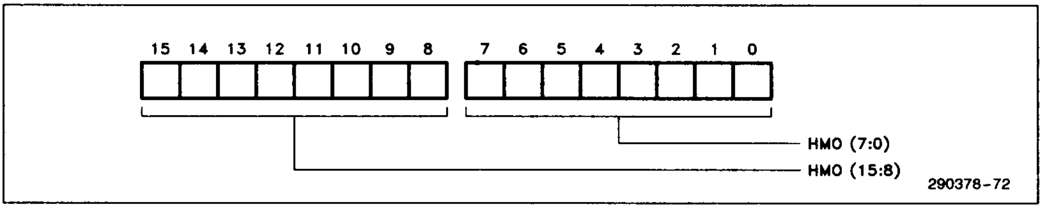


HOST MEMORY OWNERSHIP (HMO)

Register Name	Offset	Default Values	Access
HMO	99h,98h	******** ********b	R

Register Description

The HMO Register is obtained from the 16-bit HMO counter. This counter gets its timebase from the OSC/2 (20 MHz). The counter is enabled when HMACK and SBREQ are low, signifying the host owns main memory and is not waiting for the system bus. It should be noted that this counter is enabled even when the host owns main memory but is not currently running cycles (cache hits for example). Like the other RAM registers, this counter is frozen once it reaches its maximum value of FFFFh and is reset when read.

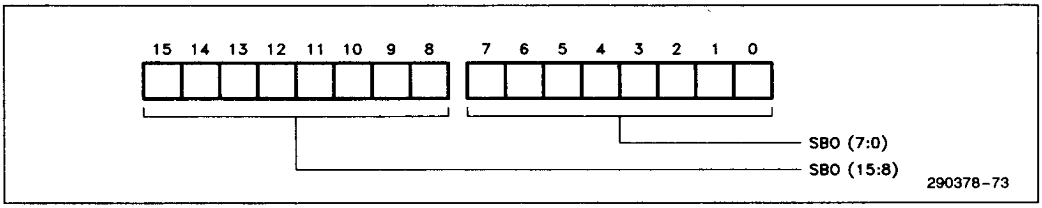


SYSTEM BUS OWNERSHIP (SBO)

Register Name	Offset	Default Values	Access
SBO	9Bh,9Ah	******** ********b	R

Register Description

The SBO Register consists of two 8-bit registers which obtain their value from the 16-bit SBO counter. This counter gets its timebase from OSC/2. The counter is enabled when SHLDA is asserted, signifying a system bus master or DMA owns the system bus. This counter is frozen once it reaches its maximum value of FFFFh and is reset to 0 when read.



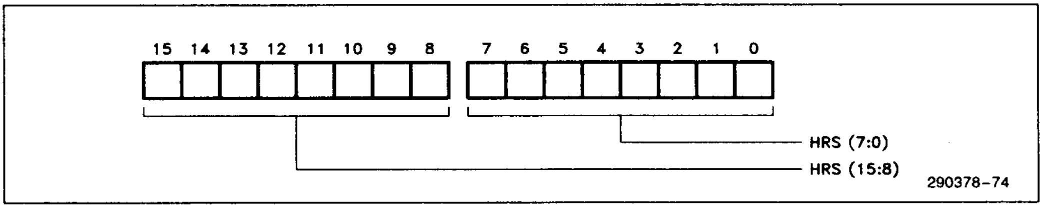
290378-73

HOST REQUEST OF SYSTEM BUS (HRS)

Register Name	Offset	Default Values	Access
HRS	9Dh,9Ch	******** ********b	R

Register Description

The HRS Register consists of two 8-bit registers which obtain their value from the 16-bit HRS counter. This counter gets its timebase from OSC/2 (20 MHz). The counter is enabled when SBREQ and SHLDA are asserted, indicating the host is requesting use of the system bus, and it is frozen when it reaches it maximum of FFFFh. The counter is frozen when it reaches its maximum value of FFFFh and is reset when read.



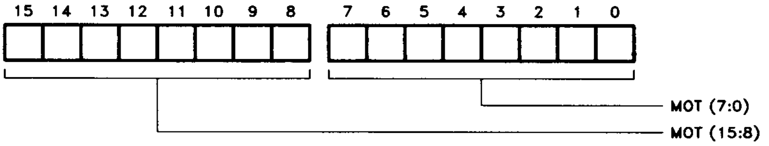
290378-74

MEMORY OWNERSHIP TRANSFER (MOT)

Register Name	Offset	Default Values	Access
MOT	9Fh,9Eh	******** ********b	R

Register Description

The MOT Register uses a 16-bit counter to count the number of times memory changes ownership. The counter is incremented every time memory ownership is transferred from host to system, system to host, or an internal refresh cycle is serviced while in Decoupled Refresh Mode. (Coupled refresh requests are system bus cycles to the 82359.) This register is frozen when it reaches its maximum value of FFFFh and is reset to 0 when read.



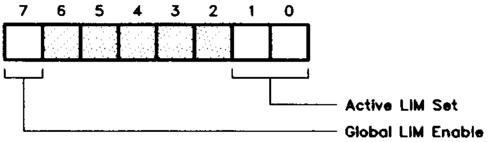
290378-75

LIM CONTROL REGISTER

Register Name	Offset	Default Values	Access
LIM Control	00h	0*****00b	W/R

Register Description

The LIM Control Register is used for determining the active LIM set as well as globally enabling LIM translations.



290378-76

Field Descriptions

Active LIM Set: This field must be set to "00" to select register set 0. (The 82359 supports only register set 0.)

Global LIM Enable: A "1" in this bit enables the LIM page translation capability. Should this bit be cleared to "0", LIM is globally disabled, regardless of the setting of any LIM Page Translation Registers.

LIM PAGE TRANSLATION REGISTERS

Register Name	Offset	Default Values	Access
LIM Page 0	81h-80h	0*****00 00000000b	W/R
LIM Page 1	83h-82h	0*****00 00000000b	W/R
LIM Page 2	85h-84h	0*****00 00000000b	W/R
LIM Page 3	87h-86h	0*****00 00000000b	W/R
LIM Page 4	89h-88h	0*****00 00000000b	W/R
LIM Page 5	8Bh-8Ah	0*****00 00000000b	W/R
LIM Page 6	8Dh-8Ch	0*****00 00000000b	W/R
LIM Page 7	8Fh-8Eh	0*****00 00000000b	W/R

Register Description

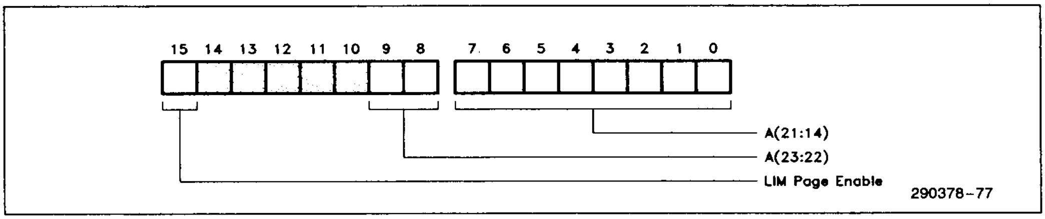
The 82359 supports eight LIM pages, each 16 Kbytes long which are physically located in the 768K-896K address range. Through the use of these LIM registers, the user can translate the physical address, mapped to a specific 16K LIM page, to any other 16K block in range 0M-16M and under control of the 82359.

Each LIM Page Translation Register contains a relocation address for the LIM page, and an enable bit. It is important to note that LIM remapping can only occur to main memory controlled by the 82359. Also, the 82359 does not actively enforce this rule. Rather, the Expanded Memory Manager (EMM) is responsible for not allowing an enabled translation to go outside the 82359's memory range.

Each of the LIM registers is assigned to a 16K blocK of main memory in the 0M-1M address range. The following table lists each of the LIM registers and its assigned address space.

LIM Page 0	768K-784K	(C0000h-C3FFFh)
LIM Page 1	784K-800K	(C4000h-C7FFFh)
LIM Page 2	800K-816K	(C8000h-CBFFFh)
LIM Page 3	816K-832K	(CC000h-CFFFFh)
LIM Page 4	832K-848K	(D0000h-D3FFFh)
LIM Page 5	848K-864K	(D4000h-D7FFFh)
LIM Page 6	864K-880K	(D8000h-DBFFFh)
LIM Page 7	880K-896K	(DC000h-DFFFFh)

In the case of a 4-way dword interleaved memory row implemented with 4M DRAMs (address depth), the 82359 does not translate address bit A14. This implies that the EMM should only load page registers 0,2,4, and 6 with addresses corresponding to the lower half of aligned 32K blocks, and page registers 1,3,5, and 7 with addresses corresponding to the upper half of aligned 32K blocks.



Field Descriptions

Translation Address A23-A14: This field specifies the translated base address for a memory access within the LIM memory page corresponding to this specific register. Address bits A16-A14 select one of the eight LIM registers, and A23-A17 validate the selection of the LIM function (i.e., the CPU address must be in the C0000h to DFFFFh range).

LIM Page Enable: A "1" enables this page register to remap the corresponding memory page according to the translation address. If this bit is a "0", the page translation register has no effect.

14.0 DETAILED PIN DESCRIPTIONS

The following section gives a detailed description of each pin of the 82359. The ordering of the descriptions is broken down into the five functional blocks: (1) Host Port Interface; (2) System Port Interface and EISA Interface; (3) Memory Interface; (4) 82353 Data Path Control Interface, and; (5) Miscellaneous Decodes and Control.

14.1 Host Port Interface

<table>
<tr><th>Symbol</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>HAS#</td><td>I</td><td><b>HOST ADDRESS STROBE:</b> HAS# is an input from the host bus. HAS# asserted signals to the 82359 that the host is currently running a cycle or is requesting the start of a cycle (actual cycle start is dependent upon HMACK as well). The falling edge of HAS# indicates that the host bus address and status are valid and causes the 82359 to transparently latch HA(31:2), HBE(3:0)#, HM/IO#, HW/R#, HD/C#, HLOCK#, and PCD. HAS# rising edge indicates the end of the current host cycle.
The falling edge of HAS# does not necessarily imply that the cycle will be run by the 82359 immediately. The 82359 will run the cycle immediately upon seeing the falling edge of HAS# only if HMACK is de-asserted. If HMACK happens to be asserted during the falling edge of HAS#, the 82359 treats it as a cycle request and the actual start of the cycle is held off until HMACK becomes de-asserted. HAS# and HMACK must never be asserted on the same clock edge.
An 82359 cycle, initiated by HAS#, may be aborted by de-asserting ST#. Such is the case when implementing a parallel cache to support the host processor.</td></tr>
<tr><td>HA(31:0),
HBE(3:0)#</td><td>I/O</td><td><b>HOST ADDRESS:</b> The HA(31:2) and HBE(3:0)# signals provide the 82359 with its 32-bit host address. These address signals are inputs for host initiated cycles and are latched on the falling edge of HAS#. Decodes such as HKEN#, HUSR#, HWP#, HBURST#, and CYCLN(2:0) are based on the latched address, such that as long as the latch is open (HAS# de-asserted), decodes combinatorially follow address. Once latch is closed (occurring on the falling edge of HAS#), decodes remain stable until HAS# is de-asserted, causing the latch to re-open.
The HA(31:2) address lines become outputs when the 82359 is transmitting a snoop address to the host cache. The snoop address is valid during the asserted time of SNUPACK#. The entire HA(31:2) bus is driven for snoops, which accommodates a cache line size as small as 4 bytes. Refer to snooping for more details.</td></tr>
<tr><td>HM/IO#,
HW/R#,
HD/C#</td><td>I</td><td><b>HOST CYCLE DEFINITION:</b> HM/IO#, HW/R#, and HD/C# are inputs to the 82359 from the host bus. These three signals define the current host bus cycle depending on the type of processor. (The type of host processor may be set in Mode Register A, bit 1.) All three follow the timing of the host address bus.
HM/IO#, HD/C#, and HW/R# decode to the following cycle definitions:
<table>
<tr><th>HMIO#</th><th>HD/C#</th><th>HW/R#</th><th>Bus Cycle Type</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Interrupt Acknowledge</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>i486 Special Cycle (386 rsvd)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>I/O Data Read</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>I/O Data Write</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Memory Code Read</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>i486 Rsvd (386 Special Cycle)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Memory Data Read</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Memory Data Write</td></tr>
</table></td></tr>
</table>

14.1 Host Port Interface (Continued)

<table>
<tr><th>Symbol</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>HM/IO#,
HW/R#,
HD/C#
(Continued)</td><td>I</td><td>All cycles except memory code reads, memory data reads, and memory data writes that are decoded as main memory accesses are forwarded to the system bus.
The exact function of special cycles is indicated by the byte enables.
<table>
<tr><th>BE3#</th><th>BE2#</th><th>BE1#</th><th>BE0#</th><th>Function</th><th>Defined For</th></tr>
<tr><td>1</td><td>1</td><td>1</td><td>0</td><td>Shutdown</td><td>386/486</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>Flush</td><td>486</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>Halt</td><td>386/486</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>Write Back</td><td>486</td></tr>
</table>
In a cache flush cycle the DEN# pin will be activated to indicate a cache flush and this signal can be used to flush all external caches. (Whether or not DEN# is generated on behalf of a special cache flush cycle is programmed via bit 3 of Mode Register A.) All special cycles are treated like I/O cycles in that the 82359 forwards them to the system port. It is the responsibility of the system bus controller to terminate these cycles. It is also the responsibility of the system to insure that these special cycles, which otherwise look like memory (386) or I/O (i486) code writes, are not broadcast in such a way as to accidentally corrupt any system memory or I/O that happens to be selected by the address. Since it's likely the system will not broadcast special flush cycles, in this case, DEN# activation follows SAS# timing rather than CMD# timing.
If the host and the system have the same status format (both i486 or 386, as determined by Mode Register A, bits 1 and 2), host status is simply propagated through the 82359 unaltered during host to system cycles. If the two bits have different values then the appropriate format conversion is made by the 82359; i.e., a memory code write is converted to an I/O code write, or vice-versa. The byte enables are propagated through unaltered. Note that if the host is i486-like, and the system 386-like, the system must still be able to terminate the i486 special cycles flush and write-back.</td></tr>
<tr><td>HARDY</td><td>O</td><td><b>HOST ASYNCHRONOUS READY:</b> HARDY is an asynchronous ready indicator from the 82359 to the host bus. The 82359 de-asserts HARDY when the host address and status decode to be a system bus cycle (a non-deterministic cycle). HARDY remains asserted during a deterministic cycle. The 82359 will re-assert HARDY when the non-deterministic cycle is complete.
The HARDY deactivation point is sampled synchronously by the host PST at a point that allows for sufficient address and status decode time. If the host PST samples HARDY de-asserted, the CYCLN is ignored and the PST waits for HARDY's rising edge to indicate the cycle is complete. When the PST sees that HARDY is re-asserted, it will return "READY" back to the CPU.</td></tr>
<tr><td>HMREQ</td><td>O</td><td><b>HOST MEMORY REQUEST:</b> HMREQ is asserted by the 82359 to the host bus to indicate that the 82359 would like ownership of main memory. The 82359 will assert HMREQ on behalf of a system-side device which has asked the 82359 for memory ownership or for a memory refresh condition.</td></tr>
<tr><td>HMACK</td><td>I</td><td><b>HOST MEMORY ACKNOWLEDGE:</b> HMACK is an input from the host PST which indicates that the HMREQ has been honored and that the host master has given memory ownership to the 82359. Upon wake-up, the memory is owned by the host as indicated by HMACK de-asserted.
HMACK falling may also signal the actual start of a host cycle to the 82359. If the falling edge of HAS# occurs while the 82359 owns main memory (HMACK asserted), the host cycle is held-off until the 82359 relinquishes memory ownership back to host as indicated by the falling edge of HMACK. It is at this time that the 82359 begins the host cycle.
<b>NOTE:</b>
The PST must never assert HAS# and HMACK on the same clock edge.</td></tr>
</table>

1

14.1 Host Port Interface (Continued)

<table>
<tr><th>Symbol</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>HBURST#/
CCRB2</td><td>O</td><td>

HOST BURST/CCRB2: The HBURST#/CCRB2 is an output of the 82359 and may take on one of two functions depending on the setting of the PCDOVERRIDE bit (Mode Register B, bit 1). If the PCDOVERRIDE bit is reset, this pin takes on the function HBURST; PCDOVERRIDE set causes this pin to function as CCRB2 (Cache Control Register, bit 2).

In a 486 system, the HBURST# function is used. As HBURST#, this signal indicates that the device or memory at the address provided to the 82359 by the host CPU is capable of a burst type transfer. HBURST# is generated only when conditions are detected that guarantee that the host will do a "full" burst; i.e., a burst equal to the programmed cache line size. Cycles that will be converted to linefills or code prefetch accesses are burstable, and will be indicated to the host interface through HBURST#. The PST interface will use this input and the IF(1:0) to decide whether to respond with RDY# or BRDY#.

If the code prefetch line size is smaller than that of the cache line size, or if the host CPU does not burst code prefetches, then Mode Register A, bit 4 should be cleared.

If the current address is less than 1 Meg and if the address is not in the DRAM's main memory space, then HBURST# will remain inactive irrespective of cacheability. The reason for restricting bursts is primarily so that slow narrow peripherals in this space (such as 8-bit ROMs) do not cause a system burst, and in so doing tie up the system bus for long periods. This restriction also applies to the high ROM mapping if it is not in the DRAM's memory space. Once ROMs are shadowed, they become part of DRAM space and are then burstable.

Bursting of addresses greater than or equal to 1M that are not in the DRAM's main memory space are controlled by the setting of the RBRST bit of Mode Register B. Below is a table summarizing the restrictions on the HBURST# signal.

<table>
<tr><th>Addr Range</th><th>DRAM</th><th>H/ROM</th><th>RBRST</th><th>Burst Enable</th></tr>
<tr><td>< 1M</td><td>0</td><td>X</td><td>X</td><td>0</td></tr>
<tr><td>< 1M</td><td>1</td><td>X</td><td>X</td><td>1</td></tr>
<tr><td>> 1M</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>> 1M</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>> 1M</td><td>0</td><td>1</td><td>X</td><td>0</td></tr>
<tr><td>> 1M</td><td>1</td><td>X</td><td>X</td><td>1</td></tr>
</table>

Upon reset, the cache wakes up disabled. Hence, HBURST# is negated until either the cache is enabled or Burst Code Pre-fetches is enabled. (Burst Code Pre-fetches wakes up disabled.) In a system that utilizes the PCDOVERRIDE function (386/385), the PCDOVERRIDE bit must be set prior to enabling the cache. Also, if the system using PCDOVERRIDE also uses the HBURST# pin as CCRB2, then the pins reset state that must be assumed is "1". If this "1" must be maintained after the PCDOVERRIDE bit is set (which causes the pin to toggle from its HBURST# function to CCRB2), then bit 2 of the CCR must be initialized accordingly **before** the PCDOVERRIDE bit is set. In an EISA 386/385 system, CCRB2 is used to enable writing into a 386 local bus cacheable map translation RAM.

Note that even in PCDOVERRIDE mode, PCD is ignored until the cache is enabled. Thus in this mode, the host PST must monitor HKEN# to know whether or not pulling PCD low will actually result in an 82359 burst.

For example if IF(1:0) = "00" (4-way) and the cache line size is 128 bits, there will be one MDS# pulse and the PST has the option to indicate to the CPU that it still could burst (independent of the HBURST# status). Based on CPU (BRDY#/BLAST#) response the PST will run the number of cycles it requires to the 82353 data path to get the data.

</td></tr>
</table>

14.1 Host Port Interface (Continued)

Symbol	Type	Name and Function
HBURST#/ CCRB2 (Continued)	O	Suppose IF(1:0) indicates a 2-way dword interleave with the same cache line size then If HBURST# is not true There will be one MDS# pulse to the DRAMs and the PST could potentially burst with two 2-dwords to the CPU (an interrupted burst). If HBURST# is true There will be two MDS# pulses to the DRAMs and the PST will burst with four dwords to the CPU. (Refer to the MDS# description for the MDS# pulse generation with different cache line sizes.) If the current cycle is a burst and the address is not in main memory, the cycle is forwarded to the system bus. The system bus port will indicate to the bus controller that the current cycle is a burst by asserting SBURST# as an output.
PCD	I	**PAGE CACHE DISABLE:** PCD is an input from the host to the 82359 typically driven from the i486 PCD output. This is an active high signal with timings similar to address. The PCD input is used by the 82359 in determining the cacheability of addresses. If sampled active, HKEN# is not asserted. In a 386/395 system, an external I/O port could be implemented to statically switch this pin. This would allow emulation of the 486 ability to "freeze" cache contents. In a 386/385 system, this pin should be driven by the 385 MISS# output. (Note MISS# is an active low cacheable indicator, while PCD is defined as an active high non-cacheable indicator, so the polarity works with no modification.) When the 385 asserts MISS# and the cycle is a read, a cache line fill cycle is already guaranteed (assuming cache is enabled). In case of a 386/385, MISS# alone should determine cacheability and burstability, so the PCD override bit (bit 1 of Mode Register B) should be set = 1.
HLOCK#	I	**HOST LOCK:** HLOCK# is an input to the 82359 from the host bus. When the HLOCK# pin is asserted and the bit LOCKEN# = 0 of the CCR register, the current bus cycle is a locked cycle. When the current cycle is locked, irrespective of the cycle type, the system bus will be arbitrated for and system bus ownership obtained before the 82359 runs the host cycle. System bus ownership will remain in the host's possession until the complete lock sequence is done. Locked cycles always run as non-deterministic cycles. If LOCKEN# = 1, the HLOCK# input is ignored.
SNUPRQ	O	**SNOOP REQUEST:** The 82359 asserts this signal to the host interface when it has a pending cache invalidation cycle. The 82359 de-asserts SNUPRQ from the falling edge of SNUPACK#.
SNUPACK	I	**SNOOP ACKNOWLEDGE:** SNUPACK# is an input from the host PST to the 82359 to acknowledge the 82359's request to run a snoop cycle. The snoop address is driven onto the host bus from the falling edge of SNUPACK#. The rising edge of SNUPACK# occurs when the snoop cycle has been complete, and causes the 82359 to tri-state the host address. Typically it is the host PST which monitors the SNUPRQ and drives SNUPACK# onto the host bus at the appropriate time.
HUSR#, HKEN#, HWP#	O	**HOST PROGRAMMABLE ATTRIBUTES:** The 82359 provides three attribute bits: HUSR#, HKEN#, and HWP#. The user can select desired address decode range(s) by programming the Programmable Attribute Map (PAM) register set (see register descriptions). Upon reset, the attribute bits will be de-asserted. The attributes are combinatorially decoded off a valid address in the host address latch. The attribute bits will reflect the setting of one of the Programmable Attribute Map registers if the host address falls within one of the registers programmed address range. Should the current host address not be included in the address range of any of the PAMs, HWP# and HUSR# will remain deasserted, and HKEN# is the result of the setting of the Block Cache Enable bits exclusively. **HOST USER ATTRIBUTE (HUSR#):** The HUSR# pin is reserved for the system designer to use as needed.

1

14.1 Host Port Interface (Continued)

Symbol	Type	Name and Function
HUSR#, HKEN#, HWP# (Continued)	O	**HOST CACHE ENABLE ATTRIBUTE (HKEN#):** This pin is used by the host CPU/cache in order to determine if the current cycle is cacheable. All memory is assumed to be cacheable until portions of it are defined non-cacheable through the use of the Programmable Attribute Map, Block Cache Enable Register, or the Cache Control Register. Upon reset, the cache is disabled (HKEN# = 1) until the appropriate Cache Enable bit is programmed through the Cache Control Register. HKEN# may be asserted only for memory data reads, memory code reads, and memory data writes, since only these three cycle types can be cached. (A memory code write is actually the 386 special cycle indicator.) In an 80386/385 system which establishes cacheable attributes on the local CPU bus, the PCDOVERRIDE bit (Mode Register B, bit 1) must be set to "1". (Since the CCR cache enable bit wakes up reflecting the cache as disabled, HKEN# will not be asserted until the cache is enabled. This provides an opportunity in a 386/385 system to set the PCDOVERRIDE bit, which otherwise wakes up = 0, before enabling the cache.) In a 386/385 system, HKEN# should be combined externally with DEN# and SW/R# to create the 385 FLUSH function: $(385\ \text{FLUSH} = 1) = \overline{\text{HKEN\#}} + (\text{HKEN\#}) \cdot (\text{DEN\#}) \cdot (\text{SW/R\#})$ (Note that this function requires additional external logic to force FLUSH to 0 around the falling edge of 385 RESET to prevent accidental invocation of 385 test modes.) **HOST WRITE PROTECT ATTRIBUTE (HWP#):** This pin is used by cache controllers with write protection capabilities, such as the 82485 or 82395, to identify write protected blocks of memory. All memory is assumed to be writable until portions of it are defined as write protected through the use of the Programmable Attribute Map. This decode does not write protect main memory. The HWP#'s only impact is to manipulate the HWP# output and does not in itself prevent the 82359 from executing a write cycle to memory.
ST#	I	**START (Continue) HOST CYCLE:** ST# is an input to the 82359 from the host bus. If ST# sampled inactive after a DRAM cycle has started, the cycle will be aborted. Another way to look at the ST# signal is to consider it a "continue" signal, causing the 82359 to continue its current cycle as long as ST# remains asserted. ST# is typically used in a parallel cache configuration, allowing a DRAM cycle and cache look-up to start in parallel. The cache will negate ST# if the cycle can be serviced from the cache. For example, ST# may be derived from the START# signal of the 82485 Cache Controller. ST# should be tied low if not used. **NOTE:** The ST# signal should be used to abort DRAM cycles only when the HP1 register (offset 10h) is programmed to a value of 0 through 4 (inclusive). Correct functionality of the ST# input is not guaranteed when HP1 is programmed to a value of 5 or 6. The 82359 explicitly treats ST# as a don't care in all but CPU memory reads and is ignored in writes and I/O cycles; i.e., only CPU memory reads are abortable once the cycle has begun. If the current cycle is a row hit but DRAM page miss (RAS precharge needs to be satisfied), de-activating ST# causes RAS# to remain de-asserted after RAS precharge has been met. A subsequent cycle to the same DRAM row would then proceed as a row miss instead of a page miss. If the current cycle is a row miss (RAS precharge already satisfied), only RAS# is activated with the correct setup/hold time on row address. CAS# is not activated in this cycle. **NOTE:** In this case, the actual row address latched is the one associated with the cache hit, effectively arming the 82359 for a page hit should the next CPU cycle be a cache miss to the same page. If the current aborted cycle is a DRAM page hit, then CAS# for the cycle will be activated only once, and subsequent CAS#s will be eliminated if needed (example: i486 burst to 1 or 2-way dword interleaved memory).

14.2 System Port Interface

Symbol	Type	Name and Function
SAS#	I/O	**SYSTEM ADDRESS STROBE:** SAS# is a bi-directional pin which indicates the presence of an 82359 system-side cycle. The falling edge of SAS# starts a system bus cycle and indicates the system address and status are valid on the bus. SAS# must remain asserted throughout the entire cycle, and signals the end of the system cycle with its rising edge. SAS# becomes an output when the host is sending a cycle through the 82359 to the system bus. The cycle starts with HAS# being asserted and the 82359 latching the host address. The 82359 decodes the latched address and determines that the address is not in its main memory address space, and therefore the cycle should be broadcast to the system bus. In the case of special cycles (flush, halt, etc.), SAS# goes low indicating the start of a system cycle, but the bus controller won't necessarily run an EISA cycle. This is to insure that these cycles which otherwise look like memory (386) or I/O (486) code writes are not broadcast in such a way as to accidentally corrupt any memory or I/O that happens to be selected by the address. SAS# becomes an input when the host does not own the system bus. The falling edge of SAS# transparently latches SA(31:2), SBE(3:0)#, SD/C#, SW/R#, SM/IO#, and SLOCK#, similar to its counterpart HAS#. System PST masters (Buffered Configuration) assert SAS# when they want to run a cycle. Since this cycle's destination may be to either the EISA bus or to main memory, the 82359 qualifies SAS# with an internal main memory chip select, derived from the system address. If the address is not to the 82359's memory space, the cycle is ignored by the 82359. Note that when a system PST master is talking to the 82359, the protocol is almost the dual of the host side. Note also that EISA master cycles to main memory do not generate SAS#. Rather, the 82359 monitors the EISA bus directly and runs the cycle based on the EISA START# signal.
IASALE#	I	**INTERNAL ADDRESS TO SYSTEM ADDRESS LATCH ENABLE:** IASALE#, an input from the EISA Bus Controller, causes the host address to be latched into the system address latch and driven onto the system bus. Also, during burst cycles to the system bus from the host CPU, this signal increments the burst address through the predictable burst order sequence. The incrementing circuit manipulates system address bits SA(5:2) to allow a burst of up to 16 dwords. In the lead off access to the system, the 82359 simply propagates the HA(31:2) value to the system address bus. The HA(5:2) initial value is also latched by the internal burst control circuit to fix the burst pattern. A subsequent low-to-high transition of IASALE# closes the transparent system address latch and internally increments address bits (5:2) to the next address in the burst pattern. The incremented address is then propagated to the system when a falling edge of IASALE# re-opens the latch. For all but the lead-off cycle of the burst, the source of SA(5:2) is not the host address, but the internal burst sequencer.

14.2 System Port Interface (Continued)

Symbol	Type	Name and Function
SHOLD	I	**SYSTEM HOLD:** SHOLD is an input to the 82359 which indicates that a system master needs the system bus to run cycles. The 82359 will respond with a SHLDA when the 82359 has given ownership of the system bus to the requesting master. SHOLD should be asserted throughout the entire time the system master needs system bus ownership, and the falling edge of SHOLD indicates to the 82359 that the bus is no longer needed by system devices.
SHLDA	O	**SYSTEM HOLD ACKNOWLEDGE:** SHLDA is an output from the 82359 acknowledging the SHOLD of the system, and indicates that the 82359 has given up the ownership of system bus. SHLDA will remain asserted as long as SHOLD remains asserted. When the 82359 asserts SHLDA, all system address and control will be tri-stated to allow the system device to run cycles. In the event of a posted write, if the system requires that the 82359's system address continue to drive until the end of the system cycle, then the system must mask SHOLD to the 82359 until the system cycle is complete (automatically taken care of when using the EBC).
SBREQ/ SBREQEN	O	**SYSTEM BUS REQ/SYSTEM BUS REQ ENABLE:** The generation of SBREQ is dependent on Cache Control Register bit 6, and whether it is a 385 system (as determined by Mode Register B, bit 2). If Mode Register B, bit 2 = 0 In this mode, the pin functions as SBREQEN and the function of this pin is a direct reflection of the Cache Control Register, bit 6. The HBREQ output from a 386 state tracker can be qualified externally with this to generate a bus request to the ISP. If Mode Register B, bit 2 = 1 The function of this pin is to generate a request to the system arbiter to get the system bus so as to run a CPU to system bus cycle. This can be disabled by Cache Control Register bit 6, which when high, will not activate the SBREQ.
SARDY	I	**SYSTEM ASYNCHRONOUS READY:** SARDY is input to the 82359, used to indicate a "not ready" condition of the system slaves when the 82359 is propagating host cycles to the system bus. (SARDY is the dual of HARDY.) The EBC will negate SARDY to the 82359 during host-to-system cycles. The EBC will re-assert SARDY to indicate the completion of the cycle, causing HARDY on the host side to be re-asserted. The rising edge of this signal indicates the end of the host cycle to the host PST (who originated the cycle) causing it to return READY to the CPU. Note that the 82359 expects the system to pull SARDY low, and uses this event to turn cycle extension control over to the system. From this point the 82359 is no longer responsible for cycle extension, except to forward SARDY going high to the CPU bus.

14.2 System Port Interface (Continued)

Symbol	Type	Name and Function
SBURST#	I/O	**SYSTEM BURST INDICATOR:** SBURST# is an input to the 82359 when system PST masters are running cycles to the 82359 and is an output to the system bus controller when the host is running cycles to the system. As an input, if SBURST# is asserted, the 82359 will run as many MDS# pulses to the memory as required to fetch an entire system line in response to a single SAS# trigger. The actual number of MDS# cycles will be a function of programmed system line length and of the dword interleave factor (IF(1:0)) of the accessed row. If SBURST# is not asserted, 82359 will run one and only one MDS# cycle. Even in this case, the system can still burst up to the number of dwords latched in the 82353 by a single MDS# pulse if the system monitors IF(1:0), SBRDY#, and SBLAST#. SBURST# becomes an output to the system bus controller when the host CPU is running bus cycles to the system. The asserted state indicates the host will be expecting to run burst cycles to the system bus. The number of cycles that are run when SBURST# is asserted is a function of the cache line length (programmed in the Line Size register), and the size of the system slave. For example, a 4-dword cache line length and a 16-bit wide slave would result in 8 back-to-back system cycles for a single HAS# trigger. It is the responsibility of the system bus controller to comprehend the cache line size length and run the correct number of system cycles. Note that system bursts are only defined for memory reads. If Host burst cycles to the system bus are not supported, the host PST should use the IF(1:0) = 11 condition to use normal CPU ready rather than the burst ready (i.e., if IF(1:0) = 11, ignore HBURST#). Then, in case of the EISA EBC, its SBURST# input should be tied high.
SD/C#, SM/IO#, SW/R#	I/O	**SYSTEM CYCLE DEFINITION:** SD/C#, SM/IO#, and SW/R# are bi-directional system cycle definition signals and are defined the same as their counterparts. (See HD/C#, HM/IO#, and HW/R# for the cycle definitions.) These pins are inputs when the system master owns the system bus and should be set up to the falling edge of SAS#. These signals become outputs when a host cycle is being forwarded to the system bus. Note that forwarding a cycle to the system bus does not automatically imply that SD/C#, SM/IO#, and SW/R# will be a direct reflection of their host counterparts. These pins may differ depending on what type of processor is defined for the host and system side (programmable in Mode Register A, bits 1 and 2 respectively). If the host processor definition differs from the system processor definition, the 82359 will translate the cycle definitions accordingly.
SBE(3:0)#	I/O	**SYSTEM BYTE ENABLES:** SBE(3:0)# are the byte enables for each of the bytes in the dword. These signals may be either inputs or outputs, depending on whether the 82359 is currently accessing the system bus. SBE(3:0)# become outputs when host CPU cycles are run to the system bus. They will be a straight reflection of the host port HBE#s, although subject to a format conversion in response to the host and system processor types as set in Mode Register A, bits 1 and 2. Should the host cycle be a cache line fill or a burst cycle, SBE(3:0)# will all be forced active regardless of the actual values of HBE(3:0)#. This insures EBC will properly assemble a full 32-bit word. When a system master owns the 82359 (as indicated by SHLDA asserted), the SBE#s become inputs receiving the byte enables from the system cycle.

1

14.2 System Port Interface (Continued)

Symbol	Type	Name and Function
SA(31:2)	I/O	**SYSTEM ADDRESS:** The SA address lines are bi-directional. They become outputs for host to system cycles, and along with the SBE(3:0)# provide the cycle address to the 32-bit system address bus. SA(5:2) are also controlled by the internal system burst sequencer and provide the correct address for host burst cycles to the system bus. The SA(31:2) lines become inputs whenever a bus master owns memory (SHLDA asserted).
SLOCK#	I/O	**SYSTEM LOCK CYCLE:** SLOCK# is an input to the 82359 when a system master owns the system bus. The system master asserts this signal when it requires uninterrupted access to a device. There is no effect in asserting SLOCK# if the system master never accesses memory. The 82359 sees SLOCK# asserted and monitors system addresses. If a system address which is contained in main memory is seen, the 82359 will request memory ownership from the host PST through the HMREQ/HMACK protocol. Once ownership is obtained, SLOCK# asserted causes the remainder of the accesses to main memory to be locked; i.e., memory is not arbitrated for until an actual memory access is decoded. If the locked cycles running on the system side do not access the memory during the lock period, memory will not be arbitrated for and will be freely accessible to the CPU. SLOCK# is an output when the CPU owns the system bus. When a host locked cycle is detected, the 82359 will not start the actual cycle until both the memory and system bus are owned by the CPU. (Lock cycles arbitrate for the system regardless of the destination of the cycle.) SLOCK# can be used to lock access to the dual ported memory or any other device which can be accessed by a master who is not visible from the system bus. SLOCK# remains asserted until the end of the lock cycle.
SMREQ	O	**SYSTEM MEMORY REQUEST:** SMREQ may be thought of as the dual of HMREQ. The 82359 will assert SMREQ to system PSTs when it or the host wants to own memory. The 82359 will immediately assert SMREQ whenever the system does not need memory; i.e., the default state of SMREQ is asserted.
SMACK	I	**SYSTEM MEMORY ACKNOWLEDGE:** SMACK asserted indicates that the memory is not owned by a system PST master, just as HMACK indicates that memory is not owned by the host master. When SMACK becomes de-asserted, the system PST master has memory ownership. Upon seeing the removal of the 82359's memory request, the system PST should also remove the acknowledge SMACK. The SMREQ/SMACK protocol is not typically implemented in the Standard Configuration and SMACK should be tied high.
INVLA#	I	**INVERTED LATCHABLE ADDRESS LINES:** This strap causes the 82359 to treat the most significant byte of the system address, SA(31:24), as inverted. If INVLA# is low (asserted), addresses from the host which are being forwarded to the system have their HA(31:24) inverted before being driven onto the SA lines. When an EISA/ISA master owns the system bus, the 82359 inverts the incoming address bits prior to decode. Also, if a system master write causes a snoop, the incoming high byte system address bits are inverted to their "true" state before being driven onto the host bus for snooping. If this strap is high, SA(31:24) are treated as non-inverted.

14.2 System Port Interface (Continued)

Symbol	Type	Name and Function
BCLK	I	**EISA BUS CLOCK:** BCLK provides the 82359 with a reference for sampling EISA specific signals. Since the EISA bus is synchronous to BCLK, events on the EISA bus are sampled synchronous to BCLK edges without regard to frequency or duty cycle.
START#	I	**EISA START# SIGNAL:** START# provides timing control at the start of a cycle. A bus master asserts START# after SA(31:2) and SM/IO# become valid, and negates START# on a rising edge of BCLK after one BCLK period. The 82359 samples the EISA address and cycle definition on the falling edge of START# (internally delayed by approx. 30 ns).
CMD#	I	**EISA CMD# SIGNAL:** This input is the EISA CMD# signal used by the 82359 in directly tracking EISA master cycles. CMD# controls the bus data timings and its rising edge signals the end of the current EISA cycle.
MRDC#	I	**MEMORY READ COMMAND:** An ISA bus master asserts MRDC# to indicate that the addressed slave should drive its data onto the system bus. The 82359 will respond directly to MRDC# only if the system is in Non-Concurrent Mode (bit 0, Mode Register C) and MRDC# comes earlier than START#. Activation of this signal is automatically ignored if REFRESH# is active in an ISA master initiated refresh. In this case START# will always precede MRDC# since the EBC rather than ISA Master generates MRDC#.
MWTC#	I	**MEMORY WRITE CMD#:** The ISA master or system will assert MWTC# to indicate that the destination may latch the data on the system bus. The 82359's use of MRDC# and MWTC# are a function of whether the 82359 is in Concurrent or Non-Concurrent Mode. **1. Concurrent Mode** Bus masters can run system bus cycles independently of the ownership state of main memory. The 82359 treats MRDC# and MWTC# explicitly as don't cares, relying instead on the EBC to translate ISA master cycles to EISA master format; i.e., the 82359 behaves exclusively as an EISA slave, monitoring only START#, CMD# and MRDY (EXRDY). **2. Non-Concurrent Mode** System master cycles can be run only after arbitrating for the system bus AND gaining main memory from the 82359, regardless of the destination of the system cycle. One application of Non-Concurrent Mode is support for ISA masters that do not monitor CHRDY on the bus. These masters run memory cycles based on standard cycle lengths asynchronous to BCLK. Since these masters do not monitor CHRDY, it is not possible to indicate a memory busy condition (such as the host currently owning main memory). Therefore, to support these kinds of masters on the system bus, all system bus ownership requests must first arbitrate for the bus ownership, then gain memory ownership before any cycles are allowed to start. By doing so, the cycle is guaranteed to complete in worst case time. In Non-Concurrent Mode both ISA and EISA signals are visible (MRDC#, MWTC# and START#, CMD#, EXRDY). The 82359 will be monitoring START# as well as MRDC#/MWTC# and locks out either one depending on what it sees first. For example, if the 82359 sees MRDC# before START#, it will use MRDC# to run the cycles, and will not pay attention to START# (i.e., it runs as an ISA slave). If it sees START# first, it ignores MRDC#/MWTC# and runs as an EISA slave. As a direct ISA slave, the 82359 does not request cycle extension.

14.2 System Port Interface (Continued)

Symbol	Type	Name and Function
MSBURST#	I	**EISA BURST CYCLE INDICATOR:** An EISA bus master asserts MSBURST# to indicate that it is capable of EISA BURST cycles. The slave indicates that it is capable of supporting a burst via the slave burst indicator (SLBURST#), which in the case of main memory is the 82359 DRAMCS# signal. MSBURST# is asserted with the address lines for the second and all subsequent cycles of the burst, and is sampled on the rising edge of BCLK by the 82359. All cycles within an EISA burst sequence are confined to a 1 kbyte boundary address space, guaranteeing that they are DRAM page hits. The cycles are pipelined and a new address is on the EISA bus at the rising edge of every BCLK. The 82359 can insert wait states by pulling MRDY. (Note also that all cycles within a single burst sequence are either all reads or all writes.)

14.3 Memory Interface

Symbol	Type	Name and Function
RAS(3:0)#	O	**ROW ADDRESS STROBE:** The 82359 provides four Row Address Strobe signals for the DRAMs, one per ROW. RAS0#, RAS1#, RAS2# and RAS3# are connected to ROW 0, 1, 2 and 3 of the memory array respectively. The falling edge of RAS# is used by the DRAMs to latch in the row address. The number of RAS# signals that will remain asserted after its access is completed depends on the setting of the RAS Mode Register. If the number of active banks is programmed to one, only the current RAS# signal being used will remain asserted after the access is completed. If the number of active banks is programmed for two, the most recent RAS#s of the set RAS0#, RAS1# and the set RAS2#, RAS3# (one from each set) will remain asserted; i.e., RAS0# and RAS1# will never be low at the same time; likewise RAS2# and RAS3# will never be low at the same time. This rule is to prevent data contention between row 0–row 1, and row 2–row 3 SIMMs since row 0 and 1 share one set of CAS# lines, and row 2 and 3 share one set of CAS# lines. (See the RAS Mode Register for more details.)
CAS(7:0)#	O	**COLUMN ADDRESS STROBE:** The 82359 provides eight CAS# signals for the DRAM array. These eight CAS lines are broken-up into 2 groups of four; CAS(3:0)# for row 1 and row 2 to share, CAS(7:4)# for row 2 and row 3 to share. The CAS signals are "byte based" meaning that CAS0# is connected to byte 0 of all four dwords in the row (it is also connected to the same bytes of the other row in the row-pair). For example, activating RAS0# and CAS0# will enable Byte 0 in all four dwords in row 0. During a read, all four CAS lines in the group are asserted. This is done regardless of whether or not all four byte enables of the current master are asserted. This provides the entire row of data to the 82353 for latching. In a write cycle, only the CAS#'s that map to active byte enables are driven. (WE(3:0)# select the appropriate dword which contains the desired byte.) Thus in a write, RAS# selects the row, CAS# selects the byte(s), and WE# selects the dword. This arrangement of two groups of shared CAS# signals gives the flexibility of using single or double density SIMM modules for row-pair implementation. For host or system memory writes, or EISA/ISA memory reads, only one CAS# pulse is generated since these are all single cycle accesses. In burst (i486 type) read cycles, the number of CAS# pulses will depend on the dword interleave of the row to which the access is occurring as well as the programmed host or system line size.

14.3 Memory Interface (Continued)

Symbol	Type	Name and Function
WE(3:0)#	O	**MEMORY WRITE ENABLES:** These are buffered using external buffers before they drive the DRAM array. The RAS# and CAS# signals select a row and column of the memory array. The WE# signals select the dword within a row during write operations. Thus, using RAS#, CAS#, and WE# signals, any byte of a particular row can be identified. Note that this implies that in row accesses, CAS# selected bytes in the dwords without active write enables will actually be DRAM reads. To prevent contention on the DRAM data bus, the write enables themselves are connected to the WE inputs of the 82353 and actually serve as the 82353's memory output enables, such that only the dword being written is driven to the DRAM.

RMADDR(7:0)
MADDR(8:0)
	O
	I/O
		Row Specific Memory Address
		Common Memory Address
		The DRAM array is provided with 11 multiplexed address lines. Nine of these MADDR(8:0) provide a common address to all rows in the array. Along with the previous nine bits, each row is also given two row specific bits from RMADDR(7:0). This provides each row with 11 address bits, enough to address 4 meg DRAMs.
		The MADDR(8:0) are the 9 address bits common to all rows. Typically these address lines are outputs, providing row and column address information to the DRAM array. As a second function, eight of these bits, MADDR(7:0), become the I/O port for passing data to and from the 82359's internal registers.
		These eight address lines become inputs when writing to the 82359's internal registers. During a write to the internal registers, the 82359 will assert DEN#, causing MADDR(7:0) to tri-state and become available to accept input data. During register reads, the data from the register being read will be driven onto MADDR(7:0).
		In addition to the common memory address line of MADDR(8:0), a second group of row-specific memory address lines is provided by RMADDR(7:0). These eight bits are broken into four groups of two and each group connects to the corresponding row of the memory array. These two bits plus the nine bits provided by MADDR(8:0) provide the 11 bits needed to address 4M memories.

14.4 82353 Interface

<table>
<tr><th>Symbol</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>H/S#</td><td>O</td><td><b>HOST/SYSTEM SELECT:</b> The H/S# output indicates which port currently owns main memory and is sent to the 82353 to select either host or system data paths, to or from memory.</td></tr>
<tr><td>SEL(1:0)</td><td>O</td><td><b>SELECT:</b> The SEL(1:0) determine which one of the four possible dwords latched into the 82353 should be sent to the host. During host read sequences, the 82359/82353 will latch each of the dwords in the entire row being accessed. The SEL(1:0) bits tell the 82353 which of the latched dwords should be sent to the host. Should the host cycle be a burst, the remaining sequence of dwords is known by the 82353 based on IF(1:0) (since the i486 burst sequence is fixed, determined by the lead-off address). The burst orders for all SEL and IF values are shown in table below:
<table>
<tr><th>SEL(1:0)</th><th>IF(1:0)</th><th>Memory Array Dword Being Accessed</th></tr>
<tr><td>00</td><td>00 (4-way)</td><td>0-1-2-3</td></tr>
<tr><td>00</td><td>01 (2-way)</td><td>0-1-0-1</td></tr>
<tr><td>00</td><td>10 (1-way)</td><td>0-0-0-0</td></tr>
<tr><td>00</td><td>11</td><td>System Access</td></tr>
<tr><td>01</td><td>00 (4-way)</td><td>1-0-3-2</td></tr>
<tr><td>01</td><td>01 (2-way)</td><td>1-0-1-0</td></tr>
<tr><td>01</td><td>10 (1-way)</td><td>1-1-1-1</td></tr>
<tr><td>01</td><td>11</td><td>System Access</td></tr>
<tr><td>10</td><td>00 (4-way)</td><td>2-3-0-1</td></tr>
<tr><td>10</td><td>01 (2-way)</td><td>2-3-2-3</td></tr>
<tr><td>10</td><td>10 (1-way)</td><td>2-2-2-2</td></tr>
<tr><td>10</td><td>11</td><td>System Access</td></tr>
<tr><td>11</td><td>00 (4-way)</td><td>3-2-1-0</td></tr>
<tr><td>11</td><td>01 (2-way)</td><td>3-2-3-2</td></tr>
<tr><td>11</td><td>10 (1-way)</td><td>3-3-3-3</td></tr>
<tr><td>11</td><td>11</td><td>System Access</td></tr>
</table></td></tr>
<tr><td>MDS#</td><td>O</td><td><b>MEMORY DATA STROBE:</b> A low to high transition on MDS# indicates that the data from DRAMs is valid. This signal is used by the 82353 to latch memory data. During a memory read cycle, the MDS# signal goes low at approximately the same time CAS# goes low. After a precise delay, controlled by an internal delay line in the 82359, MDS# goes high. The 82359 determines from the dword interleave factor of the row being accessed and the cycle definition exactly how many memory cycles, and therefore, how many MDS# pulses need to be executed to fulfill the host or system cycle. Note that the 82359 will perform a maximum of four consecutive MDS# pulses. Four pulses are enough to satisfy the 16-byte default line size access from the minimum memory width of 4 bytes. To perform larger burst accesses, the design must ensure a wider memory.</td></tr>
<tr><td>HIOE#</td><td>O</td><td><b>HOST INTERNAL OUTPUT ENABLE:</b> HIOE# is a signal used for communication between the 82359 and the 82353. The 82359 will assert HIOE# during host write cycles. When active, HIOE# enables the host data port to drive the internal bus of the 82353. From the internal bus, data is provided to both the system and memory output buffers.</td></tr>
</table>

14.4 82353 Interface (Continued)

<table>
<tr><th>Symbol</th><th>Type</th><th>Name and Function</th></tr>
<tr><td>MIOE#</td><td>O</td><td><b>MEMORY/SYSTEM# ENABLE:</b> MIOE# is used in communication between the 82359 and the 82353. The 82359 will assert MIOE# when memory has been selected as the source of data (i.e., memory read cycles). When the MIOE# is asserted and HIOE# de-asserted, the 82353 allows data stored in its memory read latches to be driven onto the 82353 internal bus. The combination of MIOE# and HIOE# de-asserted causes the 82353's system port to drive its internal bus. The table below summarizes the function of HIOE# and MIOE# in controlling the choice of data source.
<table>
<tr><th>HIOE#</th><th>MIOE#</th><th>Data Source</th><th>Data Destination</th></tr>
<tr><td>0</td><td>0</td><td>Host</td><td>Memory or System (MIOE# active is meaningless since HIOE# dominates)</td></tr>
<tr><td>0</td><td>1</td><td>Host</td><td>Memory or System</td></tr>
<tr><td>1</td><td>0</td><td>Memory</td><td>System or Host</td></tr>
<tr><td>1</td><td>1</td><td>System</td><td>Memory or Host</td></tr>
</table>
</td></tr>
</table>

14.5 Miscellaneous Decodes and Control Signals

Symbol	Type	Name and Function
IF(1:0)	O	**(Dword) INTERLEAVE FACTOR:** IF(1:0) reflect the dword interleave factor of the row being accessed by the 82359. The dword interleave factor is programmed into the configuration registers of the 82359 when the system is booted. The IF(1:0) bits reflect the contents of the configuration register for the row currently be accessed and are used by the 82353 and PST during burst cycles.
REFRESH#	I	**SYSTEM REFRESH:** REFRESH# asserted indicates that a system bus master is running a refresh cycle. The 82359 ignores the system address on the bus during these cycles, since the 82359 supplies its own refresh address. The 82359 supports either a "coupled" or "decoupled" refresh, selectable through Mode Register A, bits 6 and 7. If decoupled, the 82359 runs refresh based on its internal refresh timer and the REFRESH# input is ignored. Coupled refresh allows the 82359 to monitor the EISA REFRESH# line and the 82359 will run a refresh cycle to main memory upon detection of the REFRESH# signal's falling edge.
PER#	I	**PARITY ERROR:** PER# is a parity error input to the 82359. Each of the 82353 Data Path devices combinatorially generates a PER# when the generated parity does not match the parity read from main memory. (Both 82353's PER# outputs should be logically OR'ed before being input to the 82359). This input is used by the 82359 to generate PERSTB# only for non-PST initiated read cycles from main memory.

14.5 Miscellaneous Decodes and Control Signals (Continued)

Symbol	Type	Name and Function
PERSTB#	I/O	**PARITY STROBE:** PERSTB# is the main memory parity error indication. It is an open-drain signal and will be driven by the 82359 when a parity error is generated by the 82353 Data Path during non-PST master (DMA, EISA/ISA master) read cycles to main memory. The 82359 samples its PER# input when data is stable and will pulse PERSTB# low if PER# is sampled active. Since this signal is open-drain signal, and is effectively the wired "OR" of all parity error pulse sources (EISA devices) in the system, the 82359 monitors this signal for use in error trapping. The occurrence of a PERSTB# is used internally to freeze the contents of the address, status, and byte enable error trap registers. Until a PERSTB# pulse occurs, the 82359 updates the error trap registers in every cycle, regardless of the cycle's source. When a pulse occurs, the latches are frozen until read by an error recovery routine.
MRDY	I/O	**MEMORY READY:** MRDY is an open collector output from the 82359 to the EISA bus and typically connected to the EISA bus signal EXRDY. The 82359 will pull MRDY low when it needs extra time to complete the system cycle (i.e., memory is currently owned by the host). EISA masters and system PST masters monitor MRDY and, upon sampling MRDY de-asserted, add wait states to the current cycle until MRDY returns to the asserted state. MRDY is both an input and output. As an input, MRDY is used by the EISA Cycle Tracker to monitor EISA activity. As an output, MRDY can be de-asserted for three occasions. First, MRDY is pulled low during EISA cycles to main memory which the 82359 can not complete in the standard EISA cycle time. In this case, wait-states need to be added. The 82359 accomplishes this by driving MRDY low to hold off the current EISA cycle. This situation may occur during EISA master cycles to main memory at times where the system port is not the owner of main memory. In this case, the 82359 will pull MRDY low while it asks the host PST for memory ownership. The rising edge of MRDY allows the cycle to complete. MRDY may also be pulled low when the System Throttle has expired and the host is "sneaking" cycles to main memory (see throttle descriptions). Secondly, MRDY can be deasserted for EISA burst write cycles which cause snoops. If the 82359 detects that a snoop needs to be propagated to the host, it will pull MRDY low for 1 BLCK to allow time for the host snoop logic to handle the snoop before it is lost. Depending on the snoop service latency by the host PST, not all snoop filter misses will result in a cycle stall by MRDY. Third, MRDY is used during system PST master cycles. The 82359 will be de-asserted MRDY during system PST write cycles to main memory until the resulting snoop to the host cache has been completed (as signified by the trailing edge of SNUPACK). In this case, the result of MRDY being de-asserted does not cause the current system-to-memory cycle to be stalled off waiting for MRDY to return to the high state. In fact this cycle will run as a normal deterministic cycle and complete as reflected by PAGEHIT# and CYCLN(2:0). The effect of a de-asserted MRDY, in this case, is when the system PST samples MRDY on a subsequent system-to-memory write, and should it still be de-asserted, the system PST must hold-off that cycle until MRDY returns to the asserted state. Should MRDY be asserted at the sampling time, the previous snoop has finished and the system PST is free to begin the system-to-memory cycle. The use of MRDY in this way prevents the system PST master from "outrunning" the 82359's ability to relay snoop cycles to the host cache.

14.5 Miscellaneous Decodes and Control Signals (Continued)

Symbol	Type	Name and Function
BIOSCS#	O	**BIOS CHIP SELECT:** BIOSCS# is an output from the 82359's system address decoder to indicate the BIOS ROM on the system bus is being accessed. The BIOSCS# signal combinatorially follows the system address. BIOS EPROMs are mapped to both 000E0000 to 000FFFFF and FFFE0000 to FFFFFFFF. Two mappings are provided because DOS only expects BIOS to reside at addresses between 000E0000 and 000FFFFF, while the CPU needs to read its reset vector at the top of its address space. The 82359 optionally supports a 64k rather than 128k BIOS. This option is selected via bit 3 of Mode Register B. In this case BIOSCS# only responds to the upper halves of the BIOS address ranges. If the 82359 sees a cycle address in the top of the physical address space, i.e. the top meg in the 4-gigabyte space, then the cycle will be forwarded to the system ROM regardless of whether the BIOS is currently shadowed or not.
VGAMSL#	O	**VGA MEMORY SELECT:** Output from the system address decoder to indicate VGA memory select. Memory cycles within the address range 000A0000 to 000BFFFF will be decoded. It combinatorially follows the system address. The VGA controllers have only A0 to A19, so the 82359 will essentially decode the range within the first meg that the VGA resides. This signal typically goes to the VGA sub-system to be used as a VGA memory select.
DRAMCS#	O	**DRAM CHIP SELECT:** DRAMCS# is an output from the 82359 by the system address decoder to indicate that the current system master cycle is to main memory. Like BIOSCS# and VGAMSL#, DRAMCS# responds to system addresses only; even though the host may be accessing main memory, DRAMCS# reflects the result of the system address.
LOCKEN#	O	**LOCK ENABLE:** This is a static output that is a direct reflection of the Cache Locked Cycles bit (bit 4 of the Cache Control Register). This pin can be used to enable or disable the LOCK# signal of the 386 from reaching the 82385 cache controller.
DEN#	O	**DATA ENABLE:** This pin is asserted in response to host (only) accesses to the Index Register (IR-22H), Data Register (DR-23H), and, if enabled by Mode Register A, bit 3, in response to a 486-initiated cache flush cycle. The 82359 uses MADDR(7:0) as its data bus for the programming/reading of its internal registers. All data accesses to the 82359 are done through the I/O port 22h, 23h scheme. This pin is used to control an external '245 data transceiver which enables data to/from the X-DATA bus from/to MADDR(7:0). During a write to an 82359 port, and during a 486 initiated cache flush, the 82359 tri-states MADDR(7:0) so that DEN# assertion is free to enable the '245 to drive onto MADDR(7:0) without contention. This pin is also typically used to flush the cache on writes to the 82359's internal ports. To prevent unnecessary cache flushes during register reads, DEN# should be qualified externally with the 82359 SW/R# pin before driving the cache flush input. Note that when the indexing scheme is used to access the 82359, DEN# is asserted in a read cycle only if the 82359 is selected via its chip ID. This insures the transceiver will not be enabled onto the peripheral data bus and potentially cause contention should another device using the index scheme respond. (See the section which discusses internal register programming for more information.) Note that when using the index scheme, DEN# is asserted in a write cycle to the IR or DR regardless of whether or not the 82359's is ID selected at that time. This is necessary to allow free write access to the chip ID register. In this case, since it is a write cycle the '245 is pointed at the MADDR(7:0) bus, and the 82359 will float this bus. Thus no contention occurs, even though the write may be ignored.

14.5 Miscellaneous Decodes and Control Signals (Continued)

Symbol	Type	Name and Function
SPEED(1:0)	I/O	**DRAM SPEED:** These are static "quasi-bidirectional" open drain port pins accessible via the 82359's DRAM Speed Register. Quasi-bidirectional means that if a "0" is written to the port, the open drain pin is actively pulled low, but if a "1" is written, the open drain floats. A port read looks not at the internally stored write bit but rather directly at the input pin such that even if a one is written to the port, an external device may pull the pin low causing a 0 to be read. Typically, these ports are written with DRAM speed information by the BIOS. Alternatively, SIMM modules with hard-wired speed attributes can be read via these ports. The host or system PST typically uses these bits along with IF(1:0) to determine the appropriate number of clocks in a burst cycle. These bits have no direct impact on 82359 behavior. If a particular design does not require one or both of these bits, it can apply the ports to any other application desired.
CYCLN(2:0)	O	**CYCLE LENGTH FEEDBACK:** CYCLN(2:0) returns a 3-bit code, programmed by the BIOS, for the required number of wait states for the current host or system deterministic access. During a non- deterministic access, these lines should be ignored. If HARDY is sampled asserted by the PST, the cycle is a deterministic cycle and the cycle length is given by the value of CYCLN(2:0). In case of a deterministic burst, CYCLN(2:0) reflects the length of the lead off access, and subsequent cycle lengths for the remaining dwords of the burst can be determined by the PST from IF(1:0) and SPEED(1:0). The number of wait states in a memory read/write cycle are affected by dynamic conditions; specifically, page hits, page misses or row misses. During initialization, the BIOS programs the 82359's cycle length registers with the number of wait states for each combination of DRAM page hit/miss, bank hit/miss, and write/read. There are actually two register sets; one for host accesses and one for system accesses. During a memory cycle, the 82359 decodes the cycle type and drives the appropriately stored value onto the CYCLN(2:0) pins. The external PST control logic then uses this value to count down the CPU clocks required before RDY# or BRDY# is returned to the PST master. Note that the 82359 simply drives the preprogrammed values onto the pins. The actual values have no impact on 82359 behavior. Hence, the system designer is free to modify the interpretation of these pins and the associated registers if desired.
PAGEHIT#	O	**PAGE HIT INDICATOR:** PAGEHIT# is an active low signals which indicates the current cycle to main memory can be executed by a DRAM page hit memory cycle. The PST should monitor this signal to determine that the current cycle is the fastest possible memory cycle.
OSC	I	**SDL REFERENCE FREQUENCY CLOCK:** This pin is driven by a 40 MHz, 50% ±10% duty cycle input used by the free running internal delay line (SDL) and for internal signal synchronization purposes. The throttles/watchdog registers and memory arbiter also use this signal as their time base.
RESET	I	**POWER-ON RESET:** RESET causes all programmable registers and state machines to be set to the initial state. This input should be activated only during the power-up sequence.
TEST#	I	**TEST PIN:** In a system, this pin must be pulled high (inactive). It is used for testing the device.

15.0 FUNCTIONAL TIMING DIAGRAMS

The following section provides functional timing diagrams for several 82359 cycles. These diagrams are provided to illustrate the functional characteristics of the part and do not take into account any propagational delays. All diagrams assume the 82359 is implemented in Standard Configuration.

15.1 Host to Memory Read—Page Hit

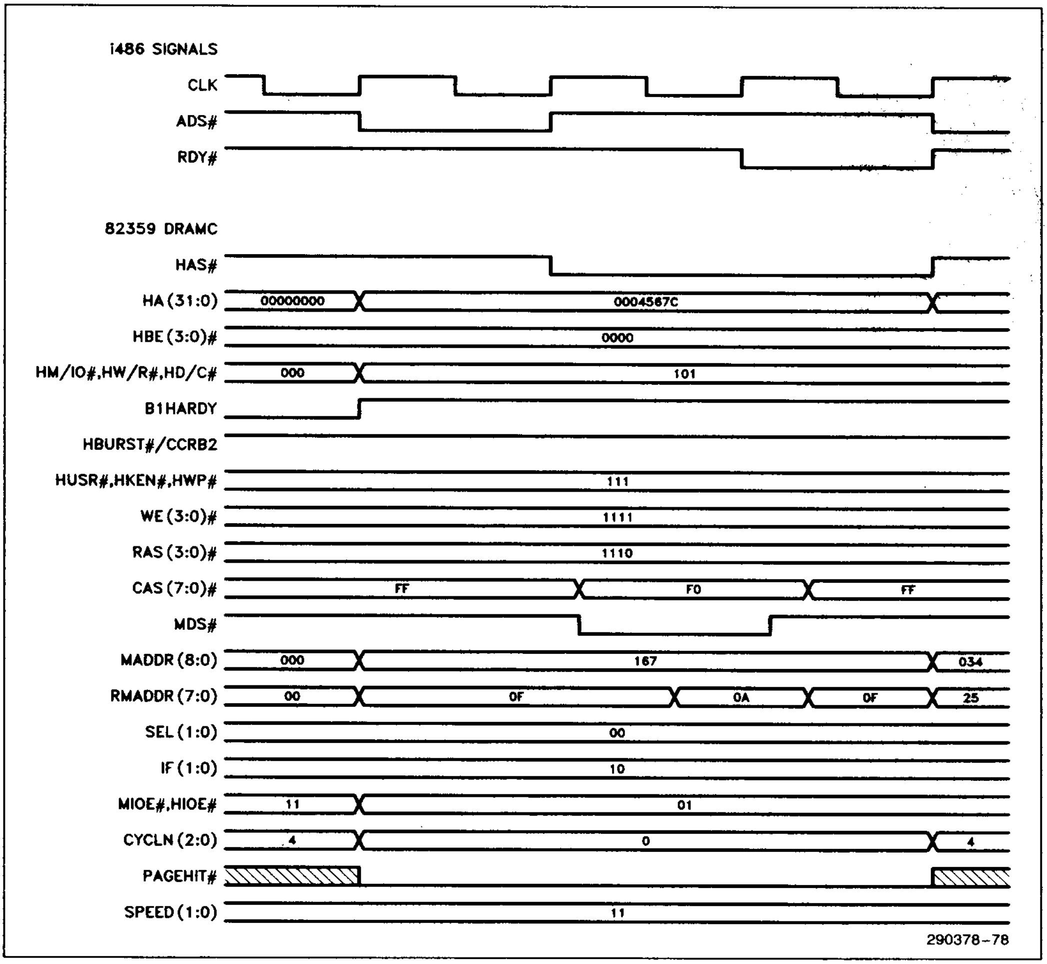


Figure 15-1. Host to Memory Read—Page Hit

The above figure depicts a host-to-memory read cycle. The cycle begins with the host providing the cycle address. From this address, the 82359 immediately begins decoding HARDY, CYCLN(2:0), PAGEHIT#, MIOE#, HIOE#, HBURST# and begins generation of the DRAM row and column address. In the above case, the host address is to a memory location in main memory and, therefore, the 82359 leaves HARDY in its asserted state to indicate a deterministic cycle. The 82359 also configures MIOE# and HIOE# to enable the 82353 Data Path for a route from memory to host. Also from the host address and memory configuration registers, the 82359 decodes the appropriate row and column DRAM address. In this case, the row happens to be a row used during the last RAS# cycle and the value programmed into the Page Hit Cycle Length Feedback Register is driven onto the CYCLN(2:0) lines and PAGEHIT# is driven active. The Programmable Attributes (HKEN#, HWP#, and HUSR#) are decoded off the host address as well.

The 82359 cycle actually starts with the falling edge of HAS#. Since the address has already been decoded (resulted in a DRAM page hit in this case), the CAS# signals for the appropriate row are asserted immediately, latching the column address in the DRAMs. (When CAS# is actually fired is dependent on the value of the Programmable Timing Registers.) Note that during main memory reads, all CAS# lines for that row are asserted to enable the reading of the entire row's data. MDS# also is asserted with CAS# and its rising edge (also programmable) causes the 82353 Data Path device to latch the memory data.

When the PST has counted down from the CYCLN(2:0) value, it will de-assert HAS#, ending the host cycle.

15.2 Host to Memory Read—Page Miss

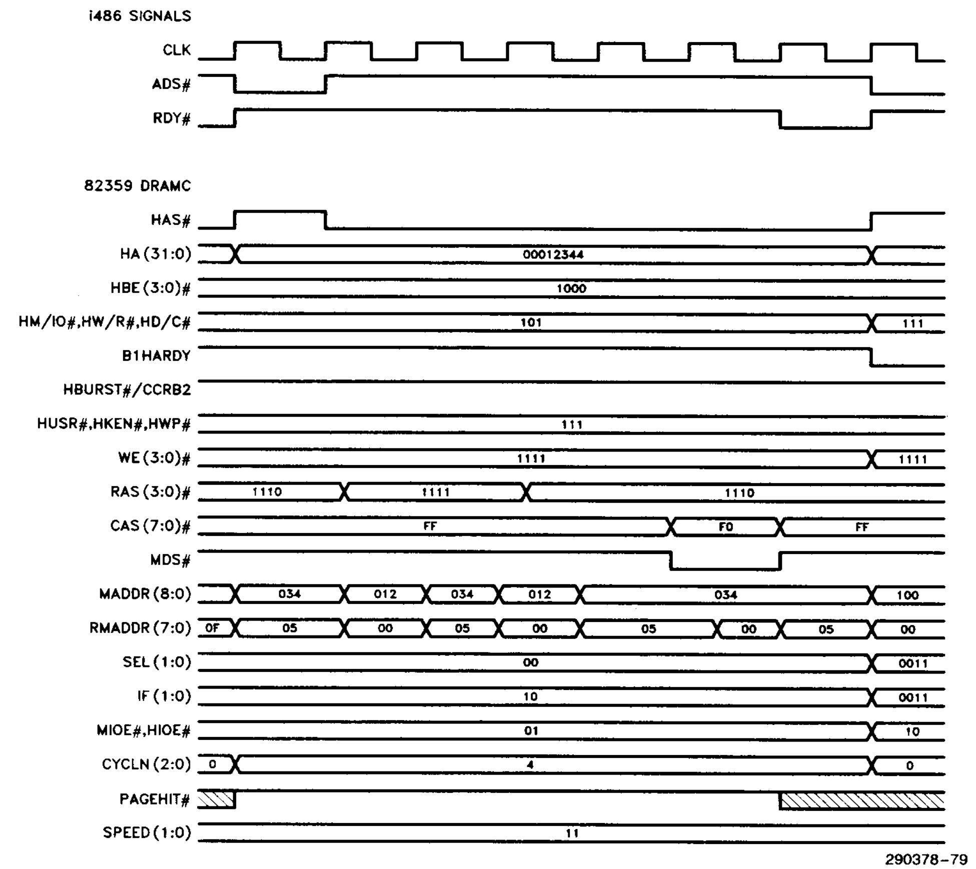


Figure 15-2. Host to Memory Read—Page Miss

The case shown above illustrates the sequence for a DRAM page miss. Specifically, One Active RAS# Mode is shown as determined from the single RAS# left active from the previous cycle.

The 82359 determines from the host address that a DRAM page miss memory cycle is required. Since a new DRAM row needs to be strobed in with the already active RAS#, row precharge must first be satisfied. The RAS# is deactivated for the programmable row precharge period and then asserted, latching in the new DRAM row address. The MADDR(8:0) and RMADDR(7:0) change state to reflect the column address which is latched into the DRAM by the falling edge of CAS#. Like all memory read cycles, all CAS# signals for that row are activated to allow the reading of all data bytes in the row.

MDS# is asserted with CAS# and its rising edge causes data to be latched into the 82353 Data Path. The cycle ends when the PST counts down to zero from the returned CYCLN(2:0) value and returns RDY# to the CPU.

15.3 Host to Memory Read—Row Miss

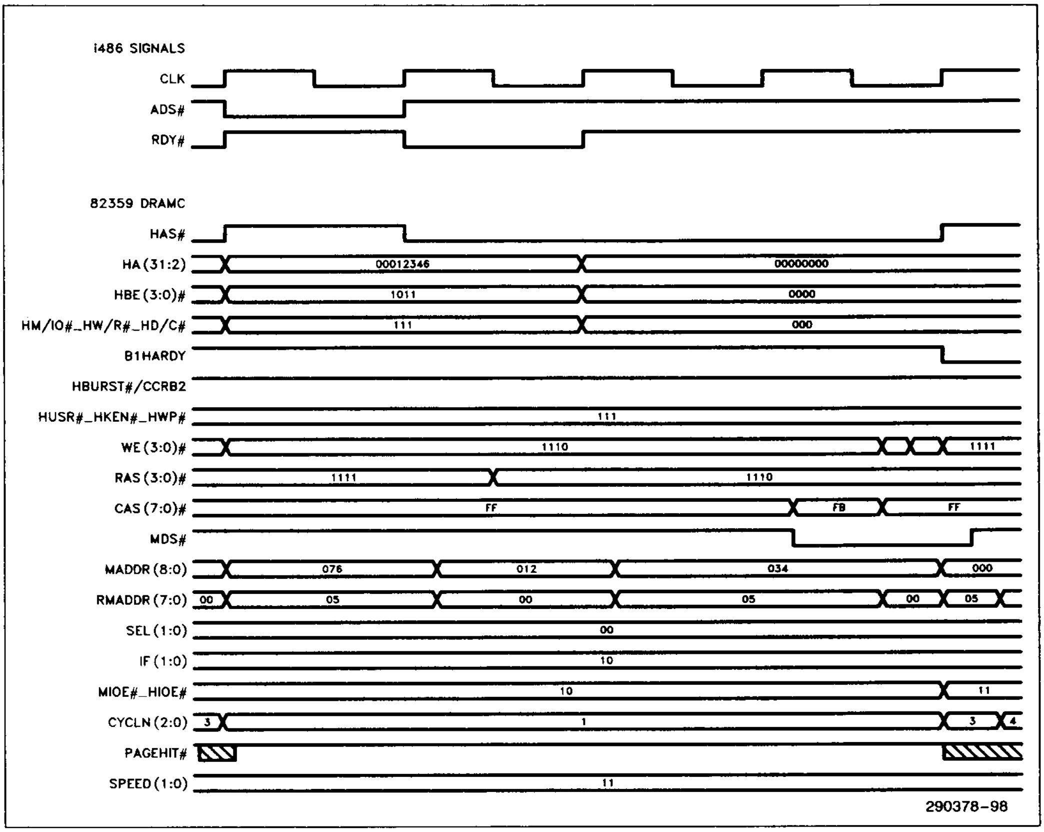


Figure 15-3. Host to Memory Read—Row Miss

The above diagram illustrates a host-to-memory access which results in a row miss DRAM cycle. In this case, the current memory access is to a row of the memory array which has already satisfied its row precharge requirement before the host cycle started. This allows the 82359 to bypass the row precharge sequence and assert RAS# immediately after the start of the host cycle (after the programmed delay, HP1/SP1 has expired).

15.4 Host to Memory Burst Read

(1-Way Dword Interleave)

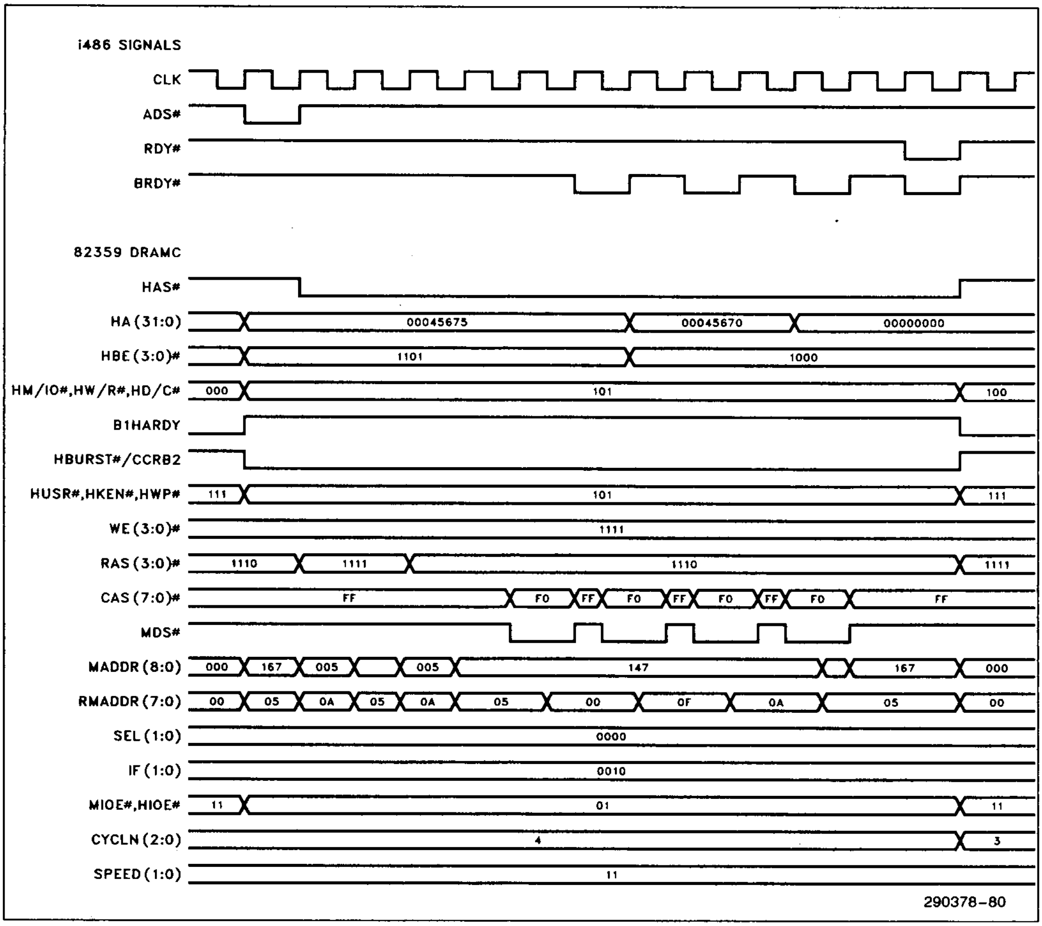


Figure 15-4. Host to Memory Burst Read (1-Way Dword Interleave)

The host-to-memory burst read is similar to a single dword read for the lead-off memory access. The difference between the two is that the 82359 determines that the host address is to memory which is capable of bursting. This is shown in the diagram by HBURST# being activated. The host PST typically samples HBURST# at the first dword transfer to determine whether to return RDY# or BRDY# and also to determine if a burst transfer is available.

The example above is one which shows One Active RAS# Mode and the lead-off cycle causes a DRAM page miss (and thus row precharge must be satisfied). The IF(1:0) signals show that the row being accessed is 1-way dword interleaved, and thus four memory accesses are required to complete the burst. For each of the three remaining accesses of the burst, only CAS# and MDS# need to be strobed because each memory access is guaranteed to be a DRAM page hit.

The value of CYCLN(2:0) returned during the host cycle refers to the time required for the lead-off access of the burst only. The time required to fetch the remaining dwords of the burst is determined by the PST from IF(1:0) and SPEED(1:0).

15.5 Host to Memory Burst Read

(2-Way Dword Interleave)

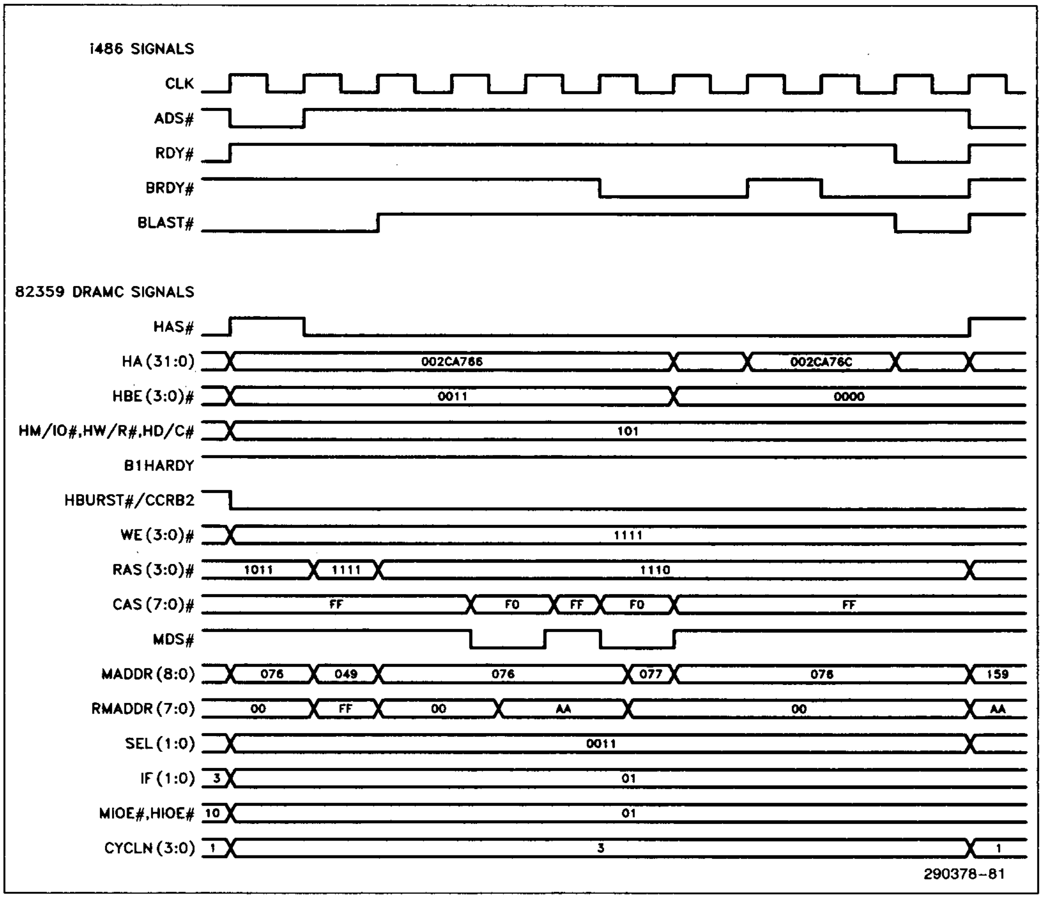
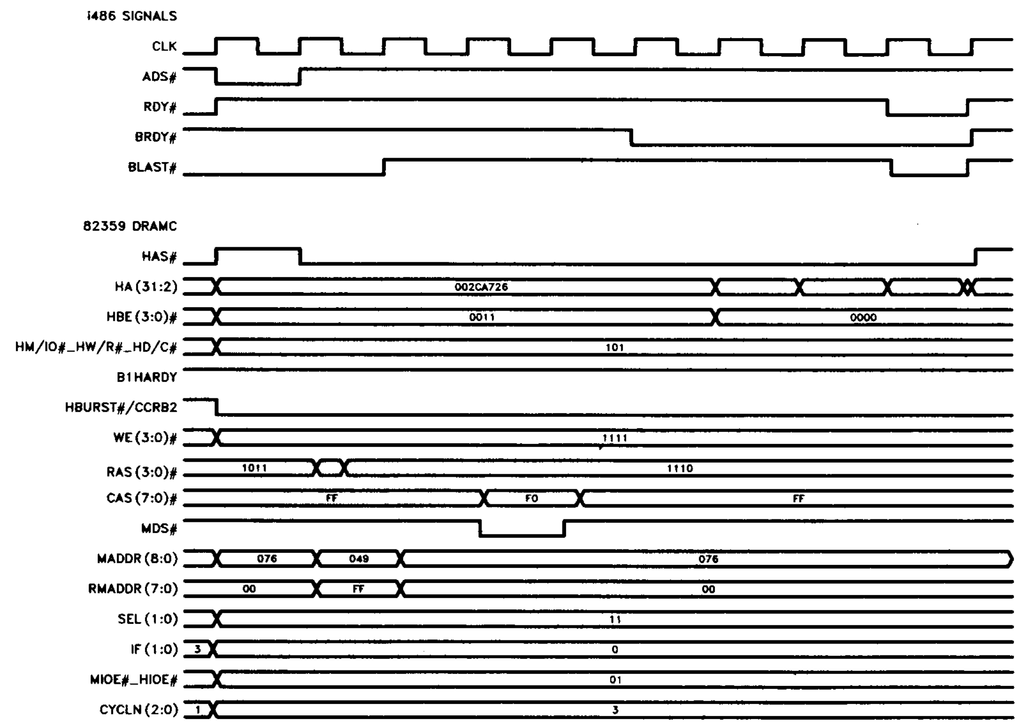


Figure 15-5. Host to Memory Burst Read (2-Way Dword Interleave)

This diagram is very similar to the previous diagram, differing in that the row of memory being accessed is 2-way dword interleaved. In this case the memory is 64 bits wide and two dwords are read from the array with each memory access. Therefore only two memory access cycles (two CAS# and MDS# assertions) are needed to complete the four dword burst.

15.6 Host to Memory Burst Read

(4-Way Dword Interleave)



290378-95

Figure 15-6. Host to Memory Burst Read (4-Way Dword Interleave)

This diagram presents the case of main memory being four-way dword interleaved. Since the row being accessed is four dwords wide, the entire burst is read with one RAS#/CAS#/MDS# memory cycle. In this case, once the data from memory has been latched into the 82353 Data Path, each of the four dwords will be sequentially routed to the host and the host PST returns a BRDY# to the CPU on four sequential CPU clocks.

15.7 Host to Memory Write

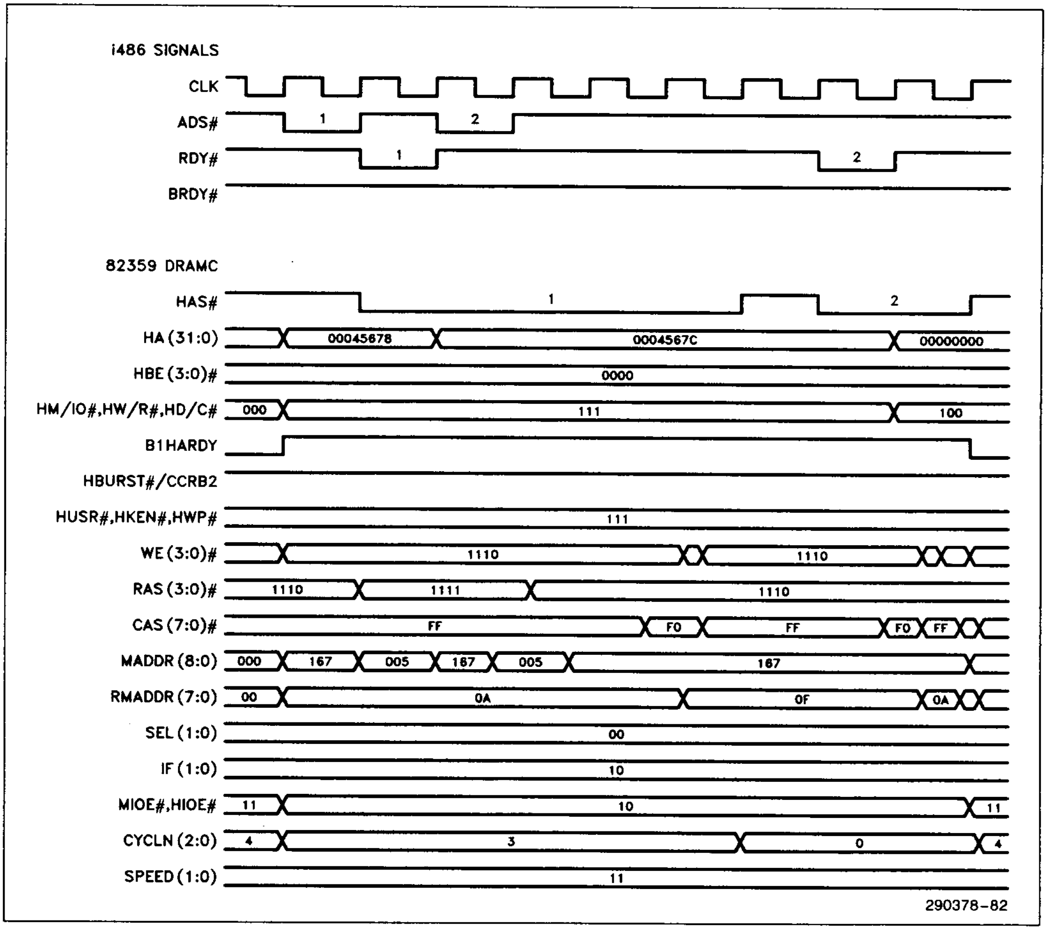


Figure 15-7. Host to Memory Write

The above diagram shows a host-to-memory write cycle. In this example the write is posted (a feature of the PST), allowing the PST to return RDY# to the processor immediately. Even though the processor cycle has ended, the 82359 continues the DRAM write cycle and the PST should keep HAS# asserted until the returned CYCLN(2:0) has expired.

During the DRAM cycle, all columns of the memory array receive the RAS#/CAS# cycle. The 82359 selects the appropriate column to be written by asserting the column's WE#. Since each CAS# is connected to one byte of the dword, all CAS# lines are asserted to allow all of the bytes of dword to be written. Since only one dword has its WE# asserted, only that dword is written and the other dwords of the row are read. This does not cause a data contention problem since the WE#s are typically connected to the 82353 Data Path's memory output enables and thus dwords being read are not driven by the Data Path device.

In the above example the second cycle is also a write. It is the PST's responsibility to hold all cycles to the 82359 off until the previous write cycle has completed. Once the PST counts down from the CYCLN(2:0) of the first write cycle, it will de-assert HAS# and begin the second cycle it had been holding off.

15.8 Host to Memory Write—Single Byte

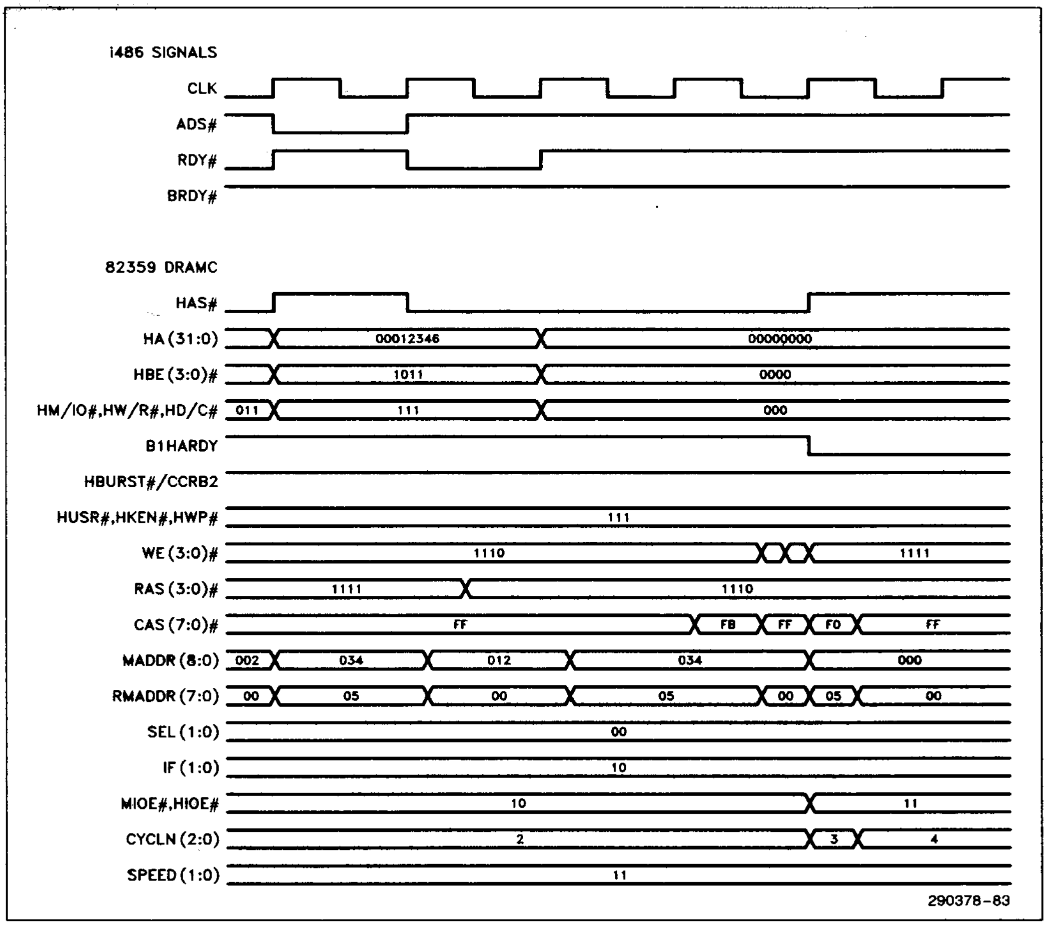


Figure 15-8. Host to Memory Write—Single Byte

The figure above illustrates the case of writing a single byte to memory. The cycle is very similar to writing a dword except for one point. Unlike the dword write which asserts all CAS# signals for the row, writing a byte (or word) causes the 82359 to assert only the CAS# signals for the byte (or word) being written. The remaining CAS# signals are inactive and cause those bytes to remain inactive also.

15.9 Host to System Single Dword Cycle

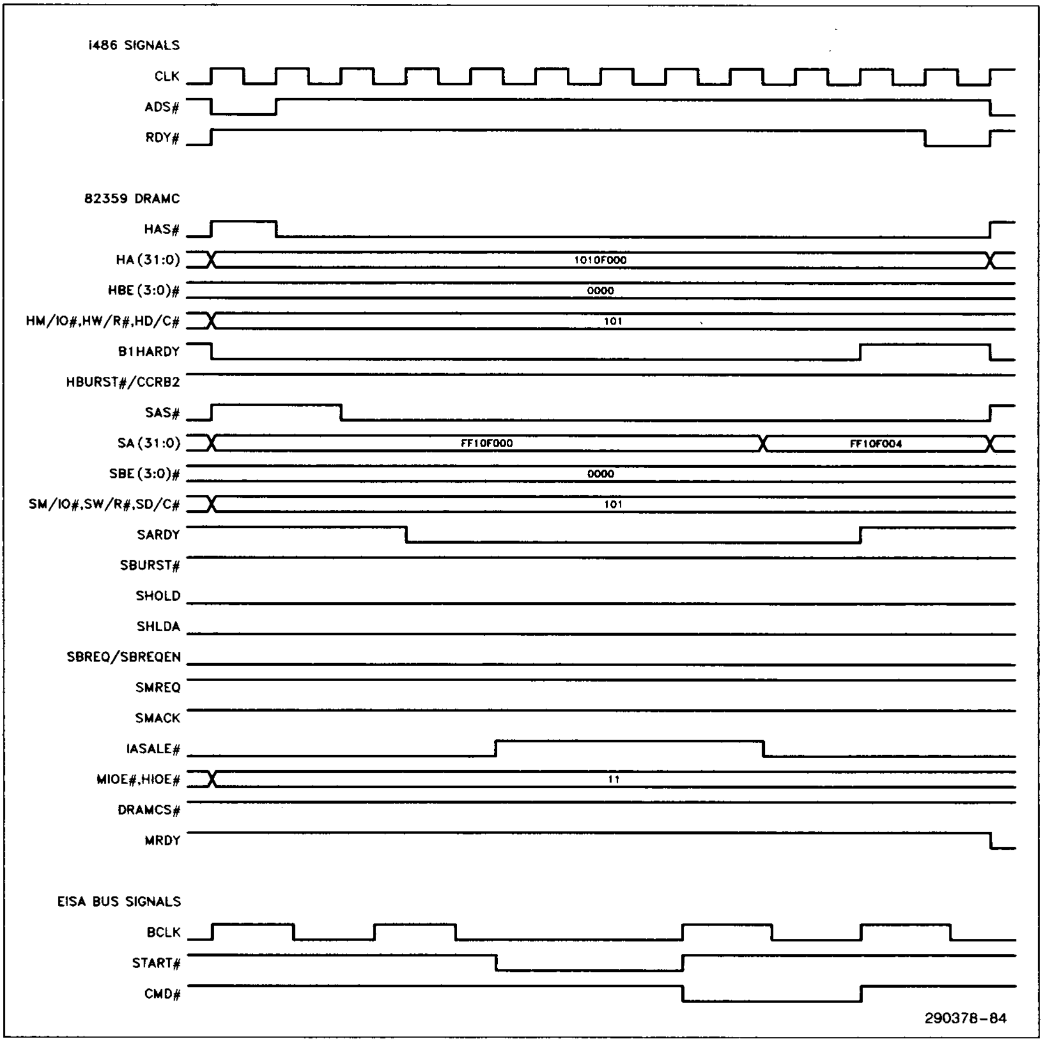


Figure 15-9. Host to System Single Dword Cycle

From the host address, the 82359 determines that the current cycle is not to main memory because the address is not in the address range under its control. It then de-asserts HARDY# telling the host PST that the cycle is non-deterministic and then the 82359 passes the cycle on to the system bus. The figure above illustrates the case where the EISA bus is not in use at the current time. The 82359 knows this because EISA activity can only occur when the 82359 asserts SHLDA back to the ISP, and in this particular example it is de-asserted. Because the EISA bus is currently free, the 82359 will begin the system cycle without going through EISA arbitration.

The sequence for the system cycle is the 82359 forwarding the cycle to the EBC by driving the address, status and asserting SAS#. The EBC translates the cycle to EISA bus protocol and asserts SARDY back to the

82359 until the EISA cycle has completed. Once the EISA read cycle is complete, SARDY will go high causing the 82359 to drive HARDY high, indicating to the host PST that the non-deterministic cycle is complete. Since this is a single dword cycle as opposed to a burst, the host PST de-asserts its HAS# to end the host cycle. This in turn causes the SAS# to be de-asserted ending the system cycle.

15.10 Host Burst to System Cycle

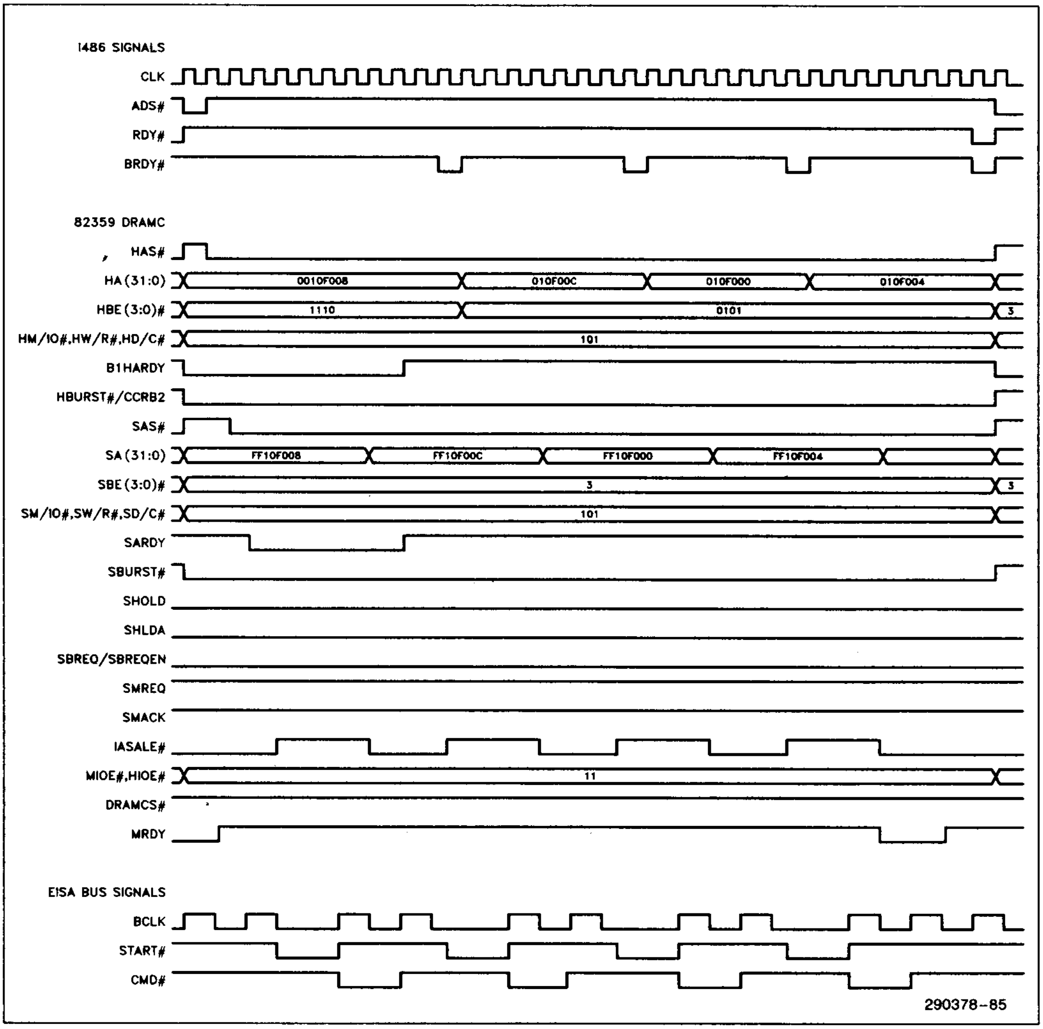


Figure 15-10. Host Burst to System Cycle

The host burst to system cycle is very similar to the single dword cycle presented earlier. Like a single dword transfer of the previous diagram, HARDY and SARDY are de-asserted. But in this case, the SBURST# signal is driven to the 82359 from the EBC, indicating the system slave is capable of bursting. This causes the 82359 to assert HBURST# to the PST to indicate that the cycle is capable of a burst transfer. The rest of the lead-off cycle of the burst is exactly the same as the single dword cycle.

Once the lead-off cycle of the burst is complete, the EBC asserts SARDY causing HARDY to rise, indicating to the host PST that the lead-off cycle of the burst is complete. Unlike a single dword read, the host PST keeps HAS# asserted to continue with the burst and provides the next address on the host bus.

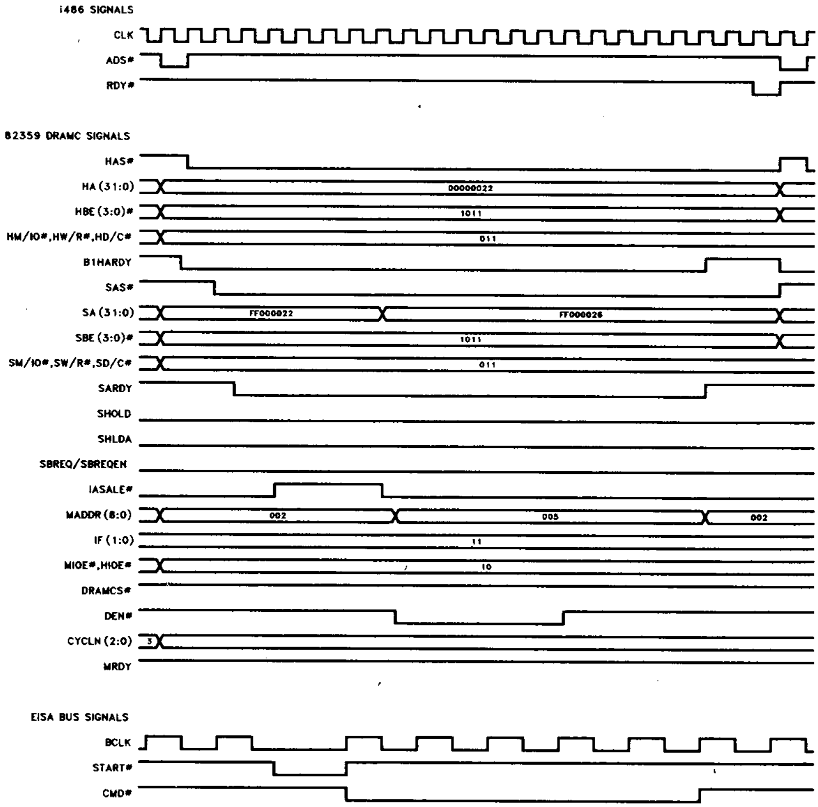
Notice that the system address has already changed to the next address of the burst before the host address. This is due to the action of IASALE#. Recall that the first rising edge of IASALE# of the cycle causes the system address latch to capture the host-provided address.

All subsequent falling edges manipulate SA(5:2) to follow the i486-like burst order and since the EBC causes a rising edge of IASALE# before the host address changes, the next address of the burst appears on the system address lines first. Also, since the SA(5:2) is controlled by the system interface for bursting, bursts of up to 16 dwords can be sustained (although not typically used).

Also notice that even though all transfers of the burst are considered non-deterministic, the 82359 de-asserts HARDY only for the lead-off transfer. The 82353 Data path controls subsequent HARDY generation to the host PST through its own HARDY signal.

The burst transfer ends after the PST has detected a complete burst at which time it de-asserts HAS# to end the host cycle.

15.11 Accessing the 82359's Internal Registers



290378-87

Figure 15-11. Accessing the 82359's Internal Registers

Programming the 82359's internal registers is done via two I/O ports. The host does its first I/O cycle to select the desired internal register's offset and the second I/O cycle to read or write the data. (See Register Programming for more details.)

The I/O cycles must always originate from the host. System cycles which try to access the internal registers are ignored by the 82359. The 82359 treats the register accesses like ordinary I/O cycles, passing them on to the system port. The system bus controller (EBC) then runs the I/O cycle on the EISA/ISA bus as any normal I/O cycle.

When the 82359 detects an access to the Index Register or Data Register (typically I/O port 22h and 23h respectively), it asserts its DEN# output. This is typically connected to a '245 device to allow data transfer between the X-bus and the MADDR(7:0) lines (MADDR8 is not used for register programming). The 82359 tristates its MADDR(7:0) lines for register writes and drives the MADDR(7:0) lines for register reads.

15.12 Snoop Cycle

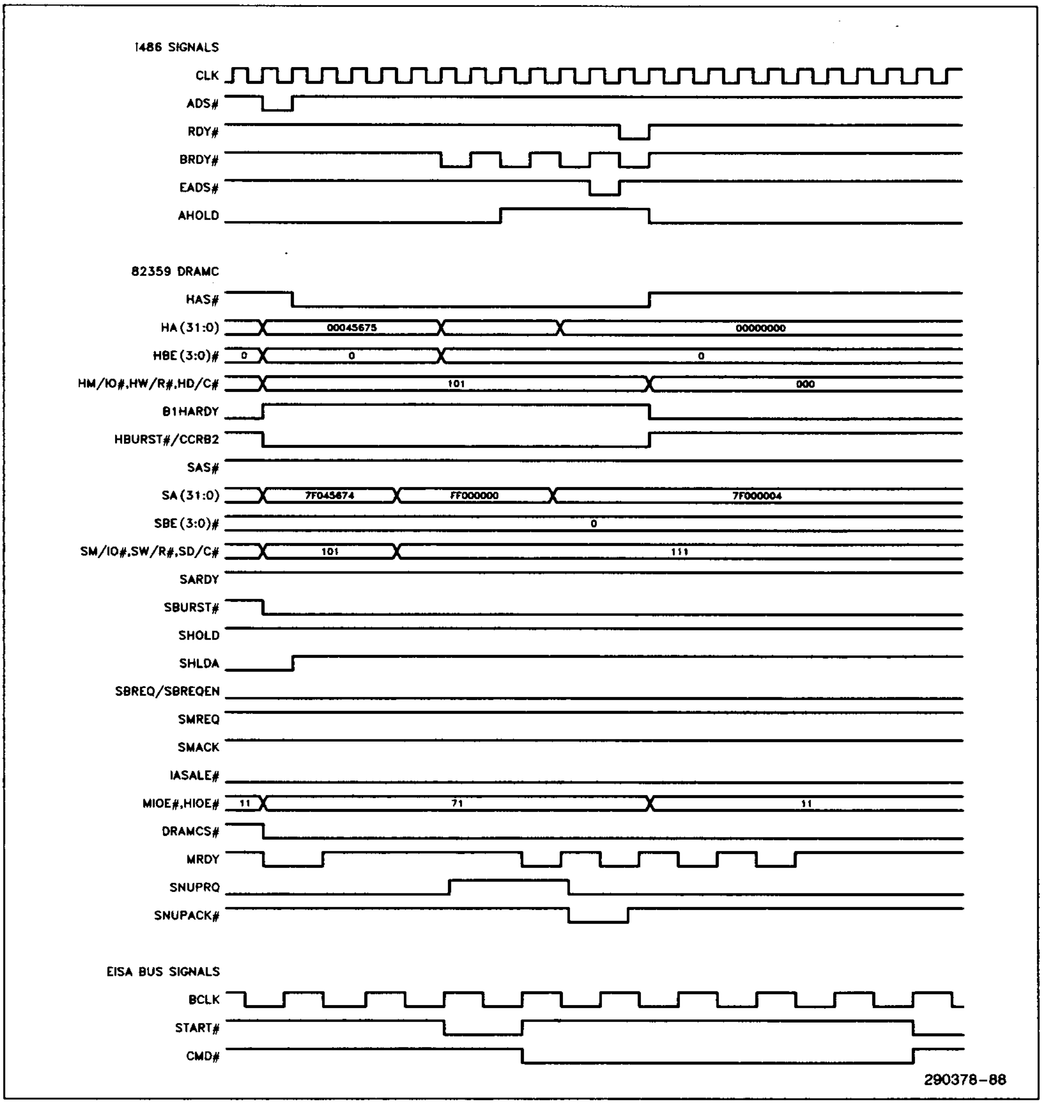


Figure 15-12. Snoop Cycle

The figure above illustrates the 82359 generating a snoop cycle to the host cache. An EISA master executes a memory write cycle to address 00000000h. The 82359 generates a snoop request to the host PST by asserting SNUPRQ. The PST asserts AHOLD to the CPU to tri-state the host address lines. After the host address bus is tri-stated, the host PST returns SNUPACK# to the 82359 causing the 82359 to drive the snoop address onto the host address bus and to de-assert SNUPRQ. The PST finishes the snoop cycle by asserting EADS# causing the cache line to be invalidated, and also de-asserts AHOLD and SNUPACK. (The diagram is specific to the i486 processor which allows for cache invalidation cycles after the lead-off cycle of a burst without affecting the remaining cycles.

15.13 Snoop Filter Example

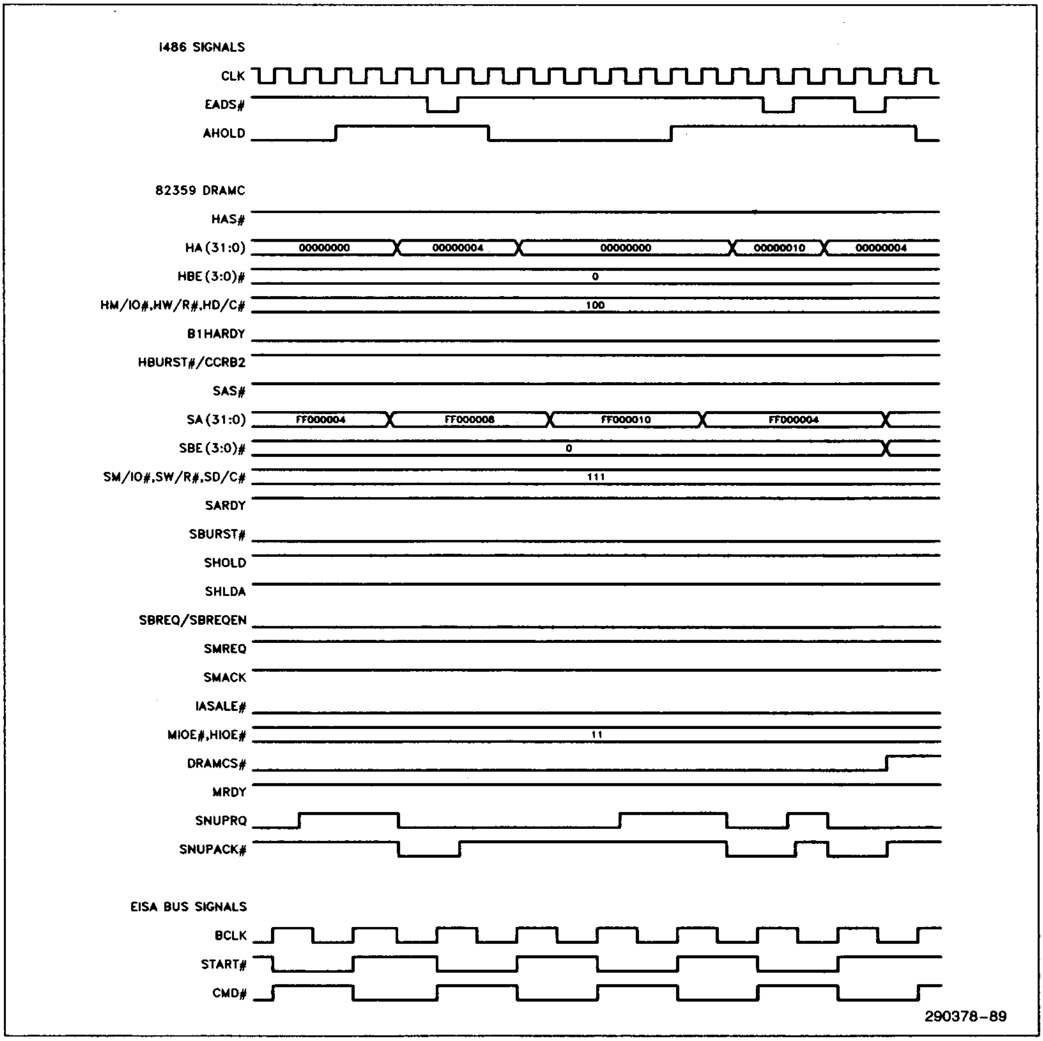


Figure 15-13. Snoop Filter Example

To illustrate the snoop filtering implemented by the 82359, an EISA master is shown executing a series of back-to-back memory writes. The first EISA cycle is to address 00000004h which causes a cache invalidation cycle (assuming the previous snoop was to a different cache line). The second EISA cycle is to address 00000008h, and assuming a 16-byte cache line, will be located on the same cache line as the previously invalidated snoop cycle; thus the 82359 "filters out" this snoop by not propagating it to the host cache. The third EISA cycle is to address 00000010h, a different cache line than the previous cache invalidation cycle. This causes a snoop cycle to be generated. The fourth cycle is to address 00000004h. Although this line had been invalidated earlier, a snoop will occur due to the fact that a different cache line was invalidated in the previous snoop cycle.

16.0 ELECTRICAL CHARACTERISTICS

16.1 Absolute Maximums

Case Temperature under Bias ... −65°C to +110°C

Storage Temperature .... −65°C to +150°C

Voltage on Any Pin with
Respect to Ground .... −0.5V to V_{CC} + 0.5V

Supply Voltage with
Respect to V_{SS} .... −0.5V to +6.5V

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

******WARNING: Stressing the device beyond the "Absolute Maximum Ratings" may cause permanent damage. These are stress ratings only. Operation beyond the "Operating Conditions" is not recommended and extended exposure beyond the "Operating Conditions" may affect device reliability.*

The following table is a stress rating only, and functional operation at the maximums is not guaranteed. Functional operating conditions are given in the D.C. and A.C. Specifications.

Extended exposure to the maximum ratings may affect device reliability. Furthermore, although the 82359 DRAM controller contains protective circuitry to resist damage from static electric discharge, always take precautions to avoid high static voltages or electric fields.

16.2 D.C. Specifications (T_C = 0°C to +85°C, V_{CC} = 5V ± 5%)

Symbol	Parameter	Limits			Units	Test Conditions
		Min	Typ	Max		
V_{IL}	Input Low Voltage	−0.5		0.8	V	(Note 5)
V_{IH}	Input High Voltage	2.0		V_{CC} + 0.5	V	(Note 5)
V_{OL1}	Output Low Voltage			0.45	V	I_{OL} = 5 mA (Note 2)
V_{OH1}	Output High Voltage	2.4			V	I_{OH} = −1 mA (Note 2)
V_{OL2}	Output Low Voltage			0.45	V	I_{OL} = 24 mA (Note 1)
V_{OH2}	Output High Voltage	2.4			V	I_{OH} = −4 mA (Note 1)
I_{LI}	Input Leakage Current			±15	μA	$0 < V_{IN} < V_{CC}$
I_{LO}	Output Leakage Current			±15	μA	$0.45 < V_{IN} < V_{CC}$
I_{CC}	Supply Current		70	100	mA	(Note 3)
			132	200	mA	(Note 4)
C_{IN}	Clock		8.5	10	pF	
	Input		10	13	pF	
	I/O		14	16	pF	

NOTES:

1. V_{OL2} and V_{OH2} apply only to SA(31:2), SM/IO#, SD/C#, SW/R#, SLOCK#, MRDY, and CAS(7:0)#.
2. V_{OL1} and V_{OH1} applies to those pins not specififed in Note 1.
3. OSC = 40 MHz, no external loads. This specification is derived from device characterization data, and is not tested.
4. OSC = 40 MHz. This specification is tested in a dynamic environment with maximum DC loads (I_{OL}, I_{OH}) applied to each pin.
5. Several input signals are sensitive to noise in the threshold region (between V_{IL} and V_{IH}). Multiple transitions through the threshold level could result in false triggering of the internal state machines or delay lines which may result in system failure. Care should be exercised that the following signals have monotonic edges during transitions: BCLK; IORC#; MRDC#; MWTC#; NPERROR#; REFRESH#; SARDY; SAS#; SNUPACK#; START#. It is strongly recommended that the REFRESH# input be buffered with a Schmidt trigger device when using the 82359 in concurrent mode with decoupled refresh.

16.3 A.C. Specifications

The A.C. specifications given in the following table consist of output delays and setup and hold requirements. The A.C. diagrams' purpose is to illustrate the specific clock edges and specific signal edges from which the timing parameters are measured. The reader should not infer any other timing relationships. For specific information on timing relationships, refer to the appropriate functional timing diagram or pin description.

Many of the A.C. specifications depend on the values programmed into the Programmable Timing Registers. These timing register dependent specifications are presented as a constant plus one or more of the register fields. A brief summary of the Programmable Timing Registers along with each register's associated symbol used in the A.C. timing specifications follows:

Symbol	Field Name	Min	Max	Unit
HP1	HAS# Delay	2.61	18.27	ns
SP1	SAS# Delay	7.83	20.88	ns
P2	RAS# Precharge	44.37	80.91	ns
P3	Row Address Hold	33.93	62.64	ns
P4	RAS# to CAS# Delay	62.64	93.96	ns
P5	Column Address Hold	10.44	33.93	ns
P6	CAS# to MDS#	31.32	62.64	ns
P7	CAS# Recycle Time	52.20	83.52	ns
P8	CAS# Low for Reads	41.76	67.86	ns
P9	CAS# Low for Writes	20.88	46.98	ns
HP10	Host Write Data Setup	31.32	52.20	ns
SP10	System Write Data Setup	7.83	20.88	ns
HP11	Host to System Delay	41.76	67.86	ns

A.C. timings are also dependent on which configuration mode the 82359 is operating under. Either Buffered Mode or Standard Mode is selected via the programmable timing register System Cycle Start Delay, bit 7. These timing parameters differ from the programmable timings in that these are fixed values, dependent only upon the mode of operation selected. These parameters appear in the A.C. timing specification as "K" symbols listed below:

Symbol	Description	Standard Mode	Buffered Mode	Unit
K1	START# Active to Cycle Trigger Point (EISA Standard Cycle)	20.88	39.15	ns
K2	BCLK Rising to Cycle Trigger Point (EISA Burst Read Cycles)	10.44	26.10	ns
K3	BCLK Falling to Cycle Trigger Point (EISA Burst Write Cycles)	13.05	26.10	ns
K6	MRDC# Active to Cycle Trigger Point (ISA Read Cycles)	7.83	7.83	ns
K7	MWTC# Active to Cycle Trigger Point (ISA Write Cycles)	26.10	26.10	ns
K8	BCLK Rising where START# is Sampled Active to Cycle Trigger Point (EISA Standard Write Cycles)	23.49	41.76	ns
K9	REFRESH# Active to Cycle Trigger Point (Refresh Cycle)	26.10	26.10	ns

16.3 A.C. Specifications (Continued)

Symbol	Parameter	Min	Max	Units	Note	Figure #
CLOCK AND RESET						
t1	OSC Period	25	25	ns	14	1
t2	OSC High Time	6		ns		1
t3	OSC Low Time	6		ns		1
t4	OSC Fall Time		4	ns	8	1
t5	OSC Rise Time		4	ns	8	1
t6	RESET Pulse Width	20		μs	15	1
SLAVE PORT INTERFACE						
t10	CMD# Pulse Width during Slave Access	500		ns	5	22, 23
t11	DEN# Pulse Width (Write)	200	300	ns	6	22
t14	DEN# Delay from CMD# Active	50	150	ns		22, 23
t14A	DEN# Delay from CMD# Inactive (Read)	3	40	ns		23
t15	Write Data Setup to DEN# Rising	150		ns		22
t16	Write Data Hold from DEN# Rising	0		ns		22
t17	MADDR(7:0) Float		30	ns	8	22
t18	MADDR(7:0) Memory Address Redrive from CMD# Inactive		32	ns		22
t19	Output Port Switch from DEN# Rising (SPEED(1:0), SBREQEN, FREF#, LOCKEN#, CCRB2)		40	ns	9	22
t20	Read Data Hold from CMD# Inactive	0		ns		23
t21	CMD# Active to Read Data Valid		300	ns		23
t22	DEN# Delay from SAS#	−4	40	ns	29	24
HOST MEMORY INTERFACE						
t30	HA(31:2) Valid to MADDR(8:0), RMADDR(7:0) Valid (Page Hit)		20	ns	17	4
t31	HA(31:2), PCD, HBE(3:0)#, HM/IO#, HD/C#, HW/R#, HLOCK# Setup to HAS# Active (for latching)	5		ns	3	2, 3
t31A	HA(31:2), PCD, HM/IO#, HD/C#, HW/R#, HLOCK# Setup to HAS# Active (for decoding)	20		ns	4	3
t32	HA(31:2), PCD, HBE(3:0)#, HM/IO#, HD/C, HW/R#, HLOCK# Hold from HAS# Active	13		ns		2, 3
t32A	HLOCK# Hold from HAS# Active	78		ns	50	2, 3
t33	HA(31:2) Valid to SEL(1:0) Valid (Page Hit)		41	ns	17	3
t33A	HAS# Active to SEL(1:0) Valid (Page Miss)		85+HP1	ns	1	3
t33B	HA(31:2) Valid to SEL(1:0) Valid (Row Miss)		65	ns	17	3
t34	HA(31:2) Valid to IF(1:0) Valid (Page Hit)		41	ns	17	3
t34A	HAS# Active to IF(1:0) Valid (Page Miss)		85+HP1	ns	1	3
t34B	HA(31:2) Valid to IF(1:0) Valid (Row Miss)		65	ns	17	3

16.3 A.C. Specifications (Continued)

Symbol	Parameter	Min	Max	Units	Note	Figure #
HOST MEMORY INTERFACE (Continued)						
t35	HA(31:2), HW/R# Valid to HIOE# Valid		35	ns	17, 25	3
t35A	HA(31:2), HW/R# Valid to MIOE# Valid		70	ns	17, 25	3
t36	HA(31:2), HW/R# Valid to WE(3:0)# Valid (Page Hit, Row Miss)		34	ns		4
t36B	HAS# Active to WE(3:0)# Valid (Page Miss)		70 + HP1	ns	1	5
t37	HIOE#, MIOE# Hold from HAS# Inactive	0		ns		3
t38	HAS# Active to CAS# Active (Read Page Hit)	4 + HP1	22 + HP1	ns	1, 36	4
t38A	HAS# Active to CAS# Active (Row Miss)		31 + P4 + HP1	ns	1, 36	6
t38B	HAS# Active to CAS# Active (Page Miss)		33 + P2 + P4 + HP1	ns	1, 36	5
t38C	HAS# Active to CAS# Active (Write Page Hit)	4 + HP10	23 + HP10	ns	1, 36	4
t39A	HAS# Active to RAS# Active (Row Miss)		42 + HP1	ns	1	6
t39B	HAS# Active to RAS# Active (Page Miss)		47 + P2 + HP1	ns	1	5
t42A	HAS# Active to Column Address Valid (Row Miss)		38 + P3 + HP1	ns	1	6
t42B	HAS# Active to Column Address Valid (Page Miss)		46 + P2 + P3 + HP1	ns	1	5
t44	HAS# Active to MDS# Inactive (Read Page Hit)	P6 + HP1	30 + P6 + HP1	ns	1	4
t44A	HAS# Active to MDS# Inactive (Read Row Miss)	2 + P4 + P6 + HP1	40 + P4 + P6 + HP1	ns	1	6
t44B	HAS# Active to MDS# Inactive (Read Page Miss)	P2 + P4 + P6 + HP1	48 + P2 + P4 + P6 + HP1	ns	1	5
t45	HAS# Active to CAS# Inactive (Read Page Hit)		27 + P8 + HP1	ns	1, 36	4
t45A	HAS# Active to CAS# Inactive (Read Row Miss)		37 + P4 + P8 + HP1	ns	1, 36	6
t45B	HAS# Active to CAS# Inactive (Read Page Miss)		41 + P2 + P4 + P8 + HP1	ns	1, 36	5
t45C	HAS# Active to CAS# Inactive (Write Page Hit)		23 + P9 + HP10	ns	1, 36	4
t45D	HAS# Active to CAS# Inactive (Write Row Miss)		28 + P4 + P9 + HP1	ns	1, 36	6

16.3 A.C. Specifications (Continued)

Symbol	Parameter	Min	Max	Units	Note	Figure #
HOST MEMORY INTERFACE (Continued)						
t45E	HAS# Active to CAS# Inactive (Write Page Miss)		32.5 + P2 + P4 + P9 + HP1	ns	1, 36	5
t48A	HAS# Active to RAS# Inactive (Row Miss)	30 + P4 + HP1	70 + P4 + HP1	ns	1, 16	6
t48B	HAS# Active to RAS# Inactive (Page Miss)	30 + P2 + P4 + HP1	75 + P2 + P4 + HP1	ns	1, 16	5
MEMORY INTERFACE RELATIVE OUTPUT SPECIFICATIONS						
t60	CAS# Pulse Width (Read)	P8 − 7		ns		4, 5, 14
t60A	CAS# Pulse Width (Write)	P9 − 6		ns		4, 5, 14
t61	RAS# Pulse Width (Non-Page Mode)	P4 + 7	P4 + 40	ns	46	5
t62	RAS# Active to CAS# Active	P4 − 26		ns	36	5
t63	RAS# Precharge Time	P2 + 5		ns		5
t64	CAS# Active Hold from MDS# Inactive	P8 − P6 − 5		ns	36	5, 7, 14
t65	CAS# Active to MDS# Inactive	P6 − 14		ns		4, 7, 14
t66	CAS# Active to Next CAS# Active (Read Burst)	P7 − 10	P7 + 9.5	ns		7
t67	CAS# Precharge Time	P7 − P8 − 5.5		ns		7, 15
t68	Column Address Setup Time (Row or Page Miss)	P4 − P3 − 12		ns	36	5, 14
t68A	Column Address Setup Time (Read Burst for 2nd, 3rd, or 4th CAS#)	P7 − P5 − 25		ns	36	7
t69	Column Address Hold Time	P5 − 13		ns		4, 5, 7, 14
t69A	CAS# Falling to Next Column Address Valid		P5 + 25	ns	43	7
t70	Row Address Setup Time	5		ns		5, 6, 14
t71	Row Address Hold Time	P3 − 21.5		ns	48	5, 6, 14
t73	WE(3:0)# Hold Time from CAS# Falling	P5 − 13		ns		4, 5, 14, 15
t74	RAS# Active to MDS# Inactive	P4 + P6 − 27		ns		5
t75	Column Address Valid to MDS# Inactive (Bank or Page Miss)	P4 + P6 − P3 − 12		ns		5
t76	CAS# Active to MDS# Active		16	ns		4, 14
t77A	SEL(1:0), IF(1:0) Hold from MDS# Inactive (Read)	5		ns	33	15
t77B	SEL(1:0), IF(1:0) Hold from CAS# Active (Write)	P5 + 0		ns	33	15

16.3 A.C. Specifications (Continued)

Symbol	Parameter	Min	Max	Units	Note	Figure #
SYSTEM MEMORY INTERFACE						
t90	SA(31:2) Valid to MADDR(8:0), RMADDR(7:0) Valid (Page Hit)		36	ns		11
t91	SA(31:2), SM/IO#, SW/R#, SD/C#, SBE(3:0)# Setup to SAS# Active	15		ns		9, 10
t92	SA(31:2), SM/IO#, SW/R#, SD/C#, SBE(3:0)# Hold from SAS# Active	65		ns		9, 10
t93	SA(31:2) Valid to SEL(1:0) Valid (Page Hit)		60	ns	45	10, 15
t93A	SAS# Active Active to SEL(1:0) Valid (Page Miss)		90 + SP1	ns	2	10
t93B	SA(31:2) Valid to SEL(1:0) Valid (Row Miss)		70	ns	45	10
t94	SA(31:2) Valid to IF(1:0) Valid (Page Hit)		60	ns	45	10
t94A	SAS# Active to IF(1:0) Valid (Page Miss)		90 + SP1	ns	2	10
t94B	SA(31:2) Valid to IF(1:0) Valid (Row Miss)		70	ns	45	10
t95	SA(31:2), SW/R# Valid to MIOE# Valid		45	ns	25, 42	10, 14
t96	SA(31:2), SW/R# Valid to WE(3:0)# Valid (Page Hit, Row Miss)		65	ns		11, 13
t96B	SAS# Active to WE(3:0)# Valid (Page Miss)		95 + SP1	ns	2	12
t98	SAS# Active to CAS# Active (Read Page Hit)	4 + SP1	50 + SP1	ns	2, 36	11
t98A	SAS# Active to CAS# Active (Row Miss)		60 + P4 + SP1	ns	2	13
t98B	SAS# Active to CAS# Active (Page Miss)		70 + P2 + P4 + SP1	ns	2	12
t98C	SAS# Active to CAS# Active (Write Page Hit)	4 + SP10	50 + SP10	ns	2, 36	11
t99A	SAS# Active to RAS# Active (Row Miss)		65 + SP1	ns	2	13
t99B	SAS# Active to RAS# Active (Page Miss)		75 + P2 + SP1	ns	2	12
t101A	SAS# Active to Column Address Valid (Row Miss)		60 + P3 + SP1	ns	2	13
t101B	SAS# Active to Column Address Valid (Page Miss)		70 + P2 + P3 + SP1	ns	2	12
t105C	SAS# Active to CAS# Inactive (Write Page Hit)		50 + P9 + SP10	ns	2	11
t105D	SAS# Active to CAS# Inactive (Write Row Miss)		60 + P4 + P9 + SP1	ns	2	13
t105E	SAS# Active to CAS# Inactive (Write Page Miss)		70 + P2 + P4 + P9 + SP1	ns	2	12

16.3 A.C. Specifications (Continued)

Symbol	Parameter	Min	Max	Units	Note	Figure #
SYSTEM MEMORY INTERFACE (Continued)						
t108A	SAS# Active to RAS# Inactive (Row Miss)		75+P4+P6+SP1	ns	2, 16	13
t108B	SAS# Active to RAS#		85+P2+P4 +P6+SP1	ns	2, 16	12
ISA MASTER MEMORY INTERFACE						
t110	SA(31:2) Setup to MRDC#, MWTC# Active	40		ns		17
t111	SA(31:2) Hold from MRDC#, MWTC# Inactive	0		ns		17
t112	SBE(3:0)# Inputs Valid from MRDC#, MWTC# Active		10	ns		17
t113	SBE(3:0)# Hold from MRDC#, MWTC# Inactive	0		ns		17
t113A	MIOE# Hold from MRDC#, MWTC# Inactive	0		ns		17
t114	MRDC# Active to RAS# Active	15+K6	60+K6	ns	24	17
t114A	MWTC# Active to RAS# Active	15+K7	65+K7	ns	24	17
t115	MWTC# Active to WE(3:0)# Active		36+K7	ns	24	17
t115A	MRDC#, MWTC# Active to MIOE# Valid		50	ns		17
t116	MRDC# Active to CAS# Active	5+P4+K6	50+P4+K6	ns	24, 36	17
t116A	MWTC# Active to CAS# Active	4+P4+K7	55+P4+K7	ns	24, 36	17
t117	MRDC# Active to Column Address Valid		55+P3+K6	ns	24	17
t117A	MWTC# Active to Column Address Valid		55+P3+K7	ns	24	17
t118	MRDC#, MWTC# Active Time	230		ns		17
t119	MRDC#, MWTC# Inactive Time	110		ns		17
EISA MASTER MEMORY INTERFACE						
t120	SA(31:2), SM/IO# Setup to START# Active	30.88−K1		ns		14
t121	SA(31:2), SM/IO#, SBE(3:0)#, SW/R# Hold from CMD# Active or START# Inactive	15		ns		14
t122	SBE(3:0)# Setup to CMD# Active or START# Inactive	50		ns		14
t123	SW/R# Input Delay from START#		30	ns		14
t125A	START# Active Hold from BCLK Rising	2		ns		14
t126	CMD# Active Time	115		ns		14
t127	SA(31:2), SM/IO# Setup to BCLK Rising (Burst Cycle)	5		ns		15
t127A	SBE(3:0)# Setup to BCLK Rising (Burst Cycles)	−16		ns		15
t128	SA(31:2), SM/IO#, SBE(3:0)#, SW/R# Hold from BCLK Falling (Burst Cycle)	0		ns	21	15
t130	START# Active to RAS# Inactive (Std Read Cycle)		80+P4+K1	ns	22, 28	14
t130A	BCLK Rising to RAS# Inactive (End of Burst Cycle)		60	ns		15, 16
t130B	BCLK Rising to RAS# Inactive (Std Write Cycle)		80+K8	ns	22, 26	14

16.3 A.C. Specifications (Continued)

Symbol	Parameter	Min	Max	Units	Note	Figure #
EISA MASTER MEMORY INTERFACE (Continued)						
t131	START# Active to RAS# Active	18+K1	48+K1	ns	22, 28	14, 15
t132	START# Active to CAS# Active (Std Read Cycle)		38+P4+K1	ns	22, 28, 36	14, 16
t132A	BCLK Rising to CAS# Active (Std Write Cycle)	7+K8	55+K8	ns	22, 26, 36	14, 16
t133	START# Active to Column Address Valid		43+P3+K1	ns	22, 28	14
t134	BCLK Rising to CAS# Active (Burst Read Cycle)	6+K2	38+K2	ns	22, 36	15, 16
t134A	BCLK Falling to CAS# Active (Burst Write Cycle)	6+K3	34+K3	ns	22, 36	15, 16
t135	SA(31:2) Valid to Column Address Valid (Burst Cycle)		24	ns		15
t136	MSBURST# Setup to BCLK Rising	15		ns		15, 16
t137	MSBURST# Hold from BCLK Rising	45		ns		15, 16
t138	SA(31:2), Valid to WE(3:0)# Valid (Std Write Cycle)		90	ns		14, 15
t138A	BCLK Falling to Next WE(3:0)# Valid (Burst Write Cycle)		60+P5+K3	ns	37	15
t138B	SW/R# Valid to WE(3:0)# Valid		45	ns		14
t138C	SA(31:2) Valid to WE(3:0)# Valid (Burst Write Cycle)		42	ns	34	15
t139	BCLK Rising to MRDY Output Low		35	ns		16
t140	BCLK Falling to MRDY Output Float	2	40	ns	8	16
t141	BCLK High Time	55		ns		15
t142	BCLK Low Time	55		ns		15
t143	BCLK Period	120		ns		15
t144	MIOE# Hold from CMD# Inactive	0		ns		14
t145	SLOCK# Input Setup to BCLK Rising	55		ns		14
t146	SLOCK# Input Hold from BCLK Rising	2		ns		14
t147	MRDY Input Setup to BCLK Falling	15		ns		
t148	MRDY Input Hold from BCLK Falling	5		ns		
t149	MIOE# Hold from BCLK Rising (Burst Cycles)	0		ns		15
HOST STROBED BUS HANDSHAKE INTERFACE						
t180	HAS# Active Time	75		ns	39	2
t181	HAS# Inactive Time	12		ns		2
t183	HKEN#, HWP#, HUSR# Valid from HA(31:2), HM/IO#, HD/C#, HW/R#, HLOCK#, PCD Valid		33	ns	25	2
t184	CYCLN(2:0) Valid from HA(31:2), PCD, HM/IO#, HD/C#, HW/R# Valid (Deterministic Cycles)		74	ns	11, 25	2
t184A	HBURST# Valid from HA(31:2), PCD, HM/IO#, HD/C#, HW/R#, HLOCK# Valid		43	ns	11, 25	2
t184B	PAGEHIT# Valid from HA(31:2), PCD, HM/IO#, HD/C#, HW/R# Valid		33	ns	11, 25	2
t185	HARDY Valid from HA(31:2), PCD, HM/IO#, HD/C#, HW/R#, HLOCK# Valid		33	ns	25	2

16.3 A.C. Specifications (Continued)

Symbol	Parameter	Min	Max	Units	Note	Figure #
HOST STROBED BUS HANDSHAKE INTERFACE (Continued)						
t186	CYCLN(2:0), PAGEHIT# Valid from HMACK Inactive		25	ns	12	18
t188	HMREQ Falling to HMACK Falling	20		ns	49	18

t191	IF(1:0), HBURST#, HKEN#, HWP#, HUSR# Hold from HAS# Inactive
	CYCLN(2:0), PAGEHIT#, SEL(1:0) Hold from HAS# Inactive (Page Hits)
		2		ns		2

t191A	CYCLN(2:0), PAGEHIT#, SEL(1:0) Hold from HAS# Falling (Read Row Miss, Page Miss)	P4 + P6 + HP1 + 9		ns		3
t191B	CYCLN(2:0), PAGEHIT#, SEL(1:0) Hold from HAS# Falling (Write Row Miss, Page Miss)	P4 + P9 + HP1 + 9		ns		3
t192	HMREQ Rising to HMACK Rising		1.5	μs	13	18
t193	HAS# Rising from HMACK Rising		15	ns		18
t200	SAS# Active Time	75		ns	39	9
t201	SAS# Inactive Time	15		ns		9
t202	SAS# Cycle Time	100		ns	38	9
t204	CYCLN(2:0) Valid from SA(31:2), SM/IO#, SD/C#, SW/R# Valid		70	ns	11, 25	9
t204B	PAGEHIT# Valid from SA(31:2), SM/IO#, SD/C#, SW/R# Valid		60	ns	11, 25	9
t205	MRDY Valid from SAS# Active		90	ns	11	9
t206	CYCLN(2:0), PAGEHIT# Valid from SMACK Inactive		25	ns	12	19
t208	SMREQ Inactive to SMACK Inactive	20		ns	49	19
t210	MRDY Output Low Time	10		ns		9

t211	IF(1:0) Hold from SAS# Inactive
	CYCLN(2:0), PAGEHIT#, SEL(1:0) Hold from SAS# Inactive (Page Hits)
		2				9, 10

t211A	CYCLN(2:0), PAGEHIT#, SEL(1:0) Hold from SAS# Active (Read Row Miss, Page Miss)	P4 + P6 + SP1 + 6		ns		10
t211B	CYCLN(2:0), PAGEHIT#, SEL(1:0) Hold from SAS# Active (Write Row Miss, Page Miss)	P4 + P9 + SP1 + 6		ns		10
t213	SAS# Rising from SMACK Rising		15	ns		19
t215	SMACK High to Next START#/MRDC#/MWTC#, REFRESH# Active	120		ns	40	
t216	SBURST# Input Delay from SAS# Active		5	ns		9
SNOOP INTERFACE						
t220	SNUPRQ Active to SNUPACK# Active	23		ns		21
t221	SNUPACK# Active to SNUPRQ Negated	0	31	ns		21
t222	SNUPACK# Active to HA(31:2) Valid	0	29	ns		21
t223	SNUPACK# Negated to HA(31:2) Float		15	ns	8	21
t224	SNUPACK# Cycle Time	60		ns		21
t225	SNUPACK# Active Time	40		ns		21

16.3 A.C. Specifications (Continued)

Symbol	Parameter	Min	Max	Units	Note	Figure #
MISCELLANEOUS SYSTEM INTERFACE CONTROL						
t240a	HA[31:2], HBE[3:0], HMIO#, HW/R#, HD/C#, HLOCK# Valid to SA[31:2], SBE[3:0], SM/IO#, SW/R#, SD/C#, SLOCK#, SBURST#, Valid		45	ns		8
t240b	HAS# Rising to SA[31:2], SBE[3:0], SM/IO#, SW/R#, SD/C#, SLOCK#, SBURST# Valid		45	ns		8
t241	HAS# Active to SBE(3:0), SD/C#, SW/R#, SBURST# Valid		60	ns	1, 18	8
t241A	HAS# Active to SLOCK# Valid		80	ns	18	8
t242	HAS# Active to SAS# Active	2+HP11	35+HP11	ns	1, 18, 19	8
t243	HAS# Inactive to SAS# Inactive	2	40	ns		8
t244	ST# Valid from HAS# Active		20	ns		8
t244A	ST# Hold from HAS# Inactive	−10		ns		8
t245	HAS# Active to SBREQ Active		85	ns	8	20
t247	SA(31:2), SM/IO# Valid to DRAMCS#		45	ns		8
t248	SA(31:2), SM/IO# Valid to BIOSCS#, VGAMSL#		40	ns		8
t249	SARDY Rising to HARDY Rising	0	30	ns		8
t250	IASALE# Active to SA(31:6), SM/IO#, MADE24 Valid	3	45	ns		8
t251	IASALE# Active to SA(5:2) Valid	0	20	ns		8
t252	IASALE# High Time	60		ns		8
t253	IASALE# Low Time	20		ns		8
t254	PER# Setup to CMD# Inactive (EISA Standard Cycles)	5		ns		14
t254A	PER# Setup to MRDY# Inactive (ISA Cycles)	10		ns		17
t254B	PER# Setup to BCLK Rising (EISA Burst Cycles)	5		ns		15
t255	PER# Hold from CMD# Inactive (EISA Standard)	20		ns		14
t255A	PER# Hold from MRDC# Inactive (ISA Cycles)	20		ns		17
t255B	PER# Hold from BCLK Rising (EISA Burst Cycles)	18		ns		15
t256	PERSTB# Pulse Width	10		ns	32	3, 10
t256A	PERSTB# Input Falling Prior to Next HAS# Falling	0		ns	47	3
t257	SA(31:2), SBE(3:0), SM/IO#, SD/C#, SW/R#, SLOCK#, SBURST#, SAS# Float from SHLDA High	0	20	ns	8	20

16.3 A.C. Specifications (Continued)

Symbol	Parameter	Min	Max	Units	Note	Figure #
MISCELLANEOUS SYSTEM INTERFACE CONTROL						
t258	SA(31:2), SBE(3:0), SM/IO#, SD/C#, SW/R#, SLOCK#, SBURST#, SAS# Valid from SHLDA Low	8.5	60	ns		20
t259	SHLDA Low to SBREQ Low	0	50	ns		20
t260	SAS# Output Active from SHLDA Low	75		ns	7	20
t261	SAS# Active to SARDY Rising	90		ns	30	8
t262	SARDY Input Low Time	20		ns		8
t263	SHOLD High to SHLDA High	75		ns	7	20
t270	HLOCK Hold from HAS# Active	78		ns	50	

NOTES:

1. Applies to the falling edge of HAS# when HMACK is low, or the falling edge of HMACK if HAS# is low.
2. Applies to the falling edge of SAS# when SMACK is low, or the falling edge of SMACK if SAS# is low.
3. Minimum amount of setup time required for latching.
4. Minimum amount of setup time required for decoding (when HP1 is programmed to 2.5). If the actual setup time is less than t31A, then HP1 needs to be programmed to make up the difference.
5. Not tested. The system must meet t10 to insure an adequate write data sample window, and to prevent contention from the time DEN# is de-asserted to the time when the 82359 re-drives the memory address.
6. Functional specification, not tested. Nominally 10 OSC periods wide.
7. Functional specification, not tested. Nominally 4 OSC periods wide.
8. Not tested.
9. This specification applies to any instant when a write to an 82359 slave port can directly cause an output pin to switch. Specifically; HBURST#/CCRB2, SBREQ/SBREQEN, SPEED(1:0), and LOCKEN#.
10. This note has been deleted.
11. Applies only when the falling edge of HAS#/SAS# starts a cycle (HMACK/SMACK is low).
12. Applies only when the falling edge of HMACK/SMACK starts a cycle after a request has been made by activating HAS#/SAS#.
13. Must be met to conform with the EISA bus specification for maximum EXRDY inactive time. This specification only applies for host cycles to main memory when concurrency is enabled. Not tested.
14. OSC should always be generated from a 40 MHz oscillator with tolerance of ±0.01%.
15. OSC must be stable for the duration of t6.
16. Applies only to non-page mode or dynamic page mode cycles.
17. HAS# must be high at least 5 ns before HA(31:2), HM/IO#, HW/R#, and HD/C# become valid to meet these specifications.
18. Applies only after arbitration for the system bus has been completed.
19. In non-concurrent mode, Note 1 applies. However, in concurrent mode, SAS# will be driven after arbitration for the system bus is complete, and is not dependent on the deassertion of HMACK.
20. This specification applies for page miss cycles and cycles that required an internal arbitration of main memory by the 82359.
21. The hold time specification for SW/R#, SM/IO# only applies for the last cycle of a burst.
22. The parameters K1, K2, and K3 are derived from a fixed tap out of the system asynchronous delay line specifically for support of EISA cycles.
23. Note 23 has been deleted.
24. The parameters K6 and K7 are derived from a fixed tap out of the system asynchronous delay line specifically for support of ISA cycles.
25. HAS#/SAS# must be high at least 5 ns prior to the valid time of HA(31:2)/SA(31:2) for this specification to be valid. Since HAS#/SAS# high allows HA(31:2)/SA(31:2) to flow through the 82359, the 5 ns specification is the allotted time for the internal address latch propagation delay.
26. Rising edge of BCLK where START# is sampled active.
27. Active edge of CAS# for write cycles is based on the worst case of t162 or t162B.
28. This specification refers to the falling edge of START# or the rising edge of BCLK where MSBURST# is sampled active at the beginning of the first burst cycle.
29. This specification applies only for 80486 flush cycles. For this specification, the minimum capacitance load on DEN# is assumed to be 15 pF.
30. Not tested. Must be met to guarantee a minimum pulse width on SAS# for 80486 flush cycles.
31. This note has been deleted.
32. Tested as an input only.

NOTES: (Continued)

33. Applies only to ISA or EISA cycles.
34. Applies only after the active CASx# signal(s) from the previous cycle have gone high. WE(3:0)# maximum delay is the slower of 138A and 138C.
35. Note 35 has been deleted.
36. CAS(7:0)# minimum delay specified with $C_L = 50$ pF.
37. WE(3:0)# for the next cycle is gated from the event that causes CAS(7:0)# to go inactive in the current cycle. This specification applies for when WE(3:0)# will be valid for the next EISA burst write cycle.
38. Specifications for HAS#/SAS# cycle times are component requirements for the absolute minimum allowable time for any assertion of HAS#/SAS#. However, back-to-back assertions of HAS#/SAS# for cycles accessing 82359 memory must also satisfy the following requirement: the time from the rising edge of the last CAS# assertion of a memory read or write cycle to the next assertion of HAS#/SAS# must be greater than 20 ns − HP1/SP1. Note that this will cause the actual minimum HAS#/SAS# cycle time to vary from cycle to cycle, depending on the memory organization and type of DRAM access (i.e., page hit/miss or row miss).
39. Specifications t180/t200 are component requirements for the absolute minimum allowable time for assertion of HAS#/SAS#. However, every assertion of HAS#/SAS# for a cycle accessing 82359 memory must also satisfy the following requirement: HAS#/SAS# must remain asserted until the column address hold time is met (as specified by t69) for the final CAS# of a memory write cycle, and until the final MDS# rising for a memory read cycle. Note that this will cause the minimum HAS#/SAS# active time to vary from cycle to cycle, depending on the memory organization and type of DRAM access (i.e., page hit/page miss/row miss, read/write).
40. Minimum time required for the EISA bus arbiter to transfer bus ownership from a system PST master (SMACK rising) to another EISA/ISA bus master and start a cycle (falling edge of START#, MRDC#, MWTC#, or REFRESH#).
41. Note has been deleted.
42. Applies to system PST masters and EISA masters.
43. Applies to host and system PST master burst cycles.
44. This note has been deleted.
45. Applies to system PST masters, EISA and ISA masters.
46. Applies to host and system PST master cycles only.
47. Failing to meet this specification may result in an incorrect address being captured in the parity error trap registers. This spec is not tested.
48. Minimum delay specified with equal loading on RAS and row address.
49. This is a system requirement and is a function of external logic.
50. Applies only in concurrent mode and if HAS# low time is less than 78 ns.

A.C. TEST LOADS

Loading for output delay maximum specs

240 pF:	SA(31:2), SM/IO#, SW/R#, SLOCK#, MRDY
120 pF:	CAS(7:0)#
50 pF:	All others

Loading for output delay minimum specs

50 pF:	CAS(7:0)#
15 pF:	If(1:0), SEL(1:0), MDS#, WE(3:0)#
0 pF:	All others

16.3.1 DELAY LINE TEST METHODOLOGY

To illustrate the testing methodology, the shaded region of the graph below indicates an operational range of a typical A.C. specification that involves a delay line parameter. Testing will guarantee the specification will fall within the "Spec.min" and "Spec.max" range for a given tap programming as shown on the horizontal axis as "n". Although testing cannot guarantee "delta t" for individual tap-to-tap delays, the compensated design of the delay line structure that accounts for process, voltage, and temperature variations results in a fixed slope that is guaranteed to stay within the specified minimum and maximum range.

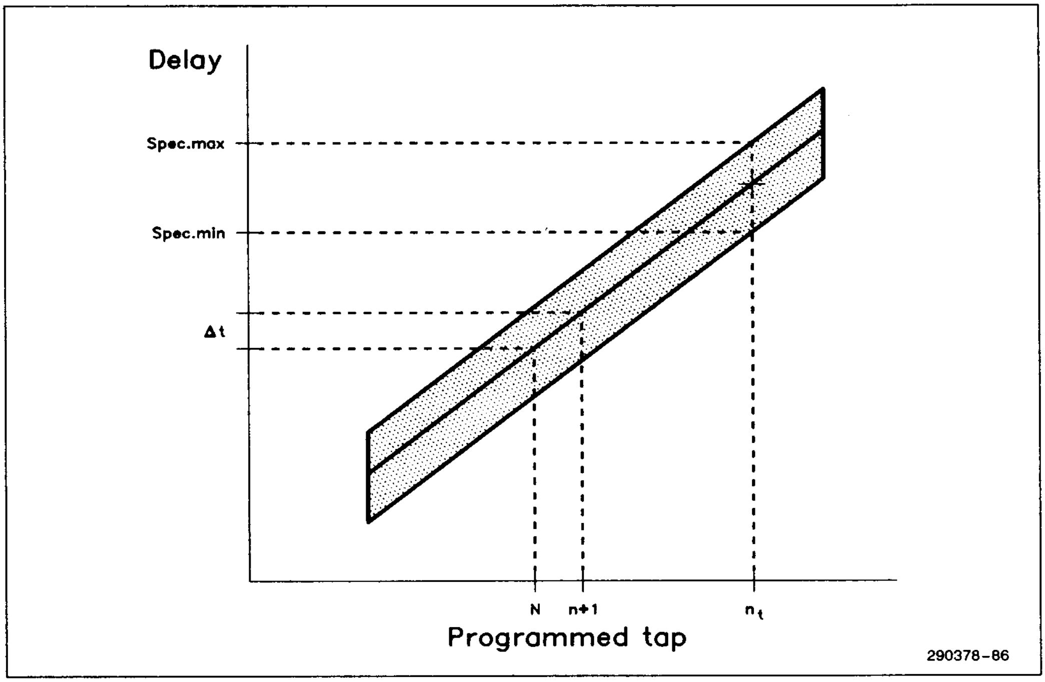


Figure 16-1. Typical Delay Line Based Specification vs Programmed Tap

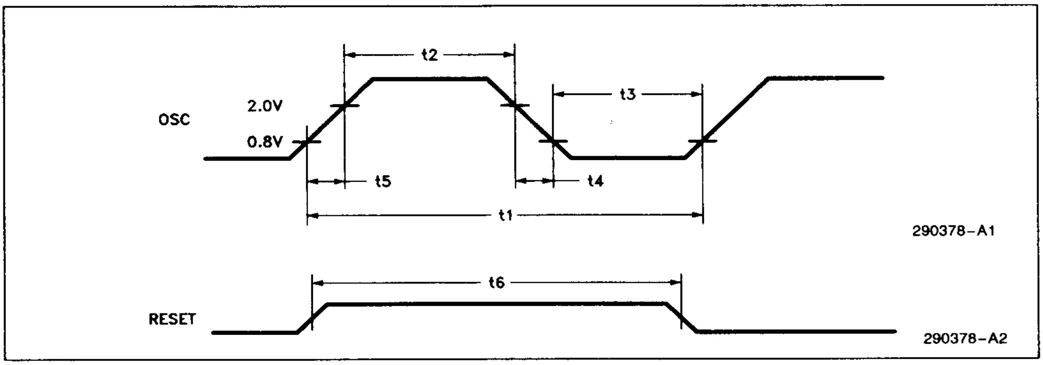


Figure 16-2. OSC, RESET Waveforms

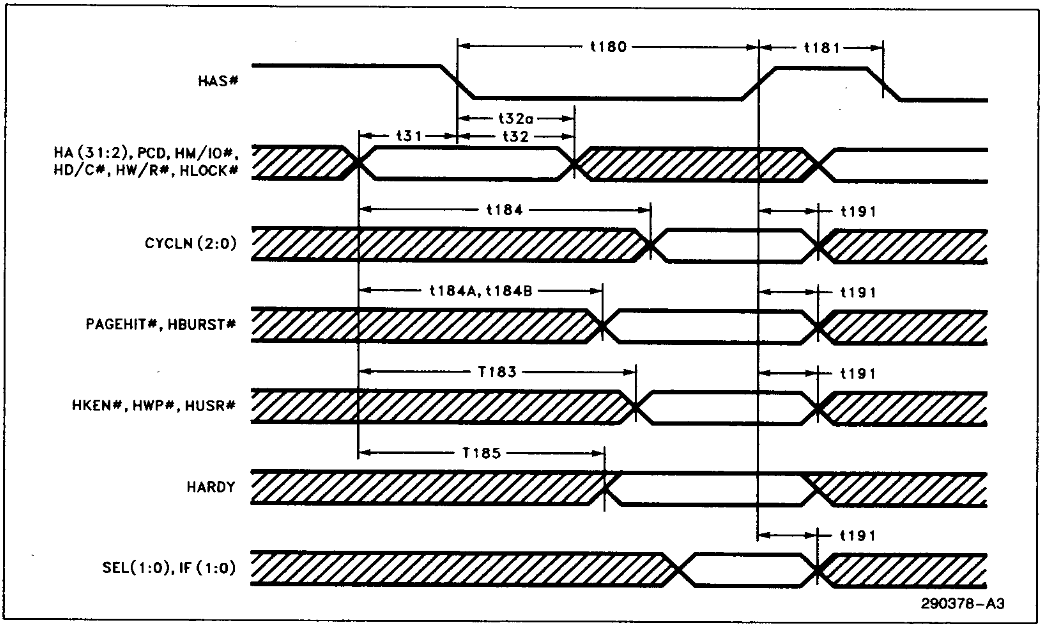


Figure 16-3. Host Interface

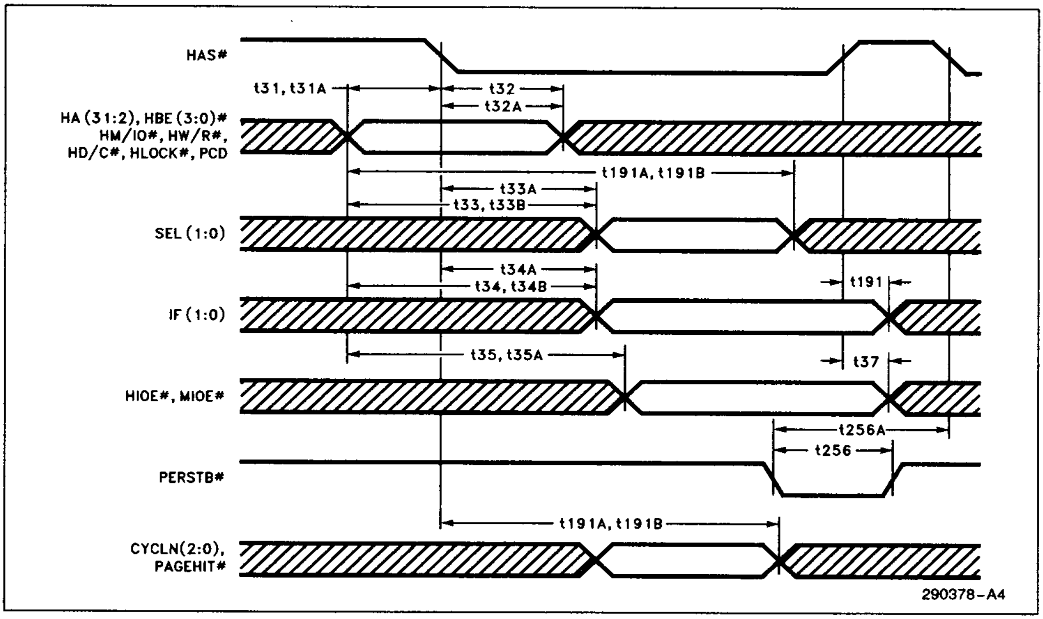


Figure 16-4. Host to Memory Interface

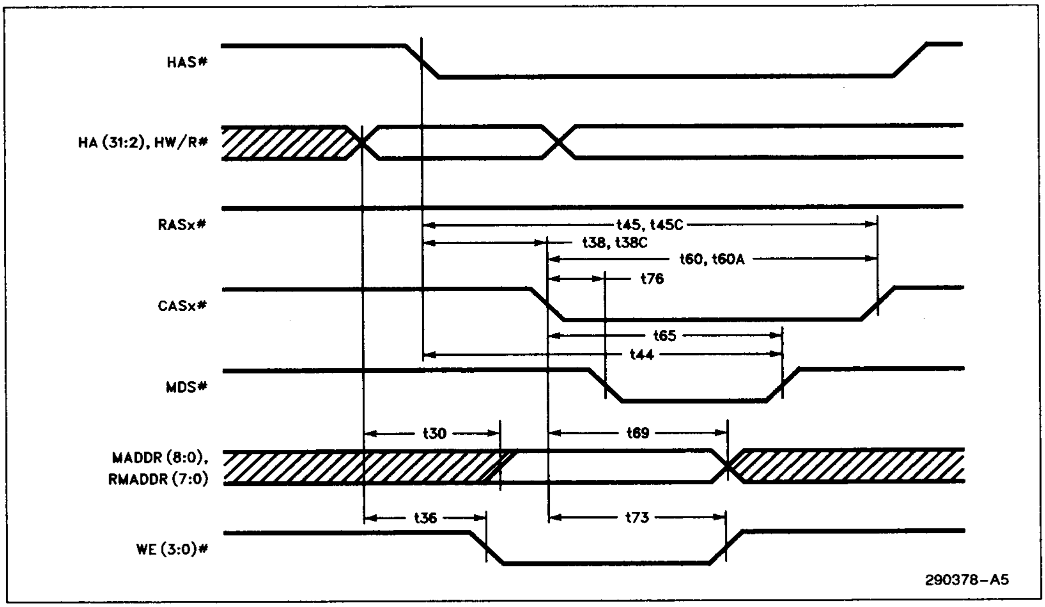


Figure 16-5. Host to Memory Cycle, Page Hit

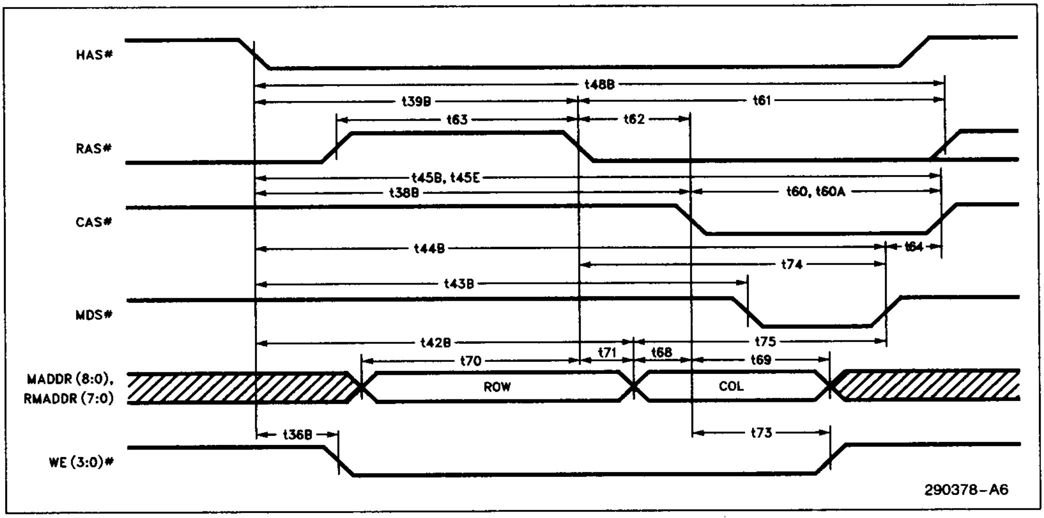


Figure 16-6. Host to Memory Cycle, Page Miss

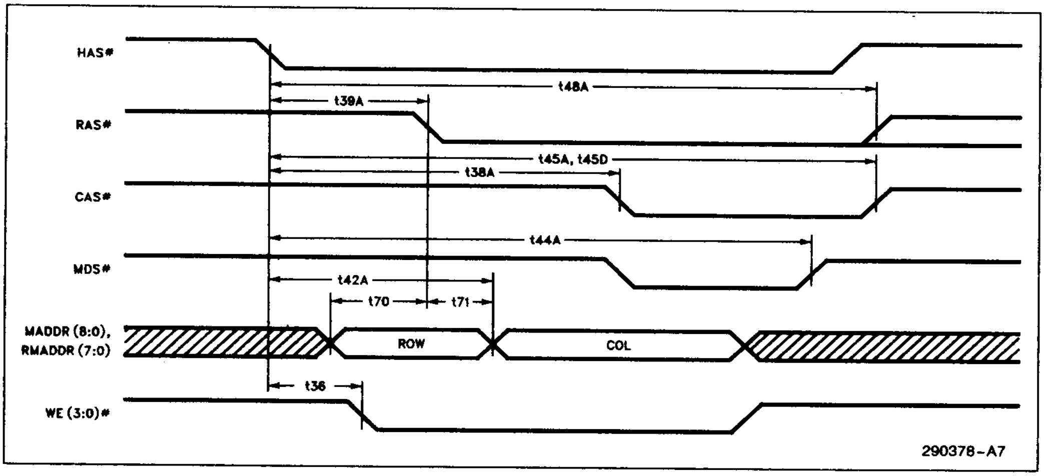


Figure 16-7. Host to Memory Cycle, Row Miss

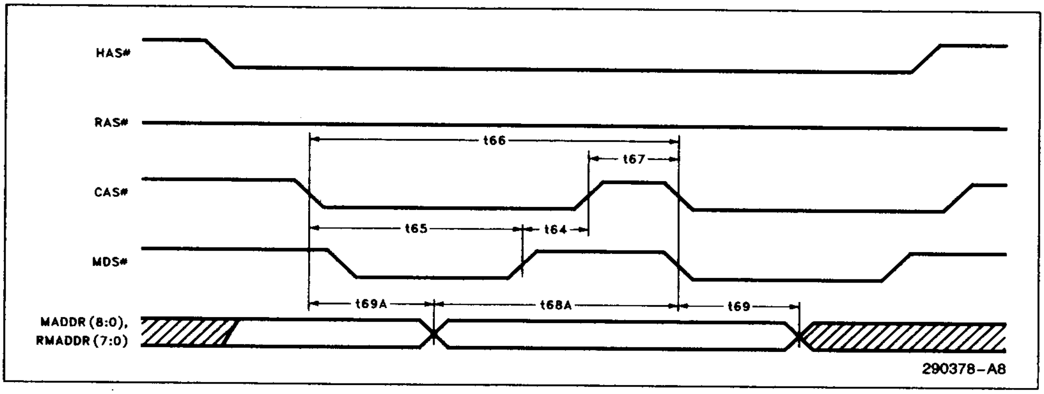


Figure 16-8. Host to Memory Cycle, Burst

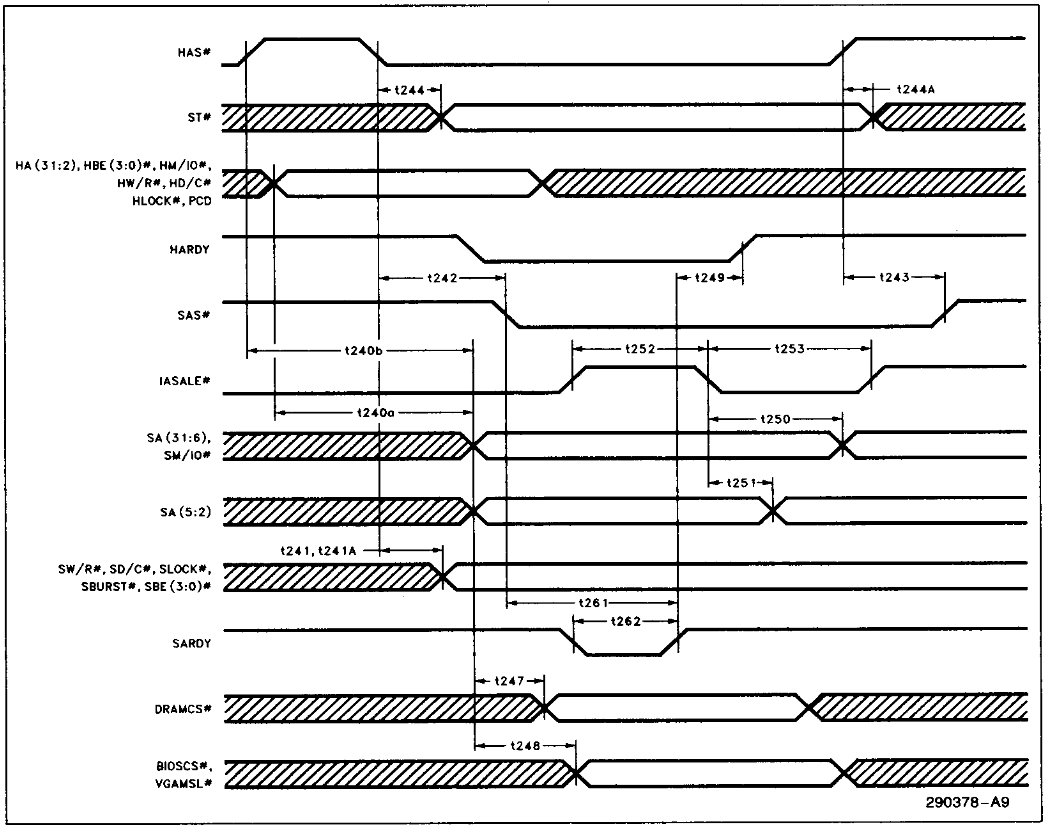


Figure 16-9. Host to System Cycle

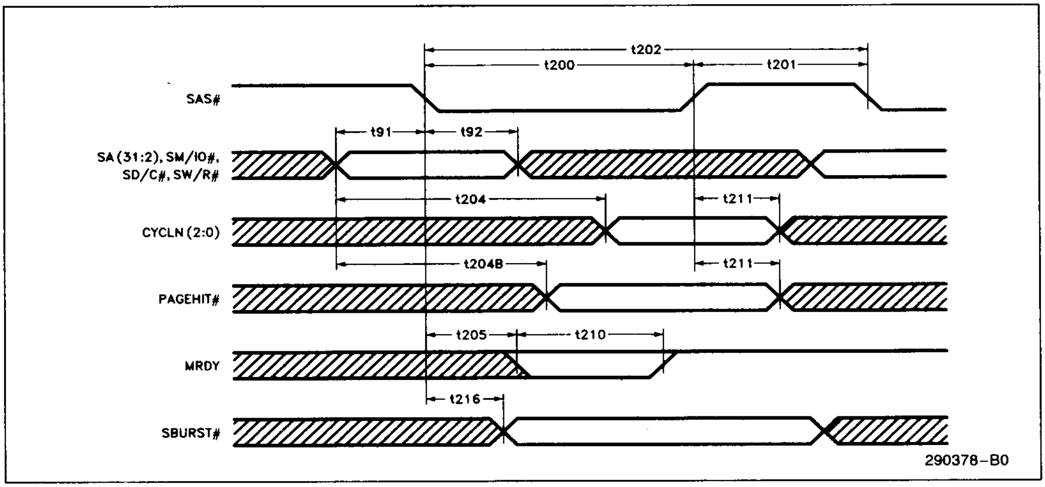


Figure 16-10. System Strobed Bus Interface

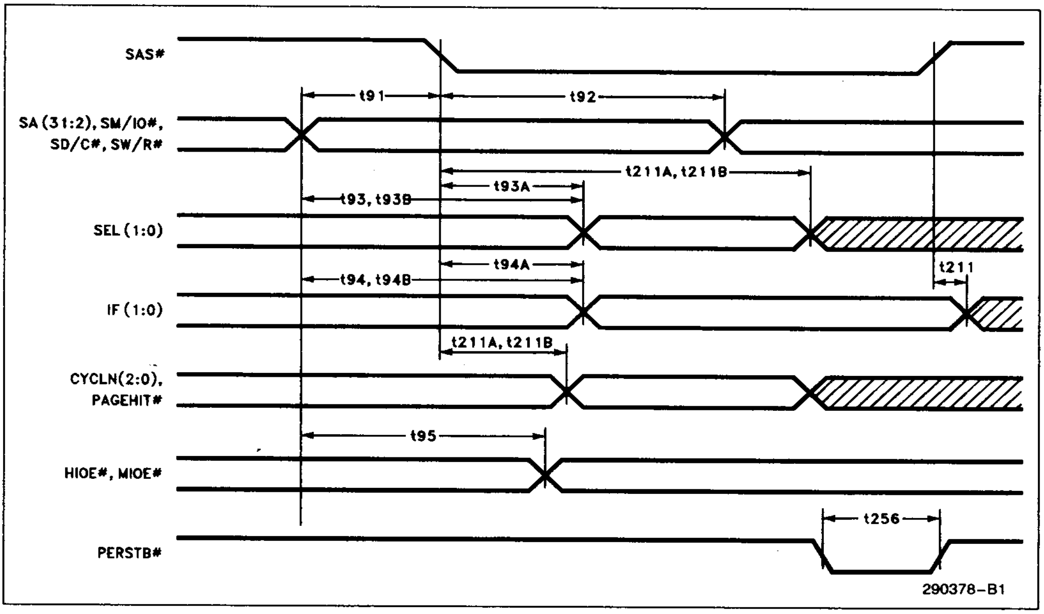


Figure 16-11. System PST Master to Memory Interface

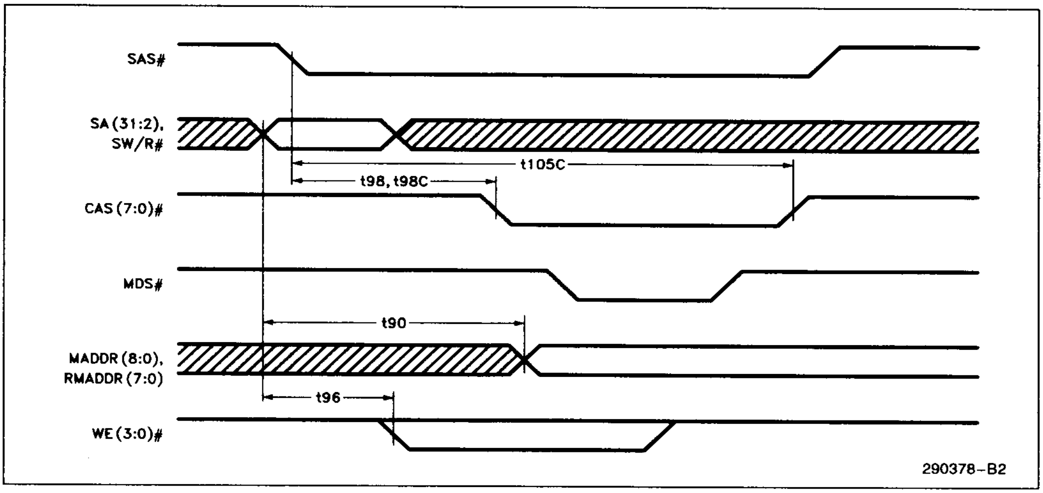


Figure 16-12. System PST Master to Memory, Page Hit

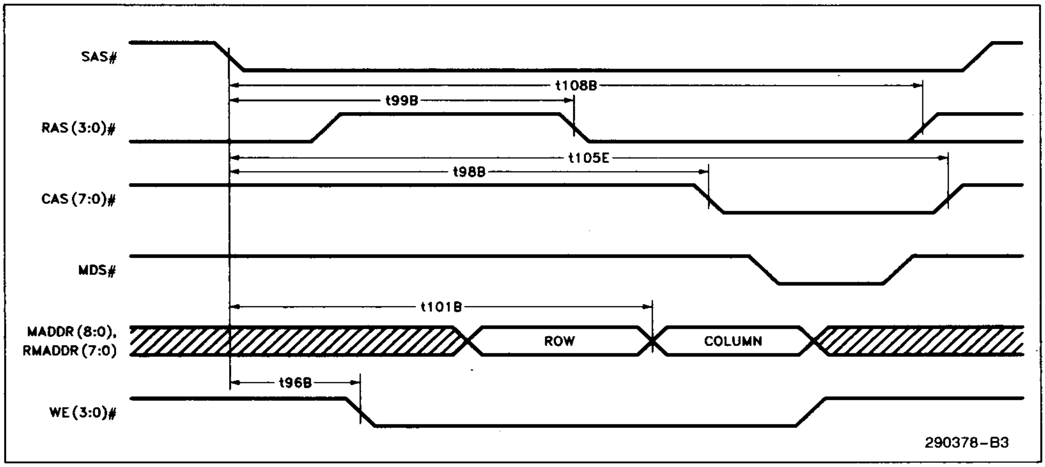


Figure 16-13. System PST Master to Memory, Page Miss

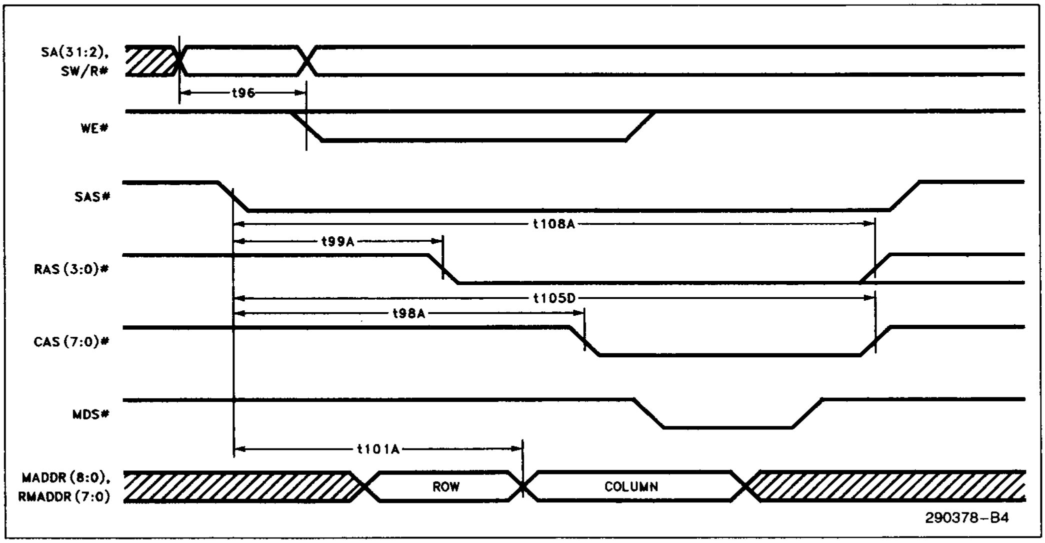


Figure 16-14. System PST Master to Memory, Row Miss

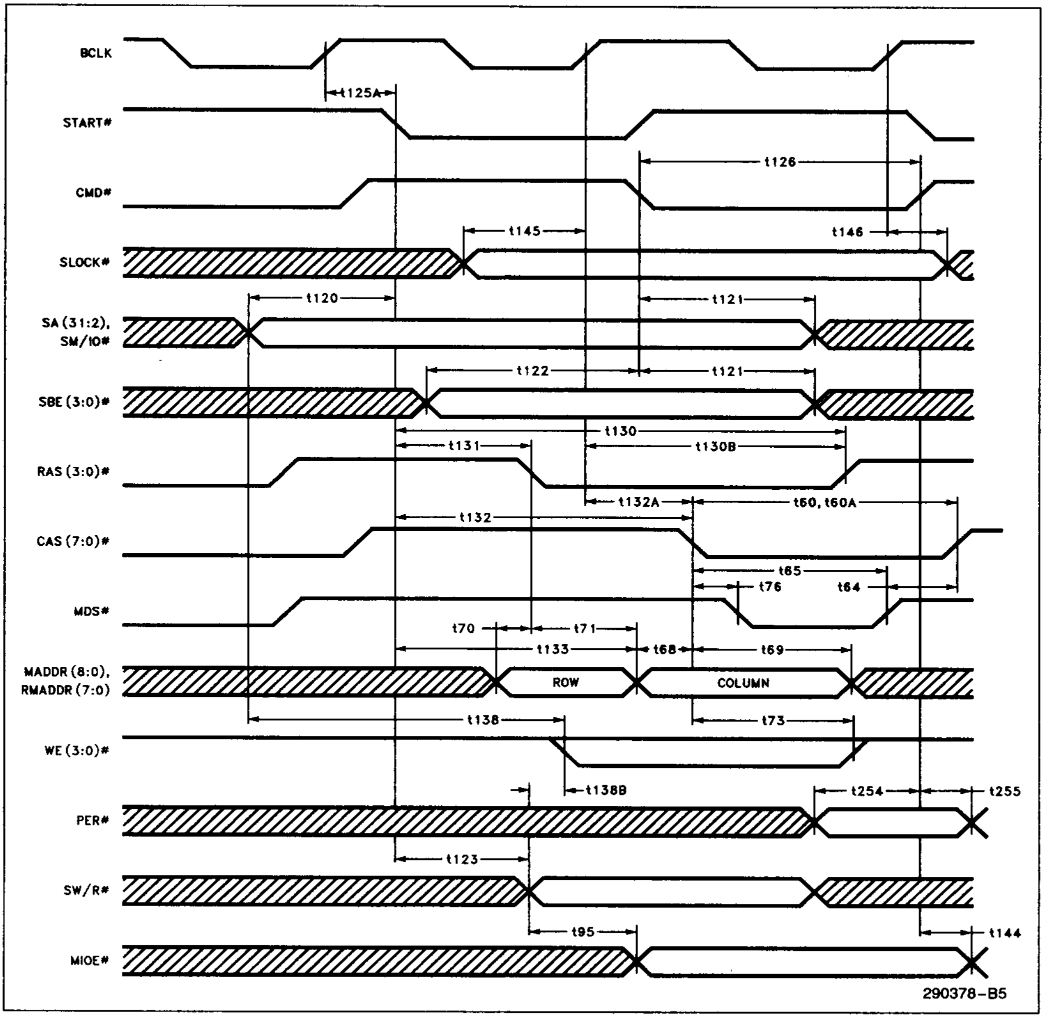


Figure 16-15. EISA Standard Cycle

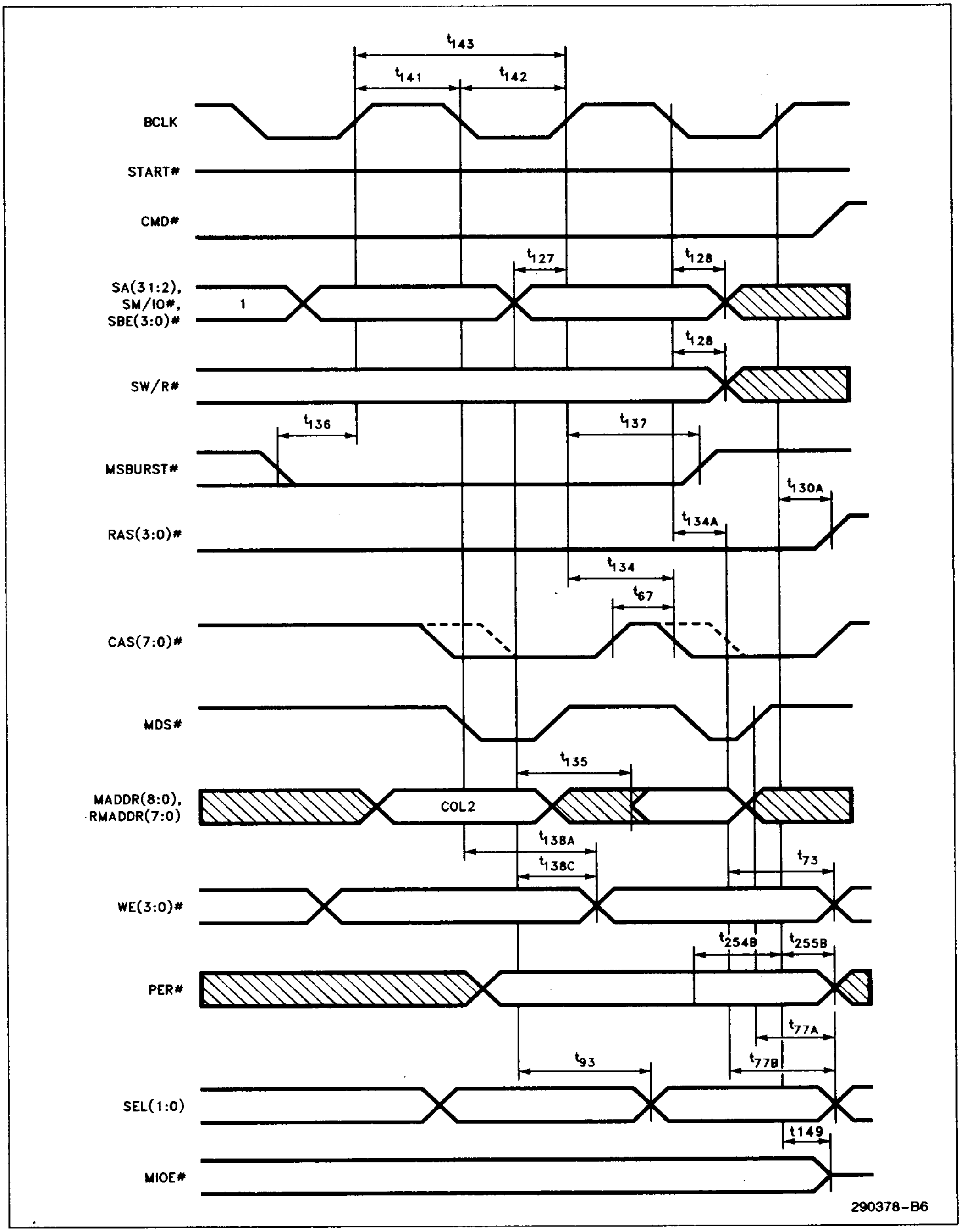
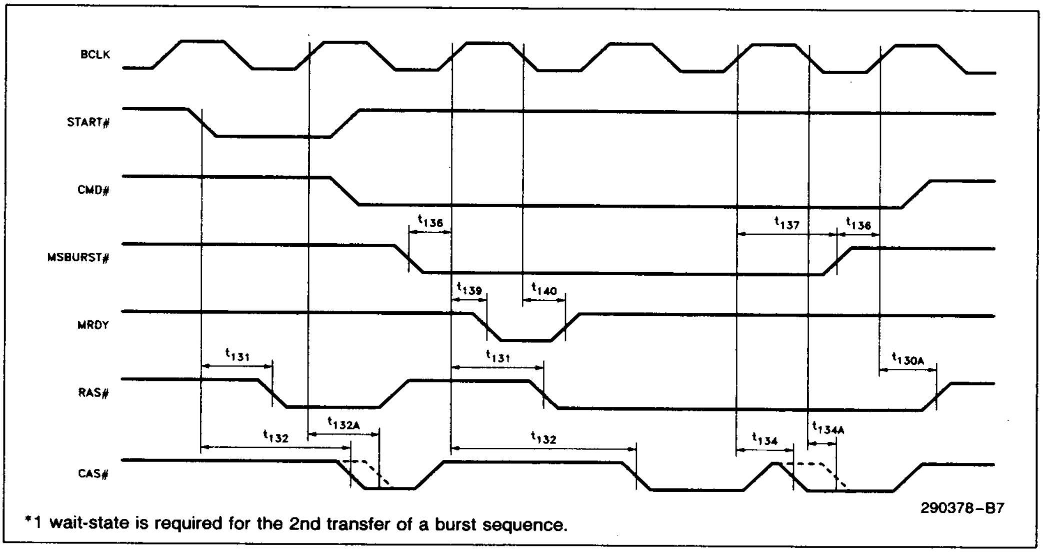


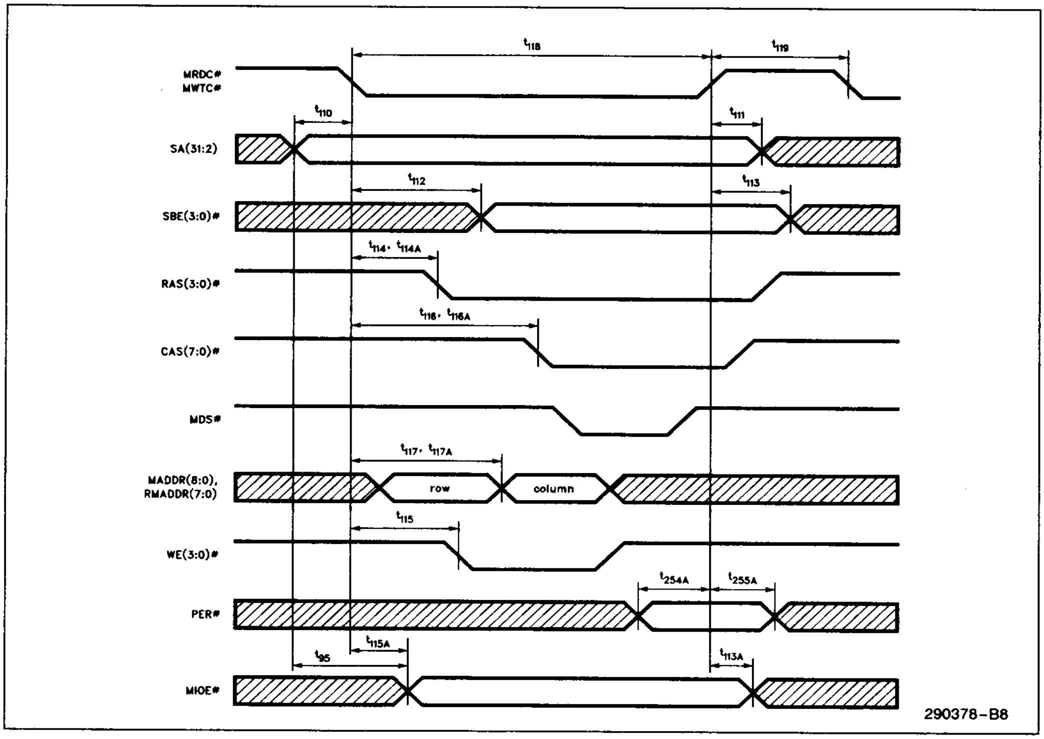
Figure 16-16. EISA Burst Cycle to Memory



290378-B7

*1 wait-state is required for the 2nd transfer of a burst sequence.

Figure 16-17. EISA Burst Cycle



290378-B8

Figure 16-18. ISA Master to Memory Interface

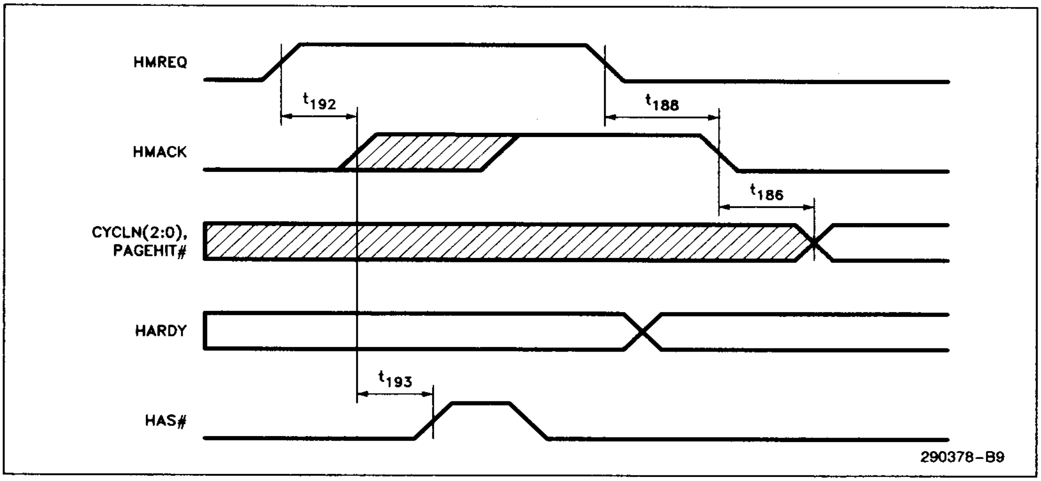


Figure 16-19. Host Arbitration Interface

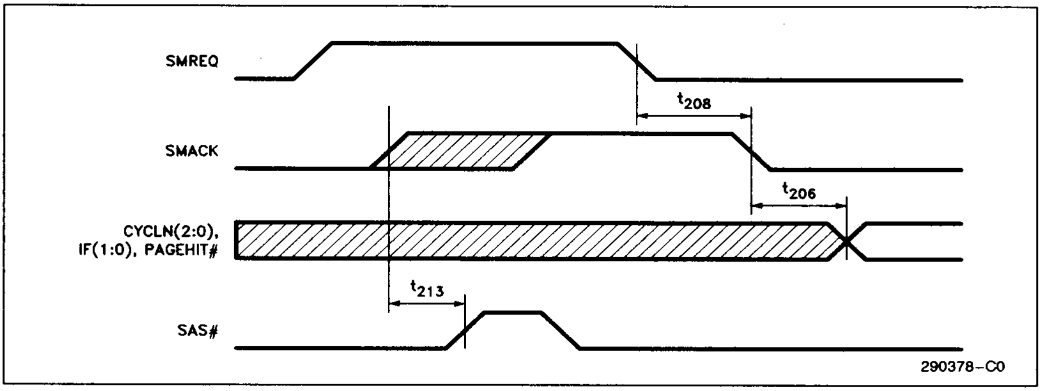


Figure 16-20. System Arbitration Interface

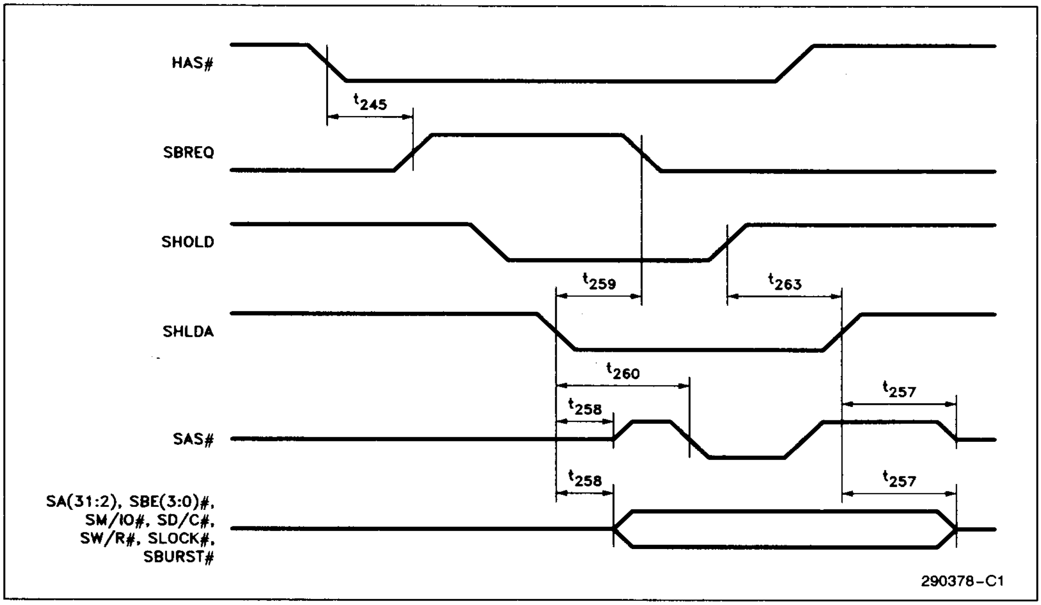


Figure 16-21. Host to System Arbitration

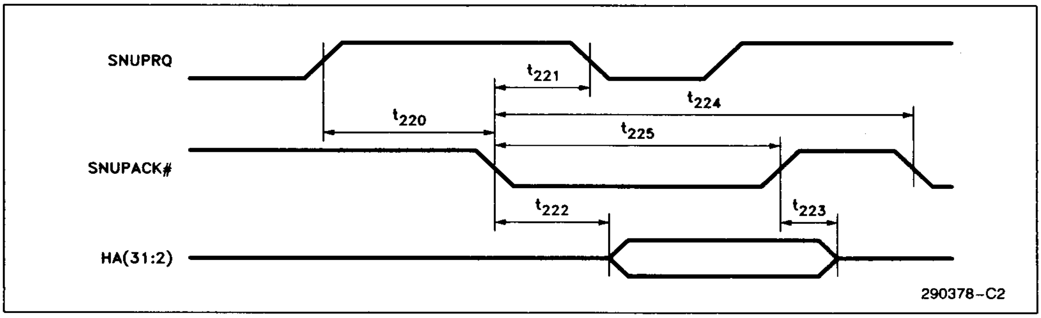


Figure 16-22. Snoop Interface

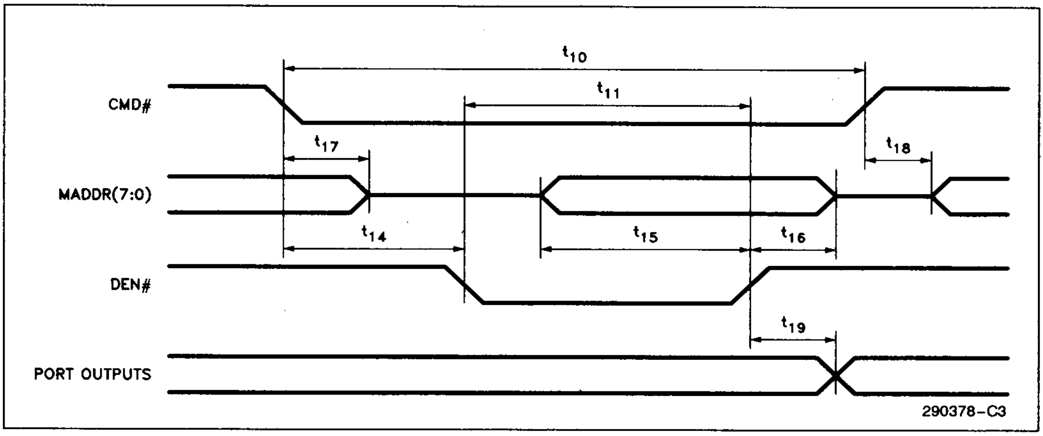


Figure 16-23. Slave Port Write Cycles

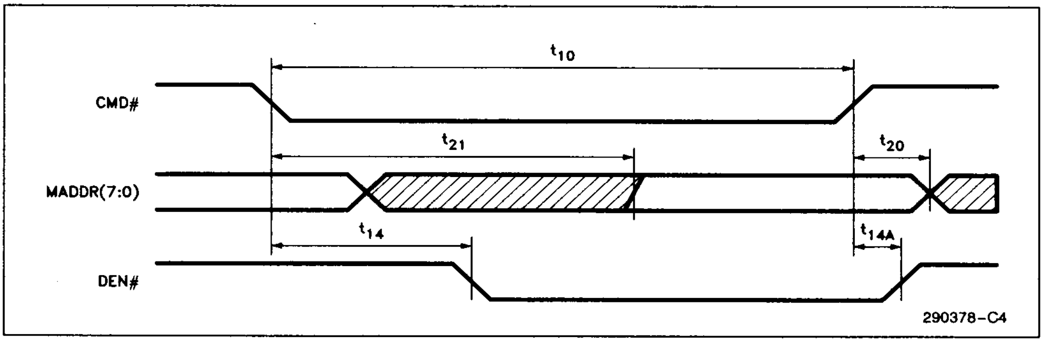


Figure 16-24. Slave Port Read Cycles

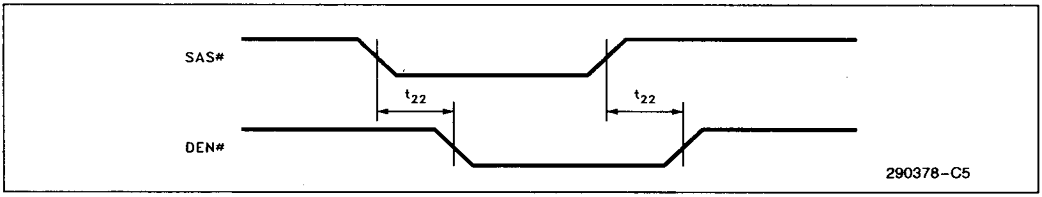


Figure 16-25. 486 Flush Cycle

16.4 DRIVER CHARACTERIZATION DATA

As systems become more complex, and components operate at higher frequencies, further information on the performance of a component's output drivers becomes helpful or even necessary. This section offers two kinds of data relevent to the A.C. performance of the 82359 drivers:

1) *Capacitive Derating Curves* for quickly adjusting timing delays under simplified loads.

2) *Output V/I Plots* for predicting or modeling an output's performance under more complex loads.

Here, a "simplified load" involves viewing the load driven by an output (traces, inputs, connectors . . .) as a sum of smaller capacitances. This method may often be valid, particularly when the trace driven is short. Longer traces, possibly with numerous stubs, present a more "complex load" to the output driver. Predicting an output's performance under these configurations requires a basic understanding of transmission line phenomena and possibly analog simulation tools.

The 82359 has four different types of outputs, labeled A, B, C, and D. The following Table associates each 82359 output with its output driver type.

82359 Signals by Output Type	
Type	**Signal Name**
A	CAS(7:0)#, MADDR(8:0), MRDY, RMADDR(7:0), SA(5:2), WE(3:0)#
B	HARDY, HBURST#, HKEN#, HUSR#, HWP#, IF(1:0), MDS#, RAS(3:0)#, SEL(1:0), SNUPRQ
C	PAGEHIT#, CYCLN(2:0), DEN#, DRAMCS#, HA(31:2), PERSTB#, SA(31:6), SAS#, SBE(3:0)#, SBURST#, SD/C#, SHLDA, SLOCK#, SM/IO#, SW/R#
D	BIOSCS#, HIOE#, HMREQ, H/S#, LOCKEN#, MIOE#, SBREQ, SMREQ, SPEED(1:0), VGAMSL#

To obtain capacitive derating or V/I information for a signal, first find its Type in the above Table and refer to the following sections for the appropriate curves corresponding to that Type.

16.4.1 Capacitive Derating Curves

The following information will be useful for adjusting the given A.C. Specifications in the following two cases:

1) The actual capacitive load on a signal trace is *larger* than the specified A.C. test load. In this case, a certain amount of nanoseconds, as derived from the curves, should be *added* to the stated A.C. Specification.

2) The actual capacitive load on a signal trace is *smaller* than the specified A.C. test load. In this case, a certain amount of nanoseconds, as derived from the curves, can be *subtracted* from the stated A.C. Specification.

NOTE:
This information is only useful if the actual signal trace is short enough to allow viewing its loading as a lumped capacitor. If the trace is longer, and the loads more distributed, then other methods of analysis must be used to accurately assess the effects of the load.

To assist the user in applying these curves to any specification, the vertical axes show only the change in nanoseconds per gradient. Use the curves to simply determine the amount of nanoseconds to either add to, or subtract from, the given delay.

NOTE:
All derating curves are derived from design and process characteristics, and are not guaranteed by test.

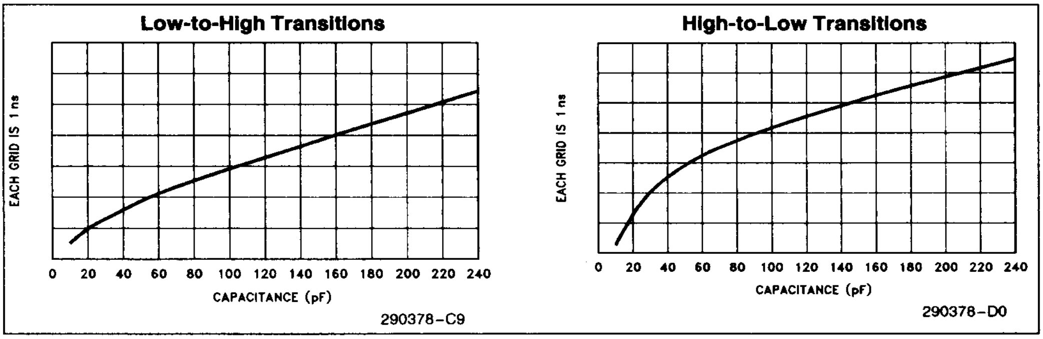


Figure 16-26. Type A Capacitive Derating Curves

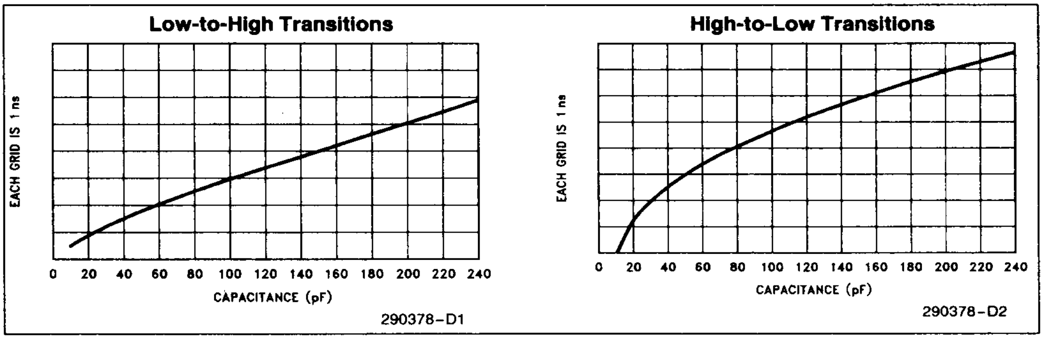


Figure 16-27. Type B Capacitive Derating Curves

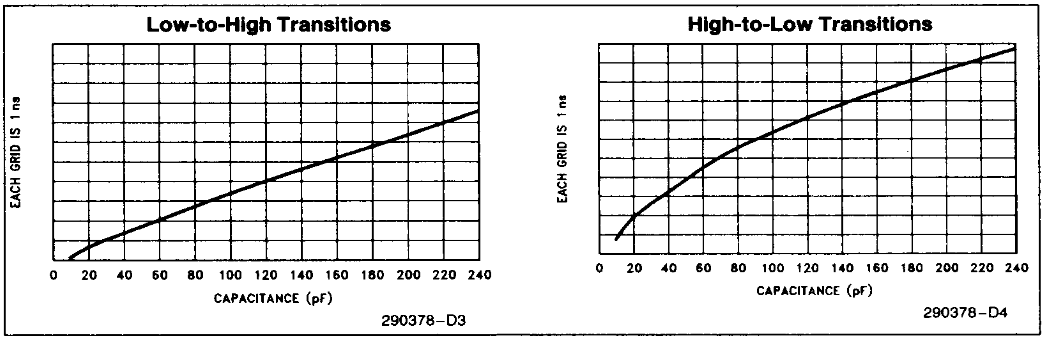


Figure 16-28. Type C Capacitive Derating Curves

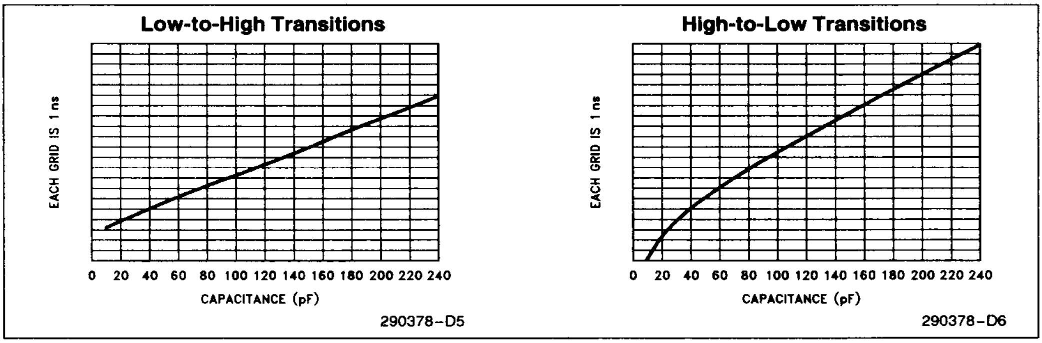


Figure 16-29. Type D Capacitive Derating Curves

16.4.2 Output V/I Plots

Following are V/I plots for each 82359 driver Type (A, B, C, and D). These plots can be used to obtain:

a) output modeling information

b) a measure of the instantaneous value of the output impedance

c) the magnitude of driven voltage steps through graphical analysis

NOTES:

1. These plots should NOT be used to extract DC parameter data.
2. These plots are derived from process and design characteristics, and are not guaranteed by test.

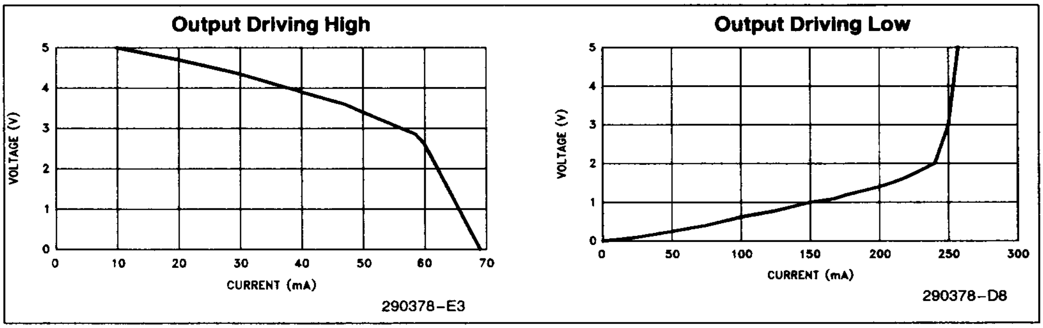


Figure 16-30. Type A V/I Plots

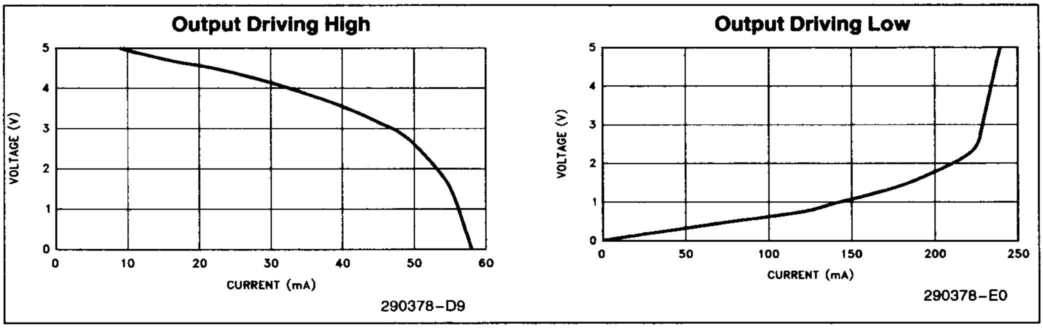


Figure 16-31. Type B V/I Plots

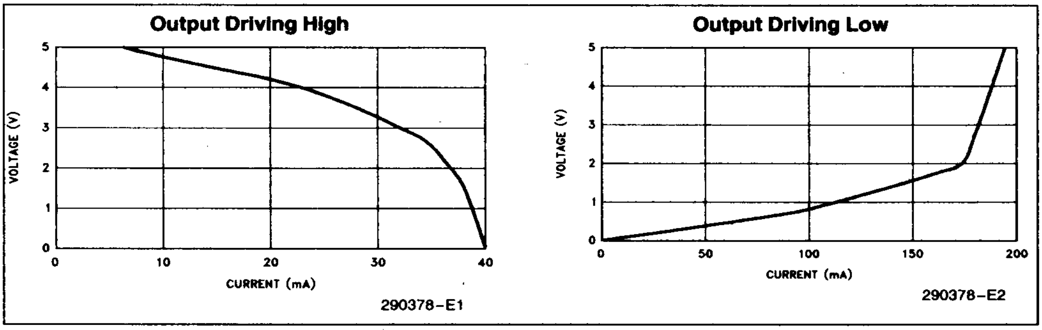


Figure 16-32. Type C V/I Plots

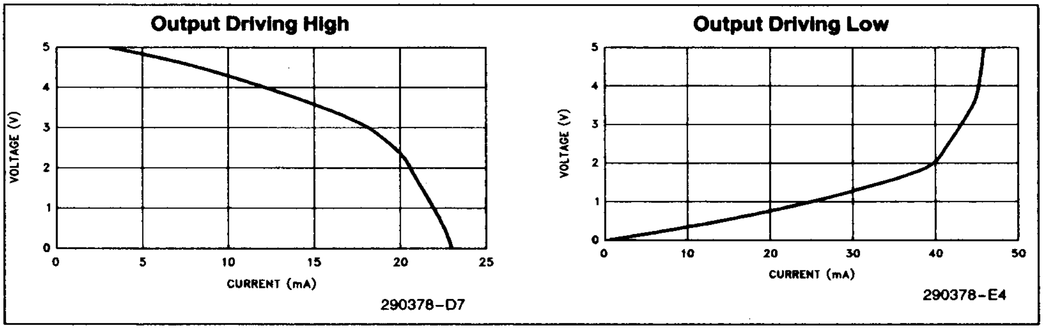


Figure 16-33. Type D V/I Plots

17.0 MECHANICAL SPECIFICATIONS

See System Architecture Overview for the 82359's package dimensions.

18.0 THERMAL SPECIFICATIONS

The 82359 is specified for operation when the case temperature is within the range of −65°C to +110°C. The case temperature may be measured in any environment to determine whether the device is within the specified operating range.

The PQFP case temperature should be measured at the center of the top surface opposite the pins, as shown below.

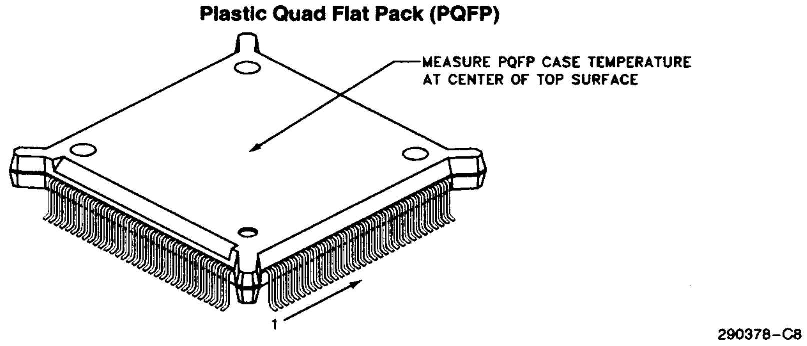

Plastic Quad Flat Pack (PQFP)
MEASURE PQFP CASE TEMPERATURE AT CENTER OF TOP SURFACE
1
290378-C8

Figure 18-1. 82359 PQFP Package Thermal Characteristics

Thermal Resistance—°C/Watt					
Parameter	Air Flow Rate (ft/min)				
	0	200	400	600	800
θ Junction to Case	5	5	5	5	5
θ Case to Ambient	17	14	11.5	9	8

NOTES:
1. The table above applies to the 82359 PQFP plugged into socket or soldered directly into board.
2. $\theta_{JA} = \theta_{JC} + \theta_{CA}$.
   Process Name: CHMOSIV

19.0 PIN NUMBER REFERENCE (Numerical Listing)

001	V_{CC}	050	V_{CC}	099	V_{CC}	148	V_{CC}
002	SNUPACK#	051	SA22	100	HM/IO#	149	HS#
003	HA27	052	HA22	101	HW/R#	150	MIOE#
004	SA17	053	V_{SS}	102	HD/C#	151	HIOE#
005	HA17	054	MADDR2	103	SLOCK#	152	PAGEHIT#
006	V_{SS}	055	SA06	104	SM/IO#	153	CYCLN2
007	MADDR5	056	HA06	105	V_{SS}	154	V_{CC}
008	SA09	057	SA19	106	SW/R#	155	CYCLN1
009	HA09	058	HA19	107	SD/C#	156	CYCLN0
010	SA20	059	V_{SS}	108	SHOLD	157	SPEED1
011	HA20	060	MADDR6	109	SHLDA	158	SPEED0
012	V_{SS}	061	SA10	110	SBREQ	159	LOCKEN#
013	MADDR1	062	HA10	111	SARDY	160	DEN#
014	SA05	063	V_{SS}	112	MRDY	161	V_{CC}
015	HA05	064	SA25	113	MSBURST#	162	HBE3#
016	SA24	065	HA25	114	SBURST#	163	SBE3#
017	HA24	066	RMADDR7	115	SMREQ	164	HBE2#
018	SA15	067	V_{SS}	116	CMD#	165	SBE2#
019	HA15	068	RMADDR5	117	START#	166	HBE1#
020	V_{CC}	069	V_{CC}	118	BCLK	167	SBE1#
021	MADDR3	070	RMADDR3	119	MRDC#	168	HBE0#
022	SA07	071	RMADDR1	120	MWTC#	169	SBE0#
023	HA07	072	SA14	121	SAS#	170	NC
024	SA13	073	HA14	122	SMACK	171	NC
025	HA13	074	MADDR8	123	V_{CC}	172	HWP#
026	V_{SS}	075	SA12	124	OSC	173	HUSR#
027	SA02	076	HA12	125	V_{SS}	174	HKEN#
028	HA02	077	SA18	126	CAS7#	175	HBURST#
029	RAMDDR6	078	V_{SS}	127	CAS3#	176	VGAMSL#
030	V_{SS}	079	HA18	128	CAS6#	177	DRAMCS#
031	RMADDR4	080	MADDR0	129	CAS2#	178	BIOSCS#
032	V_{CC}	081	V_{SS}	130	MDS#	179	SA16
033	RMADDR2	082	SA04	131	CAS5#	180	HA16
034	RMADDR0	083	HA04	132	CAS1#	181	MADDR4
035	SA26	084	INVLA#	133	CAS4#	182	V_{SS}
036	HA26	085	REFRESH#	134	CAS0#	183	SA08
037	V_{SS}	086	PERSTB#	135	RAS3#	184	HA08
038	SA11	087	V_{SS}	136	RAS2#	185	SNUPRQ
039	HA11	088	PER#	137	RAS1#	186	V_{CC}
040	MADDR7	089	TEST#	138	RAS0#	187	SA31
041	SA21	090	RESET	139	WE3#	188	HA31
042	HA21	091	PCD	140	WE2#	189	SA30
043	V_{SS}	092	HARDY	141	WE1#	190	HA30
044	SA23	093	HMREQ	142	WE0#	191	SA29
045	HA23	094	HMACK	143	SEL1	192	HA29
046	SA03	095	HAS#	144	SEL0	193	SA28
047	HA03	096	ST#	145	IF1	194	HA28
048	IASALE#	097	HLOCK#	146	IF0	195	SA27
049	V_{SS}	098	V_{SS}	147	V_{SS}	196	V_{SS}

20.0 PIN NUMBER REFERENCE (Alphabetical Listing)

118	BCLK	166	HBE1#	034	RMADDR0	108	SHOLD
178	BIOSCS#	164	HBE2#	071	RMADDR1	103	SLOCK#
134	CAS0#	162	HBE3#	033	RMADDR2	104	SM/IO#
132	CAS1#	175	HBURST#	070	RMADDR3	122	SMACK
129	CAS2#	102	HD/C#	031	RMADDR4	115	SMREQ
127	CAS3#	151	HIOE#	068	RMADDR5	002	SNUPACK#
133	CAS4#	174	HKEN#	066	RMADDR7	185	SNUPRQ
131	CAS5#	097	HLOCK#	027	SA02	158	SPEED0
128	CAS6#	100	HM/IO#	046	SA03	157	SPEED1
126	CAS7#	094	HMACK	082	SA04	096	ST#
116	CMD#	093	HMREQ	014	SA05	117	START#
156	CYCLN0	149	HS#	055	SA06	106	SW/R#
155	CYCLN1	173	HUSR#	022	SA07	089	TEST#
153	CYCLN2	101	HW/R#	183	SA08	050	V_{CC}
160	DEN#	172	HWP#	008	SA09	020	V_{CC}
177	DRAMCS#	048	IASALE#	061	SA10	069	V_{CC}
028	HA02	146	IF0	038	SA11	186	V_{CC}
047	HA03	145	IF1	075	SA12	148	V_{CC}
083	HA04	084	INVLA#	024	SA13	001	V_{CC}
015	HA05	159	LOCKEN#	072	SA14	099	V_{CC}
056	HA06	080	MADDR0	018	SA15	032	V_{CC}
023	HA07	013	MADDR1	179	SA16	161	V_{CC}
184	HA08	054	MADDR2	004	SA17	154	V_{CC}
009	HA09	021	MADDR3	077	SA18	123	V_{CC}
062	HA10	181	MADDR4	057	SA19	176	VGAMSL#
039	HA11	007	MADDR5	010	SA20	196	V_{SS}
076	HA12	060	MADDR6	041	SA21	182	V_{SS}
025	HA13	040	MADDR7	051	SA22	098	V_{SS}
073	HA14	074	MADDR8	044	SA23	043	V_{SS}
019	HA15	130	MDS#	016	SA24	049	V_{SS}
180	HA16	150	MIOE#	064	SA25	053	V_{SS}
005	HA17	119	MRDC#	035	SA26	059	V_{SS}
079	HA18	112	MRDY	195	SA27	037	V_{SS}
058	HA19	113	MSBURST#	193	SA28	030	V_{SS}
011	HA20	120	MWTC#	191	SA29	006	V_{SS}
042	HA21	170	NC	189	SA30	012	V_{SS}
052	HA22	171	NC	187	SA31	026	V_{SS}
045	HA23	124	OSC	111	SARDY	063	V_{SS}
017	HA24	152	PAGEHIT#	121	SAS#	067	V_{SS}
065	HA25	091	PCD	169	SBE0#	147	V_{SS}
036	HA26	088	PER#	167	SBE1#	125	V_{SS}
003	HA27	086	PERSTB#	165	SBE2#	087	V_{SS}
194	HA28	029	RAMDDR6	163	SBE3#	105	V_{SS}
192	HA29	138	RAS0#	110	SBREQ	078	V_{SS}
190	HA30	137	RAS1#	114	SBURST#	081	V_{SS}
188	HA31	136	RAS2#	107	SD/C#	142	WE0#
092	HARDY	135	RAS3#	144	SEL0	141	WE1#
095	HAS#	085	REFRESH#	143	SEL1	140	WE2#
168	HBE0#	090	RESET	109	SHLDA	139	WE3#

21.0 PINOUT DIAGRAM

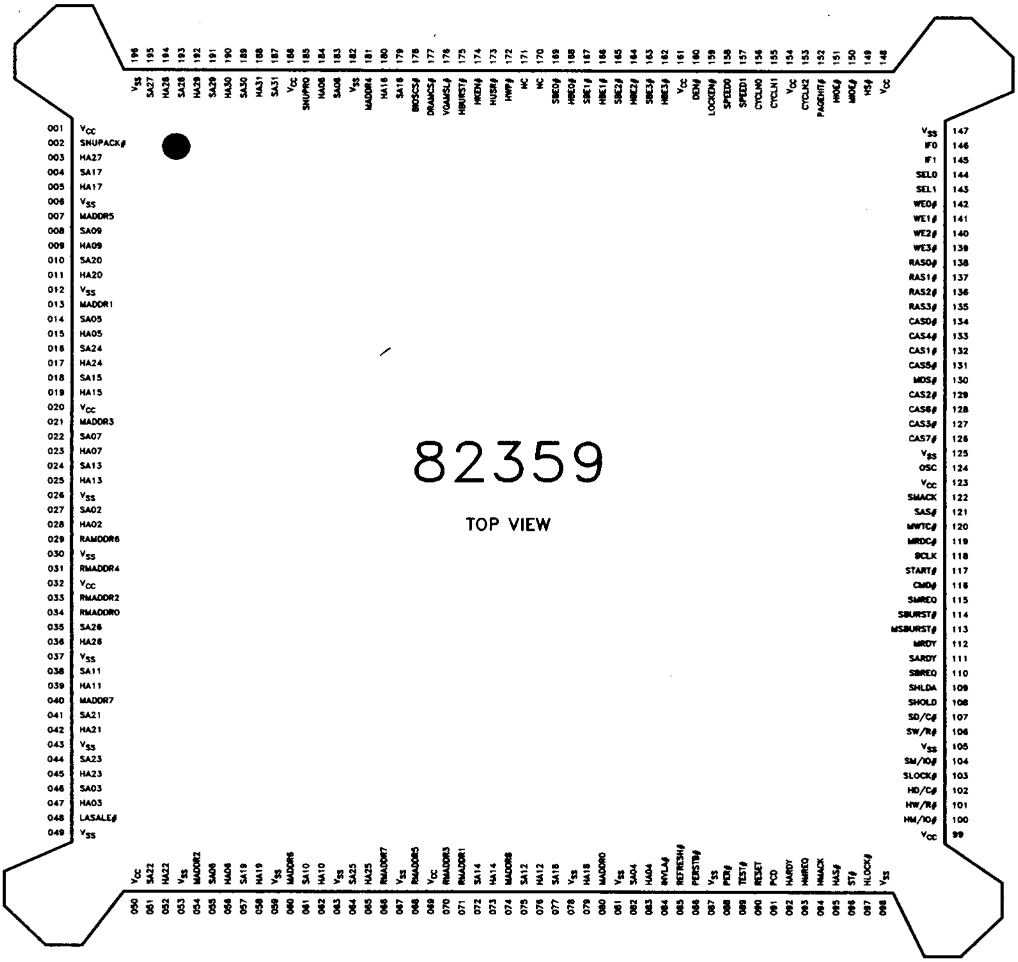


Figure 21-1. 82359 Pinout Diagram

APPENDIX A

GENERAL REGISTER LIST

Offset	Access	Description
00h	W/R	Memory Configuration Register, Row 0
01h	W/R	Memory Configuration Register, Row 1
02h	W/R	Memory Configuration Register, Row 2
03h	W/R	Memory Configuration Register, Row 3
04h	W/R	DRAM Speed Register
05h	W/R	Line Size Register
06h	W/R	Page Mode Register
07h	W/R	Block Cache Enable Register
08h	W/R	Mode Register A
09h	W/R	Mode Register B
0Ah	W/R	Mode Register C
10h	W/R	Timing Control Register, Host Delay
11h	W/R	Timing Control Register, Host to System Delay
12h	W/R	Timing Control Register, System Delay
13h	W/R	Timing Control Register, RAS# Precharge
14h	W/R	Timing Control Register, Row Timing
15h	W/R	Timing Control Register, Column Timing
16h	W/R	Timing Control Register, CAS Pulse Width
17h	W/R	Timing Control Register, CAS to MDS Delay
21h	W/R	Chip ID Register
22h	W/R	Index Relocation Register
23h	W/R	Data Relocation Register
28h	R	Parity Error Trap Register, A(7:2)
29h	R	Parity Error Trap Register, A(15:8)
2Ah	R	Parity Error Trap Register, A(23:16)
2Bh	R	Parity Error Trap Register, A(31:24)
2Ch	R	Parity Error Trap Register, Status and BE(3:0)#
30h	W/R	Cycle Length Feedback Register, Read Page Hit
31h	W/R	Cycle Length Feedback Register, Read Page Miss
32h	W/R	Cycle Length Feedback Register, Read Row Miss
33h	W/R	Cycle Length Feedback Register, Write Page Hit
34h	W/R	Cycle Length Feedback Register, Write Page Miss
35h	W/R	Cycle Length Feedback Register, Write Row Miss
40h	W/R	Lower Memory Block Enable Register, LMEMBE0
41h	W/R	Lower Memory Block Enable Register, LMEMBE1
42h	W/R	Video RAM Area Block Enable Register, VRAMABE0
43h	W/R	Video RAM Area Block Enable Register, VRAMABE1
44h	W/R	Expansion ROM Block Enable Register, EROMABE0
45h	W/R	Expansion ROM Block Enable Register, EROMABE1
46h	W/R	BIOS Area Block Enable Register, BIOSABE0
47h	W/R	BIOS Area Block Enable Register, BIOSABE1
4Eh	W/R	Remap Enable Register

GENERAL REGISTER LIST (Continued)

Offset	Access	Description
50h	W/R	PAM Register, PAM0, Attributes, Block Size
51h	W/R	PAM Register, PAM0, A(15:11)
52h	W/R	PAM Register, PAM0, A(23:16)
53h	W/R	PAM Register, PAM0, A(31:24)
54h	W/R	PAM Register, PAM1, Attributes, Block Size
55h	W/R	PAM Register, PAM1, A(15:11)
56h	W/R	PAM Register, PAM1, A(23:16)
57h	W/R	PAM Register, PAM1, A(31:24)
58h	W/R	PAM Register, PAM2, Attributes, Block Size
59h	W/R	PAM Register, PAM2, A(15:11)
5Ah	W/R	PAM Register, PAM2, A(23:16)
5Bh	W/R	PAM Register, PAM2, A(31:24)
5Ch	W/R	PAM Register, PAM3, Attributes, Block Size
5Dh	W/R	PAM Register, PAM3, A(15:11)
5Eh	W/R	PAM Register, PAM3, A(23:16)
5Fh	W/R	PAM Register, PAM3, A(31:24)
83h	W/R	Split Address Register, A(31:24)
84h	W/R	Split Address Register, A(23:20), Enable
85h	W/R	Cache Control Register
8Bh	W/R	System Memory Throttle, SMT
8Ch	W/R	Host Memory Throttle, HMT
8Dh	W/R	Host Memory Throttle Watchdog, HMTW
8Eh	W/R	Host System Throttle, HST
8Fh	W/R	Host System Throttle Watchdog, HSTW
90h	R	RAM Enable Register
91h	R	RAM Disable Register
92h	R	Resource Allocation Monitor, ETCL
93h	R	Resource Allocation Monitor, ETCH
94h	R	Resource Allocation Monitor, HMRL
95h	R	Resource Allocation Monitor, HMRH
96h	R	Resource Allocation Monitor, SMRL
97h	R	Resource Allocation Monitor, SMRH
98h	R	Resource Allocation Monitor, HMOL
99h	R	Resource Allocation Monitor, HMOH
9Ah	R	Resource Allocation Monitor, SBOL
9Bh	R	Resource Allocation Monitor, SBOH
9Ch	R	Resource Allocation Monitor, HRSL
9Dh	R	Resource Allocation Monitor, HRSH
9Eh	R	Resource Allocation Monitor, MOTL
9Fh	R	Resource Allocation Monitor, MOTH

LIM REGISTERS*

Offset	Access	Description
00h	W/R	LIM Control Register
80h 81h	W/R W/R	LIM Page 0, A(21:14) LIM Page 0, A(23:22) and LIM Page Enable
82h 83h	W/R W/R	LIM Page 1, A(21:14) LIM Page 1, A(23:22) and LIM Page Enable
84h 85h	W/R W/R	LIM Page 2, A(21:14) LIM Page 2, A(23:22) and LIM Page Enable
86h 87h	W/R W/R	LIM Page 3, A(21:14) LIM Page 3, A(23:22) and LIM Page Enable
88h 89h	W/R W/R	LIM Page 4, A(21:14) LIM Page 4, A(23:22) and LIM Page Enable
8Ah 8Bh	W/R W/R	LIM Page 5, A(21:14) LIM Page 5, A(23:22) and LIM Page Enable
8Ch 8Dh	W/R W/R	LIM Page 6, A(21:14) LIM Page 6, A(23:22) and LIM Page Enable
8Eh 8Fh	W/R W/R	LIM Page 7, A(21:14) LIM Page 7, A(23:22) and LIM Page Enable

NOTE:
*The LIM register set is completely independent of the 82359 general registers. LIM registers can only be accessed when the Chip ID Register (offset 21h) has been programmed with the value "A0h".

APPENDIX B

MEMORY SIZING ALGORITHM

The following general algorithm can be used to test the main memory for population information and DRAM size. The testing is done on a row-by-row basis, first testing for row population and then for size.

Row population is determined by assuming all four dwords in the row under test are populated with 4 Meg DRAMs. Then, by writing data to 1C00, 1C04, 1C08, and 1C0C, the 82359 performs a write to each of the four columns of the memory array, regardless of whether it is populated or not. Then these locations are read from and the data compared to the data which was written. A match indicates that the particular column is populated.

DRAM size is determined by writing four unique values to four select addresses in memory. Several of these addresses may map to the same physical DRAM location depending on the size of memory (see the table below). When the location 1C00 is read back, the data should be exactly that which was written to it. Any deviation from this means that one or more of the three other addresses is the same physical DRAM location for that particular size of DRAM. Thus from the data which is read back, the size of the DRAM devices for that row can be determined.

Physical DRAM Address Mapping for Various DRAM/Address Combinations

		Address			
		1C00	**0C00**	**0400**	**0000**
DRAM Size	4 Meg	1C00	0C00	0400	0000
	1 Meg	0C00	0C00	0400	0000
	256K	0400	0400	0400	0000
	64K	0000	0000	0000	0000

```
{ MEMORY POPULATION AND SIZING ALGORITHM }

Begin
   For all Rows Do
     Disable all rows
     Enable row to be tested as 4 Meg DRAMs, 4-way interleave

{*********************************************}
{ Test for Dword Interleave factor by writing }
{ across the entire row of memory             }
{*********************************************}
     Write 5AC3h to address 1C00h
     Write 5AC3h to address 1C04h
     Write 5AC3h to address 1C08h
     Write 5AC3h to address 1C0Ch
     If address 1C00 = 5AC3h Then
        DRAM exists in Word0
        Disp = 00h
     If address 1C04 = 5AC3h Then
        DRAM exists in Word1
        Disp = 04h
     If address 1C08 = 5AC3h Then
        DRAM exists in Word2
        Disp = 08h
     If address 1C0C = 5AC3h Then
        DRAM exists in Word3
        Disp = 0Ch

{****************************************}
{Now test for DRAM address size          }
{****************************************}
     Write 5AC3h to address 1C00H+Disp
     Write 5AD2h to address 0C00h+Disp
     Write 5AE1h to address 0400h+Disp
     Write 5AF0h to address 0000h+Disp
     Row Size Bits = [1C00+Disp] AND 03h
   End For
END.
```

REVISION SUMMARY

This data sheet contains updates and improvements to the previous version. A revision summary is listed here for your convenience.

The pages significantly revised since version -001 are:

82359 DRAM Controller

Highlights	Added reference to 16M (4M x 4) DRAM support, and changed DRAM timing resolution value.
Introduction	Changed DRAM timing resolution value.
Register Quick Reference	Changed the following register bit descriptions and values: Host Timing Register, Host to System Delay, System Timing, Row Precharge, Row Timing, Column Timing, CAS# Low Timing, and CAS# to MDS# delay registers.
Memory Ownership Protocol	Removed last sentence in first paragraph. Added to third paragraph "The 82359 will de-assert HMREQ after the Refresh, DMA, or system master cycle is complete and HAS# is active (the host has a pending cycle)".
Start of Host Cycles	The first paragraph should read "and the PST de-asserts HMACK". A note was added before the next section, "The host PST must never assert HAS# and HMACK on the same clock edge."
Locked Cycles	Added a paragraph that describes how the 82359 handles Locked cycles.
System PST Masters	This reads "The 82359 drives SMREQ to the system PST . . . ". References to "Size" were replaced with "Address depth".
Memory Array Overview	Changed Base Address remapping information.
DRAM Address Generation	Added a few paragraphs describing memory population remapping.
Decoupled Refresh Mode	Added a sentence, Decoupled Refresh Mode should only be selected in Concurrent mode.
Coupled Refresh Mode	Added a sentence, Coupled Refresh Mode can be selected in either Concurrent or Non-concurrent mode. REFRESH# Sequence Table has been corrected. Changed Prp to P2 in figure and text.
Internal Delay Lines	Changed register names in figure following the Note, changed values and names in Register Description in Table and changed DRAM timing resolution value.
System-to-Memory Throttle	Added to both the HMT and SMT descriptions: The HMT and SMT are only active in Concurrent mode. The second paragraph incorrectly described the SMT taking memory away from an EISA master during a burst cycle. This has been corrected. Added description: The HST is active in both Concurrent and Non-concurrent mode.
Snoop Filter	Reworded third paragraph.
Software Considerations	Added writing to any internal register will cause the 82359 DEN# signal (externally qualified with SW/R#) to flush the host cache. Information regarding programming the internal registers has been added.
Register Listing	Configuration register 06h is the RAS# Mode register.
Detailed Register Descriptions	Added writing to any internal register will cause the 82359 DEN# signal (externally qualified with SW/R#) to flush the host cache.
Memory Configuration Registers	In DRAM Type paragraph, and in the table below it references to "size" and "type" are changed to "address depth".
RAS# Mode Register	Bit 7 is no longer used, and the default value is ******01b. RAS# Timeout is no longer supported.
Mode Register A	Revised text in Refresh mode field description.

82359 DRAM Controller (Continued)

Host Timing Register In Host Delay (HP1) table, the 20.0 ns delay has been removed and the delay values have changed for HP1 and HP10.

Host to System Delay Register HP11 delay values have changed.

System Timing Register SP1 and SP10 delay values have changed.

Row Precharge Timing Register P2 delay values have changed.

Row Timing Register P3 and P4 default values have changed.

Column Timing Register P5 and P7 delay values have changed.

CAS# to MDS# Delay Register P6 delay values have changed.

CAS# Low Timing Register The field description bit tables should be swapped. Tables for P8 and P9 have been corrected and the delay values have changed.

Programmable Attribute Map Registers Added note at the end of the page. Also note that the starting address of a PAM block is a function of the block size (i.e., a block of size N may start on any Nk boundary).

Cache Control Register (CCR) The Cache Control Register (CCR) default value is *1*0*1*1b. The Cache Control Register diagram has been changed to show that bit 5 is not used, and bit 4 has been renamed to Lock Enable#. In the Lock Enable# paragraph, the last two sentences of that paragraph have been removed. The Flush Cache# paragraph has been removed.

System Memory Throttle (SMT) The SMT Register default value of 11111111b indicates an active state where system master access time to main memory is controlled.

Detailed Pin Descriptions Add note to both HAS# and HMACK signal descriptions—"HAS# and HMACK# must never be asserted on the same clock edge.

D.C. Specifications Revised note 3. I_{CC} max is 100 mA.

A.C. Specifications All of the programmable Timing Register minimum and maximum values have changed. K4, K5, K6, and K7 have been renamed K6, K7, K8, and K9 respectively, and all standard and buffered mode values have changed. Added t-spec values t33B, t34B, t45A, t45B, t93B, t94B, t115A, t125A, t138B, t138C, t191A, t191B, t211A, t211B, t256A. Consolidated t36A into t36. Deleted t47A, t47B, t47C, and t125. Deleted IF(1:0) and SEL(1:0) from t211A and t211B.

Added standard write cycle to t138, added HD/C# to t184, added SEL(1:0) to t191, added "output" to t210, added SEL(1:0) to t211, added "output" to t260. Added notes 8 to t4, t5; notes 25 to t35, t35A; notes 46 to t61; notes 43 to t69A; notes 48 to t71; notes 42 and 25 to t95; notes 44 to t111, t113, t113A, t118, t119, t126, t143; notes 39 to t180, t200; notes 1 to t241, t242; notes 7 to t260, and t263.

Deleted note 39 from t93, t94, t95; note 34 from 138. Changed t77A note 3 to note 33.

A.C. Test Loads Changed CAS(8:0)# to CAS(7:0)#. Added loading for output delay minimum specs.

Changed diagrams 3, 4, 5, 6, 10, 13, 14, and 17.

Thermal Specification Added Thermal Specification.

82359 Revision Summary

The Sections that were significantly revised since version 002 are:

Register Quick Reference
In LIM Page Translation Registers, changed translation address from A(21:24) to A(21:14).

DRAM Address Generation
Clarified functionality description and deleted memory address and row assignment table near bottom of page.

RAS# Timeout
Deleted section.

Internal Delay Lines
Corrected Figures at bottom of page.

Hardware Considerations
Added the following sentence to the second paragraph. "When programming the internal registers in non-EISA applications, the CMD# input must still be driven low as specified in the "A.C. Specifications" section. In addition, IASALE# must be low at the assertion of CMD# to latch a proper address. Other system-side signals (such as START# and BCLK) are not required for correct programming of the internal registers in non-EISA applications."

Detailed Register Descriptions
Deleted note at top of page.

RAS# Mode Register
Deleted last sentence of Register description.

Parity Error Trap Registers
Replaced zeros with "x" in the default values column of table, and added note " "x" indicates an undefined state at power up."

Memory Request to Ownership Register
Changed "BCLK" to OSC/2.

System Bus Ownership Register
Changed "BCLK" to "OSC/2".

Detailed Pin Descriptions
In HBURST#/CCRB2 pin description, removed the sentence "Burst cycles are always the length of cache line fills, and even when bursting code prefetches. In ST# pin description, removed "to main memory". The sentence now reads "i.e., only CPU reads are abortable once the cycle has begun".

Miscellaneous Decodes and Control Signals
Miscellaneous decodes and control signals—added TEST# pin description.

Functional Timing Diagrams
In paragraph after host to memory read-row miss diagram, replaced Phd/Psd with HP1/SP1. Corrected SA addresses and IF(1:0) bits in Accessing the 82359 Internal Register Diagram.

D.C. Specifications Section
Added typical I_{CC} specification of 70 mA, and changed Note 3 to say "OSC = 40 MHz, no external loads. This specification is derived from device characterization data, and is not tested." Added maximum 200 mA and typical 132 mA I_{CC} with loads specification, and changed Note 4 to say "OSC = 40 MHz. This specification is tested in a dynamic environment with maximum DC Loads (I_{OL}, I_{OH}) applied to each pin." and deleted the old Note 4. Added Note 2 to V_{OL1}. Added Note 2 to V_{OH1}. Added Note 1 to V_{OL2}. Added Note 1 to V_{OH2}.

A.C. Specifications Section
Deleted Note 31 from t20 and t210. Deleted Note 44 from t111, t113, t113A, t118, t119, t126, and t143.

t22 changed from 0 ns to −4 ns. Added "(for latching)" to t31, and "(for decoding)" to t31A. Added t37 HIOE#, MIOE# hold from HAS# inactive 0 ns minimum. Added t37 HIOE#, MIOE# hold from HAS# inactive 0 ns minimum in Figure 3. Changed t38B from "29+ ... " to "33+ ... ".

Changed t45C from "26+ ... " to "23+ ... ". Changed t45D from "36+ ... " to "28+ ... ". Changed t45E from "34.5+ ... " to "32.5+ ... ". Changed t66 from " ... +8" to " ... +9.5". Changed t74 from " ... −24 min" to " ... −27 min".

82359 Revision Summary (Continued)

Changed t120 from 10 ns min to 30.88−K1 ns min. Removed SBE(3:0)# from t127. Added t127A SBE(3:0)# Setup to BCLK Rising (Burst Cycles) of −16 ns minimum, and shown in Figure 15. Changed t128 from 2 ns to 0 ns minimum. Added t138B to Figure 14. Added t138C to Figure 15. Added specification t149 MIOE# Hold from BCLK Rising (Burst Cycles) 0 ns minimum, and added it to Figure 15. Added t188 and t208 Note 49 "This is a system requirement and is a function of external logic." Deleted t189 spec in text and Figure 2.

In t191, moved reference to CYCLN(2:0), PAGEHIT#, SEL(1:0) to the end of the spec, added reference to Page hit cycles. t191A and t191B changed from "... +10" to "... +9", added SEL(1:0) to the specs, added reference to Figure 3. In t211, moved reference to CYCLN(2:0), PAGEHIT#, SEL(1:0) to the end of the spec, added reference to Page hit cycles, and to Figure 3. t211A and t211B changed from "... +10" to "... +6", added SEL(1:0) to the specs, and added reference to Figure 10. Changed spec t215 from 100 ns to 120 ns minimum. Deleted Note 10 from t221. Added Note 19 to t242. New Note 19 reads: "In non-concurrent mode, Note 1 applies. However, in concurrent mode, SAS# will be driven after arbitration for the system bus is complete and is not dependent on the deassertion of HMACK." t258 changed from 0 ns minimum to 8.5 ns minimum.

Note 10 has been deleted. Removed reference to SHLDA low from Note 18. Added to Note 29 "For this specification the minimum capacitive load on DEN# is assumed to be 15 pF". Note 31 has been deleted. Changed Note 38 to read: "Specifications t182/t202 are component requirements for the absolute minimum allowable time for any assertion of HAS#/SAS#. However, back-to-back assertion of HAS#/SAS# for cycles accessing 82359 memory must also satisfy the following two requirements: (1) the time from the falling edge of CAS# in the first cycle to the assertion of HAS#/SAS# in the following cycle must be greater than P7 − HP1/SP1, and also (2) the time from the rising edge of that same CAS# assertion to the next assertion of HAS#/SAS# must be greater than 20 ns − HP1/SP1. Note that this will cause the actual minimum HAS#/SAS# cycle time to vary from cycle to cycle, depending on the memory organization and type of DRAM access (i.e., page hit/miss or row miss).

Changed Note 39 to read: "Specifications t180/t200 are component requirements for the absolute minimum allowable time for assertion of HAS#/SAS#. However, for every assertion of HAS#/SAS# for a cycle accessing 82359 memory must also satisfy the following requirement: HAS#/SAS# must remain asserted until the column address hold time is met, as specified by t69, for memory write cycles, and until the final MDS# rises for a memory read cycle. Note that this will cause the minimum HAS#/SAS# active time to vary from cycle to cycle, depending on the memory organization and type of DRAM access (i.e., page hit/page miss/row miss, read/write).

Note 44 has been deleted. In Figure 2, HARDY changed to be shown valid not just at a logic high, and added SEL(1:0) to the IF(1:0) waveform. Changed Figure 3 to show t191A, t191B, and t37. Changed Figure 10 to show t211, t211A, and t211B. Changed Figure 15 title from "EISA Cycle to Memory" to "EISA Burst Cycle to Memory," and changed t138 to t138C.

Driver Characterization Data	Added new section including Capacitive Derating Curves and Output V/I Plots.
Pin Number Reference	Numerical Pin List—Changed pin from "89 Pullup" to "89 TEST#".
	Alphabetical Pin List—Change pin from "89 Pullup" to "89 TEST#".

82359 Revision Summary

The sections that were significantly revised since version 003 are:

All sections had the section number inserted.

All sections had the figure numbers inserted.

Cover Page Under Flexible DRAM Support, reference to 100 ns has been removed.

Section 1.0 Paragraph 11, reference to 100 ns has been removed.

Section 2.0 Symbol SMREQ function changed. The following was deleted from sentence 2, "Since default memory ownership is given to the host". The sentence now reads "The 82359 will immediately assert SMREQ whenever the system does not request memory ownership (i.e., the default state of SMREQ is asserted)".

Host Port Interface, symbol ST#, note added.

System Port Interface, symbol SMREQ, sentence 2 has been modified. The sentence now reads, "The 82359 will immediately assert SMREQ whenever the system does not request memory ownership (i.e., the default state of SMREQ is asserted)".

Section 3.0 RAS# Mode Register, bit 7 description removed, and bit 4 description was added.

Mode Register A, default corrected to 00*01000b. Bit 0 description added.

Mode Register B, default corrected to ***11101b, Bit 4 description added.

Host Timing Register, default corrected to *1101000b.

Cache Control Register (CCR), bit 5 description changed.

Section 5.4.3 The following was deleted from sentence 5, "Since main memory ownership is given to the host by default". The sentence now reads "SMREQ's default state is asserted".

Section 6.1 Paragraph 6, reference to 100 ns was removed.

Section 6.5 New paragraph 3.

Section 6.7 New paragraph 3.

Section 6.8 Sentence 1 in paragraph 1 now reads, "Coupled Refresh Mode should only be selected in Non-concurrent mode".

Section 9.2 The following was deleted from sentence 5, "by default". The sentence now reads, "It does not have any affect when the host has memory ownership, nor does it effect any system PST master devices since they are self-throttling and can not be held-off.)".

Section 13.3 Under field description of the Line Size Register, added the following paragraph, "An important note when programming the 82359 for line sizes greater than 16 bytes: the 82359 will perform a maximum of four consecutive accesses during a burst cycle. This means that a successful 32-byte burst transfer can only be run from 2-way or 4-way interleaved memory. Similarly, a 64-byte burst access can only be run from 4-way memory. The system design must ensure that the proper memory width is in place if the line size is programmed for greater than 16 bytes".

RAS# Mode Register default values changed to: 0**1**01b.

RAS# Mode Register diagram changed. Bit 4 is reserved. The shading from this box has been removed.

RAS# Mode Register, reserved bit description has been added.

Mode Register A, default changed to 00*01000b.

Mode Register A, description added to bit 0, stepping indicator.

Mode Register B, default changed to ***11101b.

Mode Register B, SMT Enable description added.

Mode Register C, paragraph 2 added to 2BCLK/3 BCLK# EISA Single Cycle section.

Host Timing Register, default changed to *1101000b.

Host Timing Register, Host Delay (HP1), sentence 3 added.

Split Address register diagram changed. Register 7, Split Enable was changed to Split Enable#.

Cache Control Register (CCR), Bit 5 corrected to Snoop Enable#. Snoop Enable description added.

82359 Revision Summary (Continued)

Section 14.1 Host Port Interface, symbol ST#, note added after paragraph 2.

Section 14.4 Symbol MDS#: added new sentences to function. These sentences read, "Note that the 82359 will perform a maximum of four consecutive MDS# pulses. Four pulses are enough to satisfy the 16-byte default line size access from the minimum memory width of 4 bytes. To perform larger burst accesses, the design must ensure a wider memory."

Section 16.2 Note 5 added.

Section 16.3 Symbol t30, max changed to 20.

Symbol t32A added.

Symbol t75, min changed to read: "P4+P6−P3−12".

Symbol t182 deleted.

Symbols t183, t184B, t185, max changed to 33.

A.C. Specifications note 38 replaced.

A.C. Specifications note 50 added.

Figure 16-3 Symbol t182 removed, t32A added.

Figure 16-4 Symbol t32A added.